

基于 HMP 架构 CPU 的民机多安全级别系统实现方法

赵羚钧, 晋良银

(中电科航空电子有限公司, 成都 610015)

[摘要] 提出了一种基于 HMP 异构多核 CPU 和资源域控制器的设计方案, 可部署在同一物理 CPU 上并满足 DO178C 标准的多安全级别软件运行要求。首先讨论了 HMP 异构多核 CPU 和资源域控制器的相关概念及作用, 接着介绍了多安全级别系统的实现方案, 最后在 NXP iMX6 系统平台上进行设计和实现。结果表明, 该方案可成功地实现在机载设备上将不同安全级别的软件部署在同一物理 CPU 上的需求, 方案具有灵活、成本低廉和功能可靠的优点。

[关键词] 多安全级别软件; 资源域控制器; 异构多核; 机载软件

[中图分类号] TP332

[文献标识码] A

[文章编号] 1006-141X(2020)04-0026-06

HMP CPU-Based Multi-Security Level Software Implementation Solution

ZHAO Ling-jun, JIN Liang-yin

(CETC AVIONICS CO.LTD, ChengDu 610015, China)

Abstract: A design scheme based on HMP heterogeneous multi-core CPU and resource domain controller is proposed, which can be deployed on the same physical CPU and meets the multi-security level software operation requirements of the DO178C standard. First, the related concepts and roles of HMP heterogeneous multi-core CPU and resource domain controller are discussed, then the implementation of multi-security software is introduced, and finally designs and implementation are carried out on the NXP iMX6 system platform. The results show that the solution can successfully implement the requirements of deploying software of different security levels on the same physical CPU on the airborne equipment. The solution has the advantages of flexibility, low cost and reliable functions.

Key words: multi-security software; resource domain controller; heterogeneous multi-core; airborne software

民用航空机载电子设备对适航安全性要求严格, 目前国际上普遍采用 DO-178C 标准对机载软件的适航安全等级进行分类和认证。不同适航安全级别软

件适航认证难度和复杂度不同。不同适航安全级别软件适航认证难度和复杂度不同。根据 DO-178C 规范, 机载设备软件由于安全级别的不同被划分为设

收稿日期: 2020-07-10

基金项目: 四川省重大科技专项课题 (基金编号 2018GZDZX0021)

引用格式: 赵羚钧, 晋良银. 基于 HMP 架构 CPU 的民机多安全级别系统实现方法 [J]. 航空电子技术, 2020, 51(4): 26-31.

计保障等级(DAL: Design assurance Level)E、DAL D、DAL C、DAL B、DAL A 五种安全等级^[1],从DAL E到DAL A 软件安全级别依次递增;其中DAL A 是最高安全级别,该安全级别软件出现异常将可能导致飞机出现灾难性的事故,例如飞机坠毁。而DAL E 是最低安全级别,该安全级别软件出现异常将不会影响飞机的正常运行也不会增加飞行员的工作量。

目前不同安全级别软件通常运行在不同的CPU上或采用满足ARINC 653标准的分区操作系统上,以便让软件设计研发更容易满足DO 178C标准^[2-3]。这两种方法均有一定的缺点,ARINC 653分区操作系统存在着对Linux支持不好和价格昂贵的缺点,而部署在不同CPU上势必会增加设备体积、重量和功耗。在机载客舱系统中开源免费的Linux操作系统是目前最广泛的选择,使用Linux操作系统的客舱设备通常设计为DAL E级别。但是部分客舱设备存在大多数需求只需要满足DAL E,而极少功能需要设计为DAL D或更高级别^[4]。由于ARINC 653分区操作系统对Linux的不友好支持特性,我们尽量避免在此情况下使用。针对上述功能特点,目前的解决方案一是将功能分别部署到两个设备上分别使用独立的CPU,二是将Linux系统认证为DAL D或更高级别的安全等级。Linux系统具有上千万行代码,并且客舱设备功能偏消费娱乐,系统升级快,因此对Linux进行适航认证将是一件非常困难的事情,因此目前通用的解决方法是部署到不同的设备上。

