

航天器多核 CPU 的引导监控软件设计

代雅晴, 王泽波, 赵欣, 刘世佳, 马宏业, 刁文婷
(北京遥测技术研究所 北京 100076)

摘要: 目前, 航天器软件设计愈发复杂、功能逐渐增多, 如果能够将其模块进行合理拆分, 可以更好地进行编码与管理。然而考虑到成本、功耗、布线等因素, 难以让多个软件各自运行在不同的 CPU 芯片中。现在多核 CPU 技术发展迅速, 一块 CPU 芯片中可以集成多个 CPU 内核, 达到使用一块 CPU 芯片就可以同时运行多个软件的目的。以 S698PM 这款国产多核 CPU 芯片为基础, 本文提出一个多核 CPU 引导监控软件的设计方案, 使该芯片能够加载并启动 2 个 CPU 内核, 并令其运行不同的业务软件。经过实验测试证明, 该方案能够正常完成 CPU 的多个内核的引导功能, 并完成对业务软件的重构工作。目前以该方案为基础的引导监控软件在轨运行良好, 能够完成规定的所有功能。

关键词: 多核 CPU; 引导软件; 星载软件; 软件重构; 嵌入式系统; S698PM

中图分类号: TP368.1; V446

文献标志码: A

文章编号: 2095-1000(2025)06-0075-10

DOI: 10.12347/j.ycyk.20250817001

CSTR: 32406.14.ycyk.20250817001

引用格式: 代雅晴, 王泽波, 赵欣, 等. 航天器多核 CPU 的引导监控软件设计[J]. 遥测遥控, 2025, 46(6): 75-84.

Design of A Boot and Monitor Software Based on Multi-Core Processor for Spacecraft

DAI Yaqing, WANG Zebo, ZHAO Xin, LIU Shijia, MA Hongye, DIAO Wenting
(Beijing Research Institute of Telemetry, Beijing 100076, China)

Abstract: Nowadays, spacecraft software is becoming increasingly complex and its functions are gradually increasing. If it can be split reasonably in different software, the project can be managed better. However, considering factors such as cost, power consumption, and wiring, it is difficult to use different processors to run different software. Now, multi-core processor is developing rapidly. One processor contains more than one core, so that different software can be run on different cores. Therefore, this paper proposes a design of a boot and monitor software based on multi-core processor which can enable 2 cores to run 2 different software. The ground tests and on-orbit experiments indicated that this scheme can correctly perform boot for different cores and accomplish refactor software for updating. The boot and monitor software based on this design runs well on the orbiting satellites.

Keywords: Multi-core processor; Boot; Onboard software; Software reconfiguration; Embedded system; S698PM

Citation: DAI Yaqing, WANG Zebo, ZHAO Xin, et al. Design of A Boot and Monitor Software Based on Multi-Core Processor for Spacecraft[J]. Journal of Telemetry, Tracking and Command, 2025, 46(6): 75-84.

0 引言

我国航天事业发展迅速, 业务广度不断拓展, 任务难度也逐步提升^[1]。当前的航天器软件需要负责收发遥测遥控、与诸多外设通信、进行算法计算等等。若是裸机编码设计, 可以在软件中轮询各个任务; 如果使用操作系统, 可以由操作系统按照任务优先级统一调度。在航天领域中, 有许多对实时性要求较高的任务。对于这些高优先级但偶发的任务, 可以使用中断进行处理。但是如

果该任务需要持续运行并进行较为复杂的数据运算, 不仅会长期占用 CPU, 其他任务无法正常执行, 并且中断模式下频繁抢占低优先级任务, 任务调度也将消耗大量的时间。考虑到整个航天器的造价成本、功耗、布线等情况, 通常不能为该任务再增加一个单独的 CPU 并保证其可持续占用这个 CPU。如果使用多核 CPU, 则可在不增加硬件资源的情况下解决这个问题。多核 CPU 是由多个独立 CPU 内核集成的 CPU 芯片。目前除了针对多核操作系统^[2-4]及其应用^[5,6]、多核调度方法^[7]等基

基础研究, 多核还有着广泛的应用, 如各类算法的运行^[8-11]、图像与视频处理^[12-16]等。这些算法应用表明, 若能将算法的子功能置于合适的内核, 可以缩短处理时间。并且当前软件设计愈发复杂, 需要将多部门设计的各个功能模块整合在同一个软件中, 这加大了软件管理上的难度。如果能够将其合理拆分, 只需要保证数据交互接口正确, 也可以减少在软件整合过程中出现的问题, 并减轻跨部门管理上的难度。

当前多核的引导软件有很多种设计方式^[17]。通过给每个CPU内核创建Boot表, 让各个CPU内核争夺公用的数据总线控制权, 等所有内核从指定存储空间读取自己的代码之后再开始运行程序^[18]。还可以通过CPU核0完成硬件初始化与程序读取, 并将入口函数、堆地址通过FMN传送^[19], 使用IPC中断^[20,21]、远程处理器架构^[22]来唤醒其他CPU内核开始运行。

本文选用具有4个SPARC V8内核的S698PM芯片完成引导监控方案设计。该芯片每个内核上都可以运行一个独立的航天软件, 各个内核配合能够完成较为复杂的功能应用。本引导监控软件设计由一个CPU内核完成全部内核软件代码引导工作, 减少数据总线的争抢, 再由多处理器状态寄存器启动其他内核完成业务程序的运行。同时为软件加入判断业务软件是否正确和重构业务软件的监控功能。这样既可以在正常情况下完成业务软件的引导与启动, 也可以在业务软件异常或需要更新升级的情况下, 通过CAN总线^[23,24]、1553B^[25-28]总线等通道接收数据, 对业务软件进行重构, 恢复航天器的正常工作。

基于S698PM芯片, 本文为多核CPU的启动设计了一个引导监控软件:

① 该引导监控软件可以成功在S698PM芯片2个CPU内核上启动并重构不同的业务软件。通过一个内核完成所有软件的引导与监控功能, 减少了不同内核对数据总线的争抢。

② 通过共享DDR来传递其它内核的跳转地址, 实现其它内核软件启动地址灵活切换。

③ 通过多核程序运行时间测试表明, 该芯片的多个内核之间平等抢占数据总线。基于此结论并通过实际工程验证, 如果进行合理的多核软件设计, 并使用该引导监控软件进行启动管理, 可使各部门软件管理更加独立, 减少交叉。目前该

引导监控软件在轨运行良好, 能够正常启动、重构业务软件。

1 方案设计

1.1 总体设计

在产品中与本设计相关的内容如图1所示。该产品的CPU为S698PM, 其外接3种存储设备: 存储引导监控软件的PROM, 存储业务软件及参数的MRAM, 用于软件运行的DDR。CPU上挂载1553B控制器, 用于与综合电子设备进行遥测遥控数据的交互。CPU需要定时通过AX2000发送看门狗输入信号, 否则认为软件运行异常, 看门狗复位CPU。

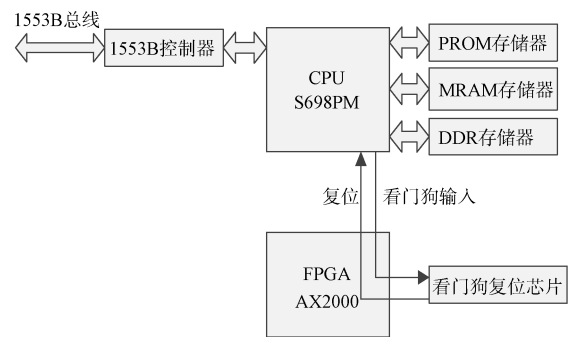


图1 软件运行环境图

Fig. 1 Running environment of software

1.2 任务与空间分配设计

引导监控软件的主要任务是将业务软件从MRAM搬移到DDR中并启动业务软件, 以及重构升级业务软件, 其主要运行在CPU核0上。因此, 需要提前明确分配到每个核的业务软件的内容, 并为它们合理分配存储空间与运行空间。

在某业务软件中, 需要完成的任务包括: 遥控接收任务、遥测发送任务、复杂计算处理任务、程序重构任务。其中, 复杂计算处理任务对实时性要求很高, 且需要持续占用CPU进行浮点计算。而其它任务对实时性要求没有那么多高, 仅需轮流使用CPU, 做简单的整型运算。则可将复杂计算处理任务编写为一个单独的业务软件, 让其在另一个核上运行; 将其它任务编写为另一个业务软件让其在另一个核上运行。业务软件任务与CPU多核的分配见图2所示。

整个系统的存储器包括固化引导监控软件的PROM、存储业务软件和业务数据的MRAM、运行业务软件的DDR, 如图3所示。

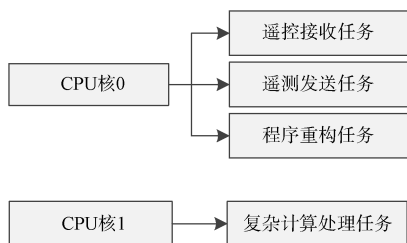
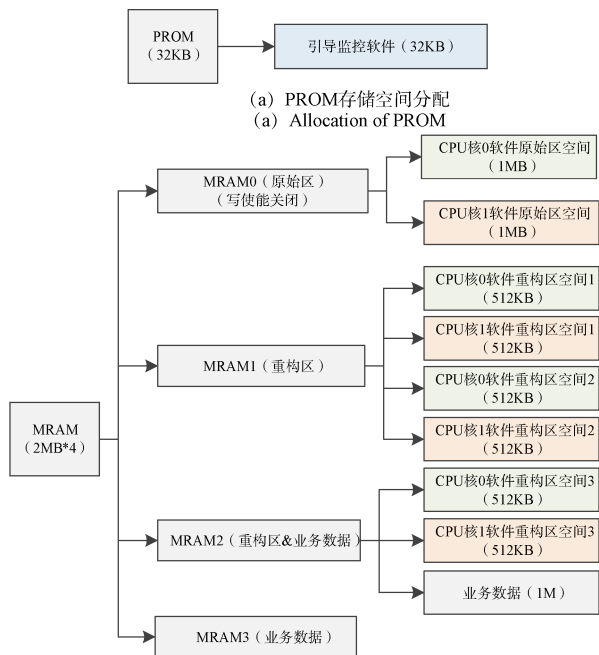
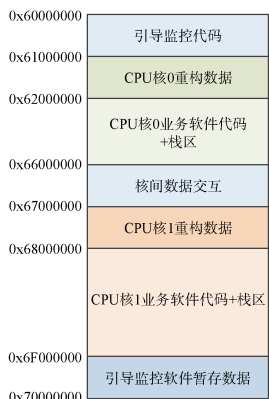


图 2 任务与内核分配

Fig. 2 Allocation of task and core



(b) MRAM存储空间分配
(b) Allocation of MRAM



(c) DDR运行空间分配
(c) Allocation of DDR

图 3 存储空间分配

Fig. 3 Allocation of storage space

由于引导监控软件固化在 PROM 上不可更改，PROM 里的数据不会被单粒子打翻，保证了引导监控软件的可靠性。

MRAM 中的业务软件分别存储在原始区和重构区。其中原始区的 MRAM 写使能被关闭，保证总有一份正确的业务软件可以运行；重构区的业务软件可以被改写、重构升级，需要三模存储，并在使用时进行三模校验，以保证业务软件的可靠性。MRAM 的其他空间被用于存储业务数据。