如能将低安全级别软件与高安全级别软件部署在同一设备的同一物理CPU上,显然具有成本、功耗和体积优势。本文提供了一种新的解决方案,该方案首先选用具有异构多核处理器(HMP: Heterogeneous Multi-Processing)和资源域控制器(RDC: Resource Domain Controller)的CPU芯片,然后对机载设备上的外围设备控制器以CPU核为中心进行安全域划分。将不同安全等级的功能所使用的硬件资源在逻辑上相互独立和隔离,然后将不同安全级别软件放在不同的逻辑域中运行,方案满足符合DO 178C适航标准,最后给出了软件的设计过程。

1 异构多核处理器

一般桌面PC采用多核处理器,每个核的处理能

力是一样的,并且多核之间共享L2/L3缓存(Cache)。HMP的概念最早出现在ARM架构中。ARM架构在解决手持设备性能和省电的这两个矛盾时,引入了big.LITTLE处理器架构。big是指CPU的大核,LITTLE是指小核,big的处理能力比LITTLE的更强,它们集成在同一颗物理CPU上。当手持设备需要高性能工作时使用大核,不需要高性能时使用功耗较小的LITTLE小核进行工作,从而达到性能和功耗的相对平衡。由于多核处理器每个核的能力是一样的,这种架构就被称之为HMP^[5-6]。

HMP架构的最终实现形式取决于芯片的供应商,并在实际中产生了一定的演化,如面向手机的CPU采用Cortex A53作为大核,Cortex A7作为小核,而NXP(恩智浦)针对汽车使用环境推出了iMX6 SOLOX CPU,它的Cortex A9作为大核,Cortex M4作为高可靠执行环境的小核。Cortex A系列是ARM的高端系列,其主要使用在高性能应用场景,Cortex M系列是低端系列面向工业控制等。在iMX6 SOLOX的这款CPU中,Cortex A9主频为1GHz,Cortex M4为200MHz,Cortex M4主要用做控制用。iMX6 SOLOX的Cortex A7和Cortex M4这两个核都分别拥有自己完全独立的Cache和取址总线,因此相比于共享L2/L3 Cache的多核CPU来说,它的大小核之间拥有完全隔离的计算资源,不会出现因为共享L2/L3 Cache而带来的执行不确定性。避免多核CPU间共享Cache带来的执行不确定性,是开发满足DO 178C标准的多安全级别软件的一个重要考虑,也是ARINC 653分区操作系统实现的一个难点。采用ARM中这种特有的HMP架构,大小核拥有完全隔离的计算资源,从源头上规避了多核间执行不确定性这个问题,避免了在DO 178C认证过程中带来的挑战。如果设备只有两个安全级别功能,把它们分别部署到大小核上,使用HMP架构的CPU具有明显的成本低和设计灵活的优势。

2 资源域控制器

RDC资源域控制器是NXP公司IMX6/IMX7/IMX8系列部分CPU上特有的一个SoC片上控制器。虽然采用HMP架构的CPU大小核拥有完全隔离的计算资源,但在实际使用中,只有计算功能远

远不够, 必需配合相应的外设控制器来完成。如网络通信需要网卡控制器, 串口通信需要通用异步收发传输器 (UART: Universal Asynchronous Receiver/Transmitter) 等。因此要实现大小核执行环境的完全逻辑性隔离, 还要一个可以对 CPU 外设进行按需隔离的机制。NXP 公司设计 RDC 资源域控制器的初衷是提高处理器性能和灵活性, 因为在某些情况下, 每个内核运行不同功能的软件, 每个内核上的软件可以由不同的提供商开发。为了向不同的软件提供商提供互不影响和受控通信的软件运行环境, RDC 应运而生。

RDC 将 CPU 上所有的控制器 (如网口控制器, UART, GPIO) 和大小处理器核心都看成一个独立的逻辑资源项目, 由用户按需对这些逻辑资源项目以 CPU 核为中心进行组合划分, 组合的结果称为访问域, 不同域之间不能相互访问。同时出于效率和灵活原因, 大小核可各自独占外设资源也可共享芯片资源。在独立的处理域之间共享芯片资源可能造成数据冲突, 例如其中一个内核上的进程存储在外围设备或内存中的信息可能会被另一内核上运行的软件覆盖。如果两个操作系统之间没有良好的同步机制, 可能会导致意外的故障或性能下降。针对不同内核之间资源冲突的问题, RDC 提供了一套完整的机制, 该机制允许系统在启动时配置 RDC, 将内核、总线主机、外围设备和内存区域等资源分配给域标识符以建立资源域。配置完成后, 系统将监视总线事务, 以将内核和总线主控器发起的访问限制于每个域各自的外围设备和内存中。对于需要共享的外围设备, RDC 提供了基于信号量的硬件锁定机制, 使得在域中的软件使用外围设备时提供暂时的排他性。一旦某个域的软件完成了任务, 可能会释放硬件信号量使外围设备可用于另一域, 否则另一个域的访问会被 RDC 所阻塞。

利用 RDC 对多安全级别功能所需的外围设备控制器以大小核为中心进行域划分, 可以实现不同域上的硬件资源在逻辑上是相互独立相互隔离的, 对部分需要共享的硬件资源可以配置为低安全级功能硬件访问属性为只读属性, 从硬件上完全避免不同域软件运行期间的相互影响^[7]。RDC 带来的不同的域上软件运行互不影响的新特性, 是解决多种安全级别软件在同一物理 CPU 上运行的关键方法和新手

段。

3 系统设计

根据民机硬件适航标准 DO 254 标准, 本系统设计不选用具有可编程功能的芯片, 硬件方案按所需功能中最高等级来设计, 同时硬件设计上对不同安全等级的功能所需的外设进行隔离设计 (如串口电平转化芯片, CAN 总线), 保证它们相互独立。这样的硬件设计可最大程度降低硬件适航的难度和工作量。目前同时集成 RDC 资源域控制器和 HMP 架构的 CPU 芯片为 NXP 公司 IMX6/IMX7/IMX8 系列的部分 CPU, 本系统设计选用 iMX6 SoloX^[8-9]。

为了实现将不同安全级别的功能部署到同一 CPU 上, 需充分考虑 RDC 的资源域区分粒度和软件如何部署的问题。本方案针对客舱设备典型的具有两个安全等级功能进行示意设计。

图 1 是多安全级别机载设备硬件架构图, 该系统由异构多核 CPU、电源、RAM、EMMC、nor flash、开关设备、串口以及 CAN 接口等设备组成。在该方案中详细构成如下:

低安全级别系统硬件组成: 由大核 (ARM Cortex-A 系列 Core)、外部 RAM、EMMC、串口 (1)、网络接口 (1) 等设备组成的子系统用于运行低安全级别的 Linux 操作系统, 该子系统主要提供 Linux 系统硬件运行环境;

高安全级别系统硬件组成: 小核 (Cortex-M4 Core)、nor flash、串口 (0)、CAN 接口 (0) 等设备组成的子系统用于运行高安全级别的控制系统, 使用的 RAM 为内部 Cortex-M4 核自带 TCAM 内存。该系统主要用于接收并执行高安全级系统的功能, 例如 Cortex-M4 Core 核解析控制指令后通过 GPIO 控制飞机一些重要开关的闭合。

其中电源模块为共用, 两个系统的受电控制 (包括外部按键复位) 由外部硬件设计来保证相互隔离。

4 软件设计

图 2 是基于 HMP 架构 CPU 的民机多安全级别软件上电加载示意图, 系统加电后 CPU 的 ARM Cortex-A 系列核开始运行, 此时 ARM Cortex-M 核处于复位态未运行。Cortex-A 核完成系统部分设备