DDR 同时运行两个不同的业务软件，需要进行合理的分配，才能够使两个程序在运行过程中不发生冲突。DDR 空间大致分配如图 3 所示。具体运行时的 DDR 使用情况将在 2.3 章节进行介绍。当使用更多的内核或者业务软件需求的运行空间大小发生变化时，DDR 的分配也应该作出适当的调整以满足业务要求。

1.3 启动引导流程设计

引导监控软件的启动流程如图 4 所示。

当代码被编译成可执行文件(exe 类型文件)之后，首先，使用 mkprom2(一款工具程序)将其编成一个可以从 PROM 启动并自动加载到 DDR 运行的镜像文件。随后，将该镜像文件使用 objcopy 编成 bin 文件，固化到 PROM 中。当芯片复位时，CPU 芯片就会从 PROM 的 0x0 地址开始执行，完成 CPU 必要寄存器和 DDR 初始化，把引导监控软件加载至首地址为 0x60000000 的 DDR 区域并运行之。图 4(a)中表明了 mkprom2 增加的内容(包括初始化 DDR 和 CPU 必要寄存器等)，这些工作只会完成一次。当 CPU 芯片复位，完成最基础的初始化之后，只有 CPU 核 0 为启动状态，因此引导监控软件将首先运行在 CPU 核 0 上。

在引导监控软件中，由 CPU 核 0 完成的工作见图 4(b)。当 CPU 核 0 从 0x60000000 开始运行后，引导监控软件首先需要运行两个内核的公用部分代码，即通过特定寄存器判断自己是 CPU 核 0 还是 CPU 核 1。此时若判断结果为核 0，则将自己的 SP、FP 指针设置到 0x66000000 处，再跳转到引导监控软件的主任务函数。进入主任务函数后，引导监控软件将进入 30 s 倒计时，在倒计时期间完成接收指令、上报遥测、周期发送看门狗输入信号、重构等任务。当 30 s 倒计时结束之后，将 CPU 核 0 的程序搬移到首地址为 0x62000000 的空间，将 CPU 核 1 的程序搬移到首地址为 0x68000000 的空间。随后引导监控软件通过多处理器状态寄存器启动 CPU 核 1。最后 CPU 核 0 将 PC 指针跳转到 0x62000000 开始运行 CPU 核 0 的业务软件。

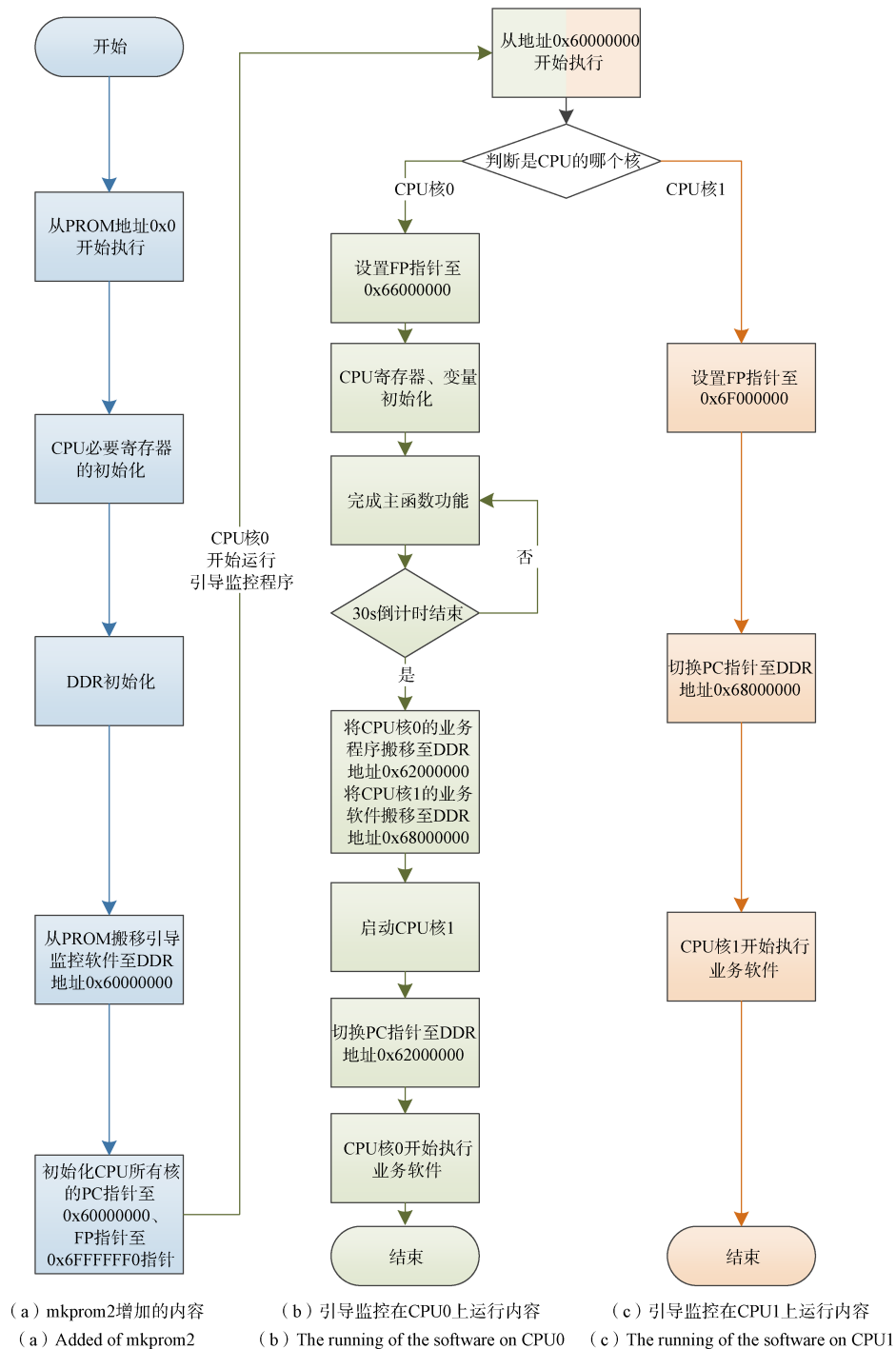


图4 引导监控软件的启动流程

Fig. 4 Startup process of boot and monitor software

CPU核1的操作如图4(c)所示。当CPU核1被CPU核0通过寄存器启动之后, PC指针将从mkprom2最开始初始化好的0x60000000开始运行引导监控公用的部分(通过特定寄存器判断自己是CPU核0还是CPU核1), 此时为CPU核1。因此将SP、FP设置到0x6F000000处, 再将PC指针跳转