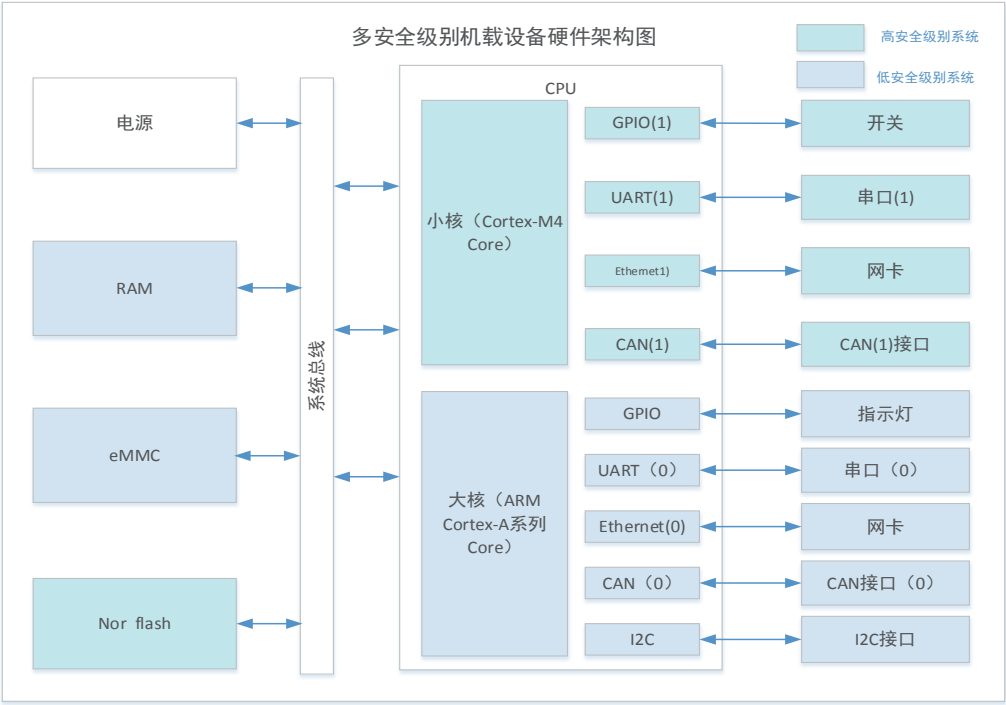


图 1 多安全级别机载设备硬件架构图

和寄存器初始化后开始执行系统引导程序。系统引导程序中重要的工作是设置资源域控制器，对各个资源域中的硬件资源进行按域分配与隔离。图 3 是基于系统设计中所要求的系统功能对 CPU 片上外设的逻辑划分，通过配置 RDC 寄存器对外设以内核为中心进行域划分；将 ARM Cortex-A 核设置为域 0，将 ARM Cortex-M4 设置为域 1。配置 RDC 相应寄存器设置 UART(1)，Ethernet(1) 以及部分 GPIO 的域为 1。同理设置 UART(0)，Ethernet(0)，外部 RAM 为设备域为 0。对系统资源进行资源域划分之后，ARM Cortex-A 系列核只能访问域 0 的设备及资源，而无法访问属于域 1 的 UART(1)，Ethernet(1) 等设备；同时，ARM Cortex-M4 核只能访问属于域 1 的设备及资源，而无法访问属于域 0 的设备。

RDC 设置及系统初始化完成后，系统引导程序分别启动资源域 1 中的高安全软件与资源域 0 中的低安全级软件。此后高安全级与低安全级软件同时 在同一物理 CPU 上运行，但资源域 0 与资源 1 中的资源相互独立，无法相互访问。其中 RDC 设置是划分高安全级域与低安全级域的关键点，其设置程序必须包含在高安全级别软件范围内并与其一同取证，而 RDC 设置工作仅从技术上来讲可放在如图 2 中的 ARM Cortex-A 或 ARM Cortex-M4 任意核中执行，