到0x68000000处, 直接开始执行CPU核1的业务软件。

1.4 多核之间的数据交互设计

引导监控软件的遥测遥控功能都是运行在CPU核0上的, 但若CPU核0需要告知核1的PC跳转地址需要改变, 是不能通过给软件局部变量

赋值实现的。因此则需要对CPU核之间的数据交换进行设计,保证读写正确。

CPU核间交互涉及公共区域读写冲突问题。由于CPU的两个核也无法知道对方何时会读写公共区域的数据,无法控制顺序,因此仿照操作系统中的信号量操作,也为公共资源加入锁,即一个地址中的数据每一位均为0则为开锁,每一位均为1则为关锁。

当一个CPU核想读写公共区域的数据,首先要判断锁是否“打开”,如果锁的状态为“打开”,则将锁的状态改为“关闭”再对公共区域进行读写。通常来说需要2条汇编语句,使用ldb(从内存中读取数据)语句读取锁的状态,如果是“开锁”则使用stb(向内存写入数据)语句关闭锁,若是“关闭”则不断读取锁的状态,尝试抢锁。

两个核不可同时获取数据总线与地址总线控制权,所以即使两个核想同时读取锁的状态,也会由处理器的总线仲裁器控制先后顺序。若总线仲裁器令两个内核先轮流完成读锁状态再安排两个内核抢锁,此时二者都会认为自己抢到锁并对公共区域进行操作而造成冲突,具体情况如图5(a)所示。

因此,为防止公共区域出现读写冲突,需要使用语句原子汇编语句ldstb,其可以完成从地址读取数据到寄存器,并将地址中的每一位置为1。整个读加写的操作不允许被中断、延迟陷阱或其他内核的读写打断。这样CPU核0读锁并关锁就不会被CPU核1打断。每次使用这个语句读取锁的状态,如果从地址中读出了0(即“打开”),相当于自动抢锁并关锁,可以继续对公共区域进行操作。如果读出来的值全为1(即“关闭”),即关锁状态,该语句操作也对其他内核无影响。如果抢锁后完成所有读写,只需要执行stb指令将锁打开,具体情况如图5(b)所示。

考虑到引导监控软件需要被存储到PROM中,而这种航天级别的存储器造价较高,容量较小,程序设计不能过于复杂。所以尽量保证在CPU核0上完成主要功能,仅在CPU核1上完成简单的跳转工作,以减少总线使用冲突、简化数据交互流程。在引导监控软件中,将业务软件从MRAM搬移到DDR后,开启CPU核1前,可一次性填好需要改变的PC指针。因此只要选取DDR中的某一地址,提前填写CPU核1的PC指针,再启动CPU核1,不使用上锁机制。CPU核1在启动后只需要读取一

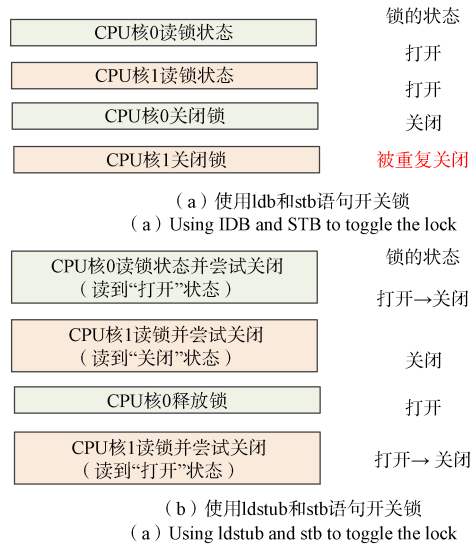


图5 不同汇编语句开关锁的情况

Fig. 5 Impact of different assembly instructions on locks

次该地址,获取自己的PC指针跳转地址,就可以跳转到新的程序进行执行。而图5中所述的复杂交互方式,适用于业务软件之间这种较为复杂情况下的交互。

1.5 监控功能设计

产品挂载1553B总线与综电(综合电子系统)进行数据通信,可新增遥控指令接收与遥测数据上报功能,从而实现了对运行中软件的监视与控制。

软件可接收的遥控指令均与启动和重构相关,例如:切换软件启动区域的指令,可以自由选择本次由原始区还是重构区启动;改变引导软件地址的指令,可以不使用三模功能启动重构区的软件;重构相关指令,在原始区业务软件也无法启动时,完成对重构区业务软件的更新升级。

通过1553B总线下传的遥测数据也与启动与重构相关,例如:引导监控软件运行的状态、时间,收到的指令数量,CPU核0与核1本次是从原始区启动还是重构区启动,重构包的计数,等等。

接收的遥控指令如表1所示,上报的遥测数据如表2所示。

2 测试与分析

2.1 测试环境

地面测试环境如图6所示。产品和地检之间使用1553B总线进行连接,地检通过网线与PC机进行连接。地检作为BC,产品作为RT。在PC机的软件中配置指令,输入不同的指令控制地检通过

表 1 遥控指令
Table 1 Telecommand

序号	指令名称	作用
1.	软件切换指令	选择业务软件是从原始区启动,还是从重构区启动。
2.	指定地址启动指令	不从程序默认的原始区和重构区地址启动业务软件,而是从指令中获取新的软件存储MRAM首地址、软件运行DDR首地址与软件长度。
3.	重构指令	指令中包含重构数据存储的首地址、有效数据长度与重构数据,准备对MRAM重构区的业务软件进行升级。
4.	软件校验指令	指令中包含校验的首地址与长度,用于判断重构数据是否正确。
5.	烧写指令	当程序连续收到两条烧写指令时,才能够将DDR中的重构数据烧写到MRAM重构区中。

表 2 遥测内容
Table 2 Telemetry content

序号	遥测名称	作用
1.	引导监控软件状态	标明引导监控软件是处于待机状态、重构状态、烧写状态等。
2.	运行时间	如果遥测中运行时间呈持续增长趋势,代表1553B总线正常,且程序一直运行。
3.	CPU核0启动区域	CPU核0的业务软件,从原始区、重构区或者自定义区域启动。
4.	CPU核1启动区域	CPU核1的业务软件,从原始区、重构区或者自定义区域启动。
5.	正确指令计数	收到的正确指令计数。
6.	错误指令计数	收到的错误指令计数,包括包格式错误、校验错误、参数错误,等等。
7.	重构包计数	收到的重构包个数,用于判断重构时是否有丢包。
8.	重构校验值	用于判断重构数据是否正确。

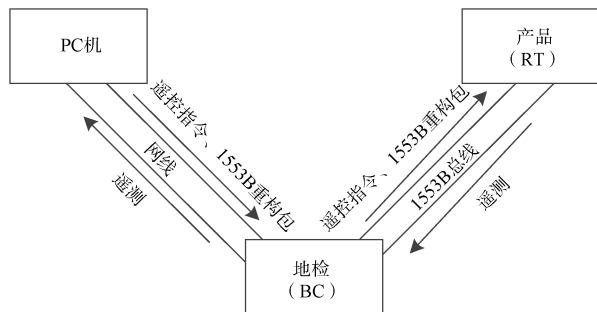


图 6 地面测试环境

Fig. 6 Environment of ground test

1553B 传输指令给产品。产品通过地检将遥测数据传输给 PC 机,以供判断软件状态的正确性。

航天器在轨测试时,产品将与综合电子系统通过 1553B 进行连接。此时综合电子系统作为 BC, 其将接收从地面测控站发来的指令并转发给 RT。

2.2 引导监控软件基础功能测试

通过使用上述的地面测试环境,对所有功能进行遍历,证明软件功能正确。测试结果如表 3 所示。

2.3 DDR 与 PC、FP 指针的变化分析

多核 CPU 运行软件重要的一点就是,多个软件的运行空间相互独立,互不干扰。在此对整个

启动流程中 DDR 的使用情况,各个程序占用空间的情况进行分析。

如图 7(a)所示为 mkprom2 完成初始化时的 DDR 空间使用情况。产品板上电后,CPU 自动从 PROM 中读取数据, mkprom2 会将引导监控软件的代码置于 0x60000000 的位置,将所有 CPU 核的 PC 指针置于 0x60000000, FP 指针置于 0x6FFFFFF0 的位置。

如图 7(b)所示为引导监控软件 30 s 倒计时的 DDR 空间使用情况, mkprom2 结束之后,会激活 CPU 核 0,此时会运行引导监控最开始的代码,将 CPU 核 0 的 FP 指针置于 0x66000000,并以此为栈底, PC 指针跳转到引导监控的主函数。在设计引导监控软件时要注意,引导监控的代码、全局变量占用的空间不能覆盖重构数据的暂存区 (0x61000000~0x62000000); 引导监控的栈区不能占用未来搬移 CPU 核 0 的业务软件代码 (0x62000000 起) 的区域。

如图 7(c)所示引导监控软件倒计时结束后完成程序搬移的 DDR 空间使用情况,引导监控在倒计时结束后,将两个内核的业务软件引导搬移至指

表3 软件功能测试与结果

Table 3 Test and result

序号	测试内容	测试结果
1.	给产品加电后无任何操作。查看PC机上的软件是否收到遥测,遥测是否正确。	产品上电后,可以在PC机的软件上收到遥测。遥测数据中的运行时间持续增长。由此说明1553B总线通路正常,且程序正常运行。
2.	给产品加电后无任何操作。通过遥测查看是否能够成功搬移并启动业务软件。	引导监控软件运行结束后,在PC机的软件上收到业务软件的遥测。由此证明引导监控软件成功搬移并启动业务软件。
3.	在引导监控软件运行期间,发送软件切换指令,改变2个业务软件的启动区域。	引导监控软件运行时,正确收到指令,2个业务软件启动区域和指令参数一致。
4.	在引导监控软件运行期间,发送软件切换指令,令CPU核0与CPU核1均为原始区启动。再给程序发送指定地址启动指令,令MRAM地址为CPU核1为重构区1的地址,这样CPU核0通过共享DDR传递新的PC指针地址给CPU核1,从新的地址启动CPU核1的程序。 软件启动后断电加电,重新启动。	引导监控软件运行时,在PC机的软件上查看遥测,CPU核0为原始区启动,CPU核1的启动区域改为自定义区域。 业务软件启动后,根据遥测,CPU核0启动的是原始区,CPU核1启动的是重构区,二核之间传递数据正确,且正常启动。 重新启动后,二者启动皆是原始区软件。
5.	在引导监控软件运行期间,发送重构指令,将2个CPU内核的重构数据传输给引导监控软件。重构完成后发送校验指令,从遥测数据判读校验值。若正确则发送烧写指令固化数据。将产品断电再加电,发送软件切换指令使2个内核从重构区启动,以启动刚才重构升级的新程序。	引导监控软件运行时,在PC机的软件上查看遥测。重构数据时,遥测数据中重构包计数持续增长。 重构完毕发送校验指令后,遥测数据中重构校验值正确。 再次加电,业务软件从重构区启动,且版本号升级,软件功能正常。