因此实际工程中可从降低适航工作量出发进行权衡选择。

由于系统上电的引导程序只能运行在 ARM Cortex-A 核上，因此 DO 178C 软件认证包括两部分工作，一个是认证运行在 Cortex-M 的高安全级别软件，另一个是运行在 Cortex-A 上的引导程序。高安全级别软件的认证工作是必选的，而运行在 Cortex-A 上的引导程序因为它需要加载高安全级别软件需要被动认证。这会带来额外的适航认证工作量，但是引导程序本身体积很小，而且功能固定，相比于把低安全级程序整体认证为高等级来说，这个工作量可忽略不计。另外，在 Cortex-M 核中运行高安全级别系统中可以使用安全认证操作系统，但符合 DO 178C 软件认证的操作系统必须针对具体芯片型号进行适航取证，目前未见针对该款芯片满足适航认证的操作系统。因此在单任务处理不能满足系统需求，必须在 Cortex-M 核中使用操作系统时，需要用户进行相应的开发和适航取证工作。

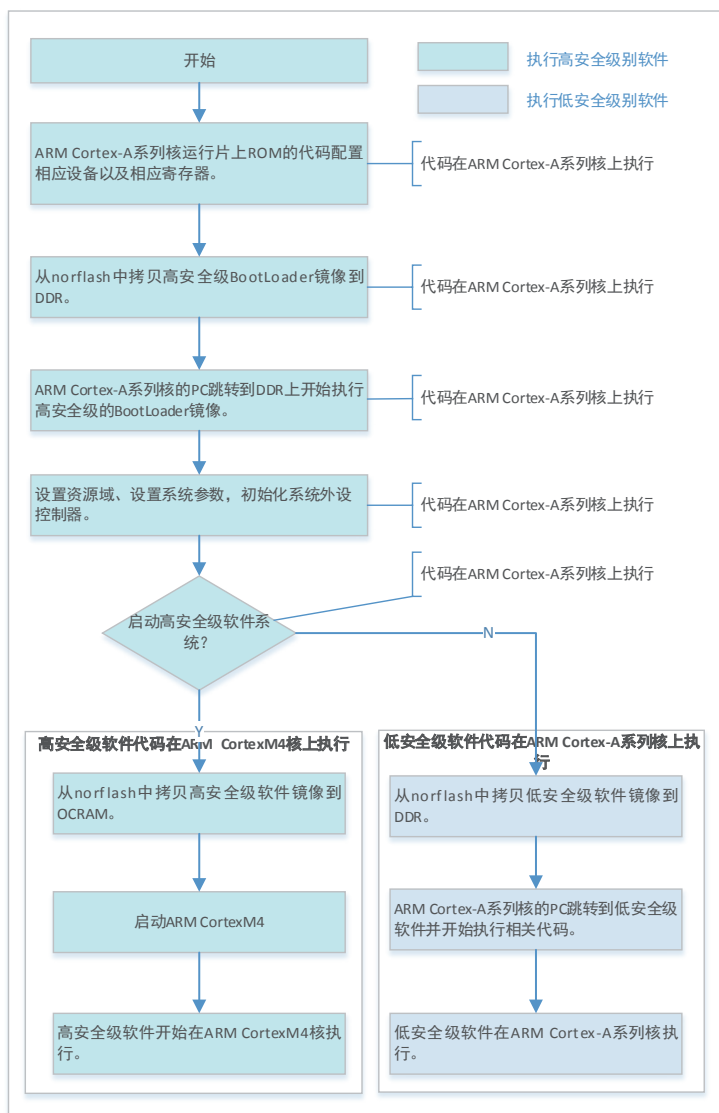


图 2 基于 HMP 架构 CPU 的民机多安全级别软件流程图

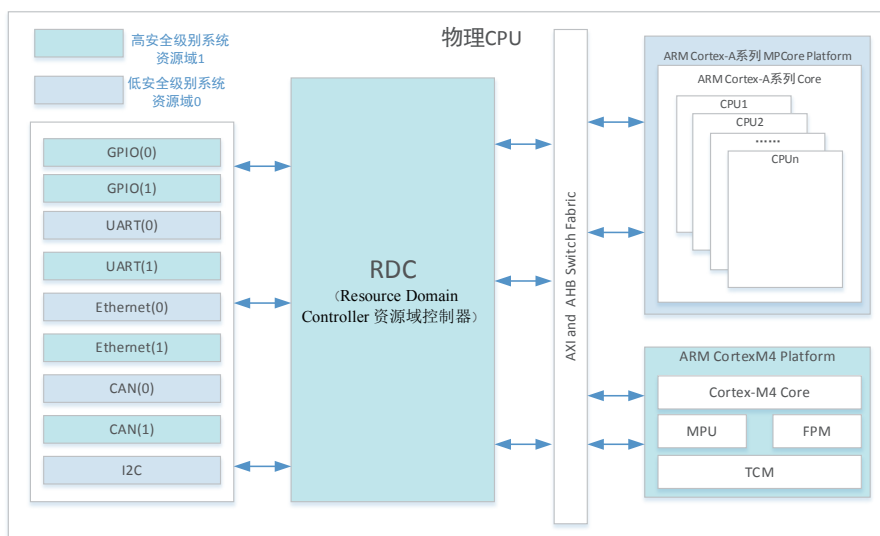


图 3 异构多核 CPU 片上硬件资源分配示意图

5 结束语

本文介绍了民机航电多安全级系统设计的现状, 讨论了 HMP 与 RDC 的相关概念和作用。通过分析现有实现方案存在的问题和缺点, 提出了一种基于 HMP 架构 CPU 的民机多安全级别系统实现方法。该方案利用微处理器平台上资源域控制器对机载设备上的外围设备资源按资源域进行划分, 将不同安全等级的硬件资源在逻辑上实现了隔离; 然后将不同安全级别软件放在不同的资源域中运行。该方案符合 DO 178C 标准对软件安全等级的要求, 同时实现了将多种安全级别软件部署在同一设备和同一物理 CPU 上; 该方案适用于需要将基于 Linux 系统的低安全级别软件与高安全级别软件放在同一物理 CPU 上运行的客舱机载设备。相较于现有方案无需使用庞大且昂贵的分区操作系统, 也无需将低安全级别软件认证为高安全等级, 有效地降低了机载设备软件适航难度与研制成本, 具有较高的实用价值。