定地址(若未经过指令修改则为0x62000000与0x68000000),并将CPU核1的PC指针地址存于引导监控软件的数据暂存区。

最后如图7(d)所示引导监控跳转多核业务软件后的DDR空间使用情况,引导监控软件搬移后激活CPU核1,随后CPU核0的PC指针跳转至0x62000000开始执行核0的业务软件;而CPU核1开始运行,将自己的FP指针置于0x6F000000,从引导监控软件数据暂存区读取PC指针进行跳转(若未经过指令修改则为0x68000000),开始执行核1的业务软件。之后引导监控软件数据暂存区可被业务软件占用,进行数据交互。

2.4 多核CPU程序运行时间分析

尽管S698PM具有多个CPU内核,但其数据线和地址线只有一组。当多个内核想从DDR或者外设读取代码或者数据时,存在总线竞争,由芯片的总线裁决确定各内核的总线使用顺序。尽管打开了一级cache对数据和指令进行缓存以减少对总线的使用,在实际使用过程中,还是会受到总线竞争的影响。

为此,本文设计了一个实验来验证多核CPU在运行时的时间耗费情况。由于原程序本身是周期性事务,在不运行时使用sleep函数释放CPU资

源,不便于计算程序运行时间,因此将两个核的程序主要工作抽象为两个测试程序。

CPU核0主要完成各类周期性IO事务与简单计算,故而将其抽象为20次外设交互、100次加减运算、25次乘除运算、100次数据赋值操作的测试程序0。CPU核1主要完成大量的计算操作,故将其抽象为100次整数加减乘除、100次浮点数加减乘除、20次三角函数计算、50次数据赋值操作的测试程序1,使用CPU硬浮点功能加速计算。测试结果见图8所示。

从图8中可知,在双核情况下,一个测试程序运行的时间约为原来的二倍。由此分析可得,尽管使用了多核CPU让两个程序同时运行,但是在地址总线与数据总线采用共用设计时,程序存在总线竞争的情况,并且该款芯片的策略是每个核轮流使用总线,每个核优先级相同,不存在主从关系。虽然测试程序的运行时间变长了,但是在两个程序优先级相同且轮流运行的情况下,从上一次程序结束到运行本次程序结束的期望时间一致。

因此在多核的情况下,如果各个内核的软件运行时间可以符合时间及性能要求,则对软件进行合理的拆分并分给不同部门的人员负责,可以

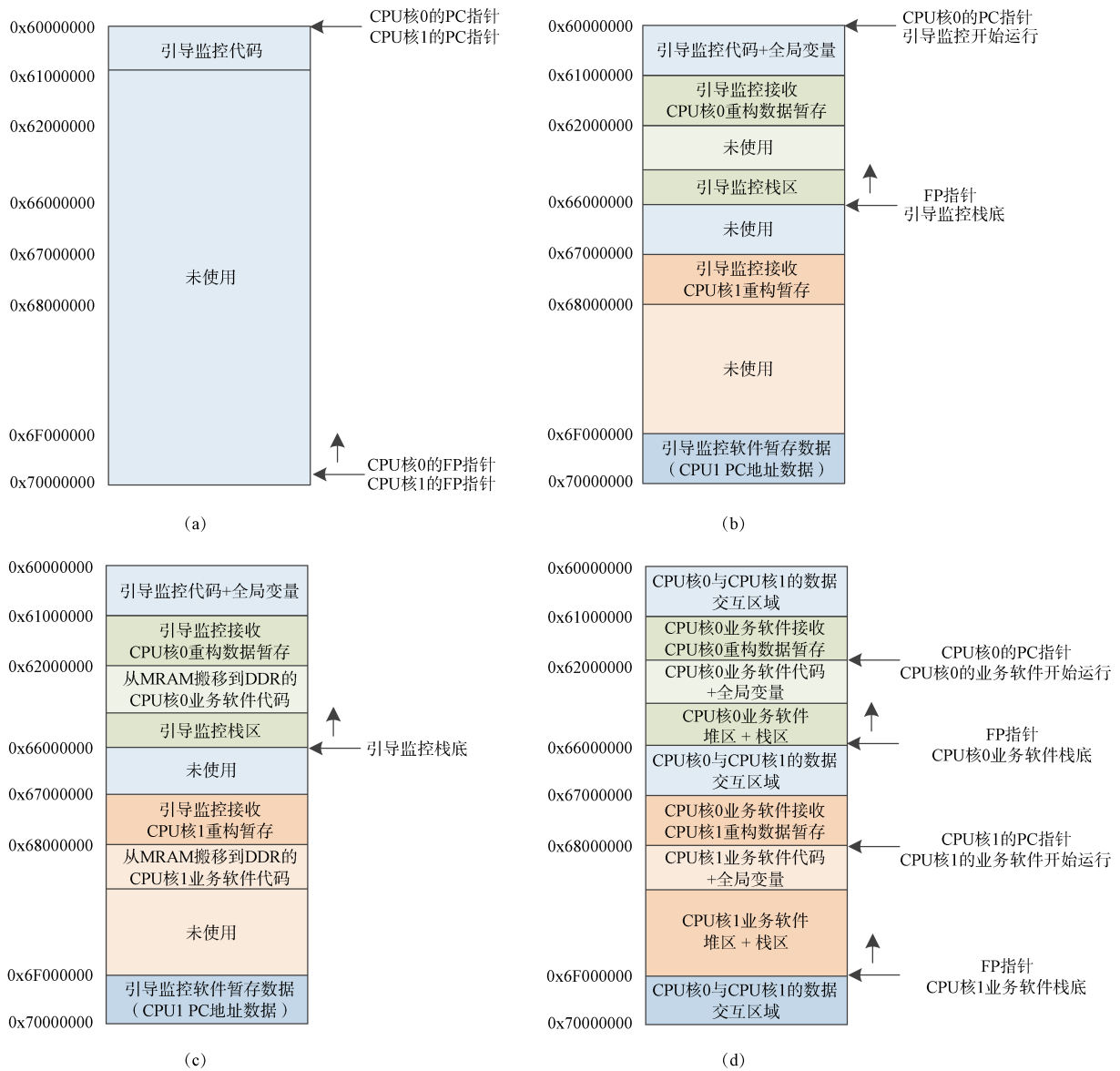


图 7 软件运行时的 DDR 占用情况
Fig. 7 DDR usage during software execution

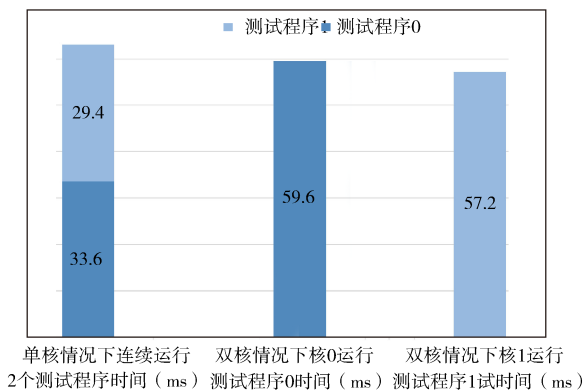


图 8 多核运行时间测试结果
Fig. 8 Test result of multi-core processor

减少软件合并时的错误以及跨部门管理的难度。

3 结束语

本文基于 S698PM 芯片设计了一个双核启动的引导监控软件, 该软件可以成功启动两个内核的业务程序, 并完成两个内核业务程序的重构。该软件使用一个内核完成所有软件的引导与重构工作, 减少了内核对总线的竞争。通过地面测试与在轨实验表明, 该引导监控软件运行情况良好。同时本文通过实验进一步测试了该芯片在多核运行时对程序运行时间的影响, 并给出了使用建议。

参考文献

- [1] 代雅晴, 韩晨阳, 王泽波, 等. 一种高速可靠的大体量星载软件重构方案设计[J]. 遥测遥控, 2023, 44(6): 19-28.
DAI Yaqing, HAN Chenyang, Wang Zebo, et al. Design of a reliable and high-speed refactor scheme for large-scale onboard software [J]. Journal of Telemetry, Tracking and Command, 2023, 44(6): 19-28.
- [2] 陈莹, 黄永彪, 谢瑾. 基于可靠性的多核系统硬实时任务并行调度[J]. 控制工程, 2021, 28(1): 176-182.
CHEN Ying, HUANG Yongbiao, XIE Jin. Reliability-based parallel scheduling of hard real-time tasks for multi-core systems[J]. Control Engineering of China, 2021, 28(1): 176-182.
- [3] 何瑞琦, 张凯龙, 吴金飞, 等. 基于多核异构操作系统的动态冗余可靠机制研究[J]. 计算机科学, 2025, 52(4): 33-39.
HE Ruiqi, ZHANG Kailong, WU Jinfei, et al. Research on dynamic redundancy reliability mechanisms based on multi-core heterogeneous operating systems[J]. Computer Science, 2025, 52(4): 33-39.
- [4] 徐双国, 刘云. 基于核间中断实现的嵌入式多核系统通信方法[J]. 舰船电子工程, 2021, 41(5): 58-61.
XU Shuangguo, LIU Yun. Embedded multi-core system communication method based on inter-processor interrupt [J]. Ship Electronic Engineering, 2021, 41(5): 58-61.
- [5] 张海涛, 胡胜, 仇林至. 基于AUTOSAR的SOME IP通信及其多核应用的实现[J]. 上海汽车, 2021(1): 17-22.
- [6] 董华玉. 浅议基于多核CPU的实时信号处理[J]. 科技与创新, 2025, 8: 127-130.
- [7] 安鑫, 康安, 夏近伟, 等. 基于机器学习的异构感知多核调度方法[J]. 计算机应用, 2020, 40(10): 3081-3087.
AN Xin, KANG An, XIA Jinwei, et al. Heterogeneous sensing multi-core scheduling method based on machine learning[J]. Journal of Computer Applications, 2020, 40(10): 3081-3087.
- [8] 刘涛, 习金玉, 宋站锋, 等. 基于多核并行计算的永磁同步电机有限集模型预测控制策略[J]. 电工技术学报, 2021, 36(1): 107-119.
LIU Tao, XI Jinyu, SONG Zhanfeng, et al. Finite control set model predictive control of permanent magnet synchronous motor based on multi-core parallel computing [J]. Transactions of China Electrotechnical Society, 2021, 36(1): 107-119.
- [9] WANG S, CUOMO S, MEI G, et al. Efficient method for identifying influential vertices in dynamic networks using the strategy of local detection and updating[J]. Future Generation Computer Systems, 2019, 91: 10-24.
- [10] DABAH A, BENDJOUDI A, AITZAI A, et al. Hybrid multi-core CPU and GPU-based B&B approaches for the blocking job shop scheduling problem[J]. Journal of Parallel and Distributed Computing, 2018, 117: 73-86.
- [11] SONI V, HADJADJ A, ROUSSEL O, et al. Parallel multi-core and multi-processor methods on point-value multiresolution algorithms for hyperbolic conservation laws[J]. Journal of Parallel and Distributed Computing, 2019, 123: 192-203.
- [12] 乔丹. 红外和高清双路视频处理传输平台的设计与实现[D]. 西安: 西安石油大学, 2021.
- [13] 曾鹏程, 王康谊, 刘维, 等. 多核异构处理器的多类型数据车载记录仪设计[J]. 单片机与嵌入式系统应用, 2020, 20(1): 75-78.
ZENG Pengcheng, WANG Kangyi, LIU Wei, et al. Design of multi-type data vehicle recorder based on multi-core heterogeneous processor[J]. Microcontrollers & Embedded Systems, 2020, 20(1): 75-78.
- [14] 文化, 张田剑南. 基于人工智能的嵌入式图像识别信息采集系统[J]. 信息技术, 2021(7): 117-118.
WEN Hua, ZHANG Tianjiannan. Embedded image recognition information acquisition system based on artificial intelligence[J]. Information Technology, 2021(7): 117-118.
- [15] 苏灏杨, 夏伟杰, 吴雪, 等. 基于多核CPU的雷达引导头数字化实时仿真研究[J]. 遥测遥控, 2025, 46(2): 92-99.
SU Haoyang, XIA Weijie, WU Xue, et al. Research on digital real-time simulation of multi-core CPU-based radar seeker[J]. Journal of Telemetry, Tracking and Command, 2025, 46(2): 92-99.
- [16] 郭少哲. 面向多核CPU的高性能位图索引算法设计与实现[D]. 南京: 南京邮电大学, 2023.
- [17] 刘楠, 杨东华, 贾逸凡, 等. 基于TMS320C6678 DSP的二级引导多核应用的研究与实现[J]. 软件, 2025, 46(5): 121-123.
LIU Nan, YANG Donghua, JIA Yifan, et al. Research and implementation of secondary bootloader multi-core application based on TMS320C6678 DSP[J]. Software, 2025, 46(5): 121-123.