参 考 文 献

- [1] RTCA/DO-178C. Software consideration in airborne systems and equipment certification [S]. USA: RTCA, Inc, 2012.
- [2] 黎明, 莫军. ARINC 653 分区操作系统多核处理器任务调度设计 [J]. 计算机系统应用, 2018, (5): 216-220.
- [3] 李丹丹, 包贵浩, 张宏志. 飞机机载软件符合性方法 DO 178B 的应用研究 [J]. 科学技术创新, 2018, (15): 75-76.
- [4] 邢亮, 卢伟. 机载适航标准 DO-178B/C 软件开发过程研究 [J]. 航空计算技术, 2016, 11(6): 74-79.
- [5] 潘皓. 多核虚拟化分区技术在航空电子系统中的应用 [J]. 计算机测量与控制, 2019, (11): 206-209.
- [6] Certification Authorities Software Team. Position Paper CAST-32A, Multi-core processor[S]. USA: Federal Aviation Administration, 2016.
- [7] 程都. 基于 ARM Cortex-M4 的数据采集系统的设计与研究 [D]. 南京: 东南大学, 2018.
- [8] 徐国根, 马晋. 机载嵌入式软件可靠性设计研究 [J]. 航空电子技术, 2014, (3): 37-41.
- [9] 施劲松, 步鑫. 民用机载航电系统安全性实现方法分析 [J]. 航空电子技术, 2014, (1): 52-56.