- 2025, 46(5): 121-123.
- [18] 刘慧, 林海虹, 刘智. 多核DSP的BootLoader程序的实现[J]. 电子技术应用, 2003(6): 72-75.
- [19] 方先康. 多核多线程处理器环境下软件系统的设计与实现[D]. 北京: 北京邮电大学, 2008.
- [20] 王春雷. 基于多核DSP的专用嵌入式系统设计与实现[D]. 成都: 电子科技大学, 2017.
- [21] 邓豹. TMS320C6678多核DSP的加载配置和实现方法[J]. 航空计算技术, 2017, 47(1): 107-111.
DENG Bao. Bootloader configuration and implement based on multicore DSP TMS320C6678[J]. Aeronautical Computing Technique, 2017, 47(1): 107-111.
- [22] 曾定立, 丁国杰. 面向基带处理的异构多核架构软硬件平台设计[J]. 单片机与嵌入式系统应用, 2017(3): 13-17.
ZENG Dingli, DING Guojie. Heterogeneous multicore architecture software and hardware platform for baseband processing[J]. Microcontrollers & Embedded Systems, 2017(3): 13-17.
- [23] 熊浩伦, 闫国瑞, 李国军, 等. 基于最小系统的小卫星在轨软件重构系统设计[J]. 遥测遥控, 2020, 41(3): 48-55.
XIONG Haolun, YAN Guorui, LI Guojun, et al. Design of on-orbit software reconfiguration system of small satellite based on minimum system[J]. Journal of Telemetry, Tracking and Command, 2020, 41(3): 48-55.
- [24] 张衡, 顾泽凌, 杨明远, 等. 基于1553B的DSP在轨编程设计与实现[J]. 制导与引信, 2020, 41(4): 33-37, 42.
ZHANG Heng, GU Zeling, YANG Mingyuan, et al. Design and realization of on-board programming for DSP based on 1553B[J]. Guidance & Fuze, 2020, 41(4): 33-37, 42.
- [25] 丁磊, 刘海峰, 张小林. 一种星载软件无线电系统的高可靠智能重构方法[J]. 航天电子对抗, 2018, 34(4): 24-28, 33.
DING Lei, LIU Haifeng, ZHANG Xiaolin. A high reliability intelligent reconfiguration method for spaceborne software radio systems[J]. Aerospace Electronic Warfare, 2018, 34(4): 24-28, 33.
- [26] 李吉锋, 杨刚, 王戈, 等. 星载环境FPGA软件在轨重加载的一种方法[J]. 现代电子技术, 2012, 35(9): 87-90.
LI Jifeng, YANG Gang, WANG Ge, et al. Method of on-orbit reloading for FPGA software in satelliteborne environment[J]. Modern Electronics Technique, 2012, 35(9): 87-90.
- [27] 庞波, 郝维宁, 张文峰, 等. 一种SRAM-FPGA在轨重构的工程实现方案[J]. 航天器工程, 2017, 26(5): 51-56.
PANG Bo, HAO Weining, ZHANG Wenfeng, et al. Scheme of SRAM-FPGA on-orbit reconfiguration[J]. Spacecraft Engineering, 2017, 26(5): 51-56.
- [28] 胡萍, 刘世佳, 马宏业. 基于1553B总线的星载嵌入式软件在轨升级方法[J]. 遥测遥控, 2022, 43(1): 66-69.
HU Ping, LIU Shijia, MA Hongye. Method of embedded onboard software upgrading based on 1553B bus[J]. Journal of Telemetry, Tracking and Command, 2022, 43(1): 66-69.

[作者简介]

- 代雅晴 1997年生, 硕士, 工程师。
王泽波 1999年生, 大专, 助理工程师。
赵欣 1993年生, 硕士, 工程师。
刘世佳 1991年生, 硕士, 工程师。
马宏业 1982年生, 硕士, 研究员。
刁文婷 1997年生, 硕士, 助理工程师。

(本文编辑: 潘三英)