[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G09B 9/08 (2006.01)
G06F 9/455 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200810104682.5

[43] 公开日 2008 年 12 月 10 日

[11] 公开号 CN 101320524A

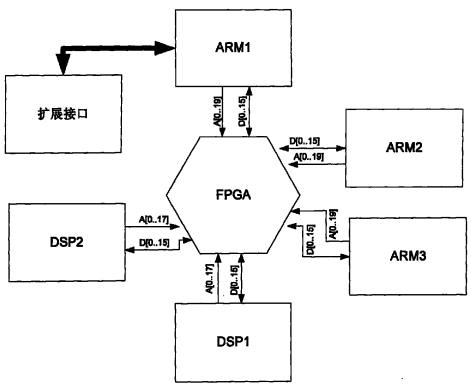
[22] 申请日 2008.4.22
[21] 申请号 200810104682.5
[71] 申请人 北京航空航天大学
地址 100083 北京市海淀区学院路 37 号
[72] 发明人 胡 迪 董云峰 高小波 郭 巍 王 皓 刘小林
[74] 专利代理机构 北京科迪生专利代理有限责任公司
代理人 贾玉忠 卢 纪
权利要求书 2 页 说明书 12 页 附图 9 页

[54] 发明名称

多处理器实时仿真平台

[57] 摘要

多处理器卫星实时仿真平台由三个或更多的 ARM 处理器和两个或更多的 DSP 处理器通过 FPGA 作为中间存储器组成，其中 ARM1 作为仿真卫星控制中心计算机；ARM2 用于仿真卫星所处的空间环境；ARM3 用于卫星的动力学计算；DSP1 完成敏感器模型的计算；DSP2 完成执行机构模型的计算；ARM1 向 DSP1 发出数据获取指令，DSP1 向 ARM2、ARM3 发出获取数据指令，从 FPGA 中读取到相关数据后即向 ARM1 传送数据，ARM1 根据所读取的数据进行计算获得相应的姿态轨道参数，如果参数满足要求，卫星即进入正常模式，如果不满足要求，ARM1 向 DSP2 发送控制指令，完成对执行机构的控制，DSP2 完成计算后通过 FPGA 向 ARM2、ARM3 输出相应数据，直到姿态轨道满足相应要求。 本发明简化了系统的复杂性，降低了仿真难度，提高了仿真的实时性。



1、多处理器卫星实时仿真平台，其特征在于：由三个或更多的 ARM 处理器和两个或更多的 DSP 处理器通过 FPGA 作为中间存储器组成，其中 ARM1 作为仿真卫星控制中心计算机，运行姿态轨道控制算法；ARM2 用于仿真卫星所处的空间环境；ARM3 用于卫星的动力学计算，完成对卫星姿态轨道参数计算；DSP1 主要是完成敏感器模型的计算；DSP2 主要是完成执行机构模型的计算；首先 ARM1 向 DSP1 发出数据获取指令，即 ARM1 要读取当前卫星敏感器数据，然后 DSP1 向 ARM2、ARM3 发出获取数据指令，即敏感器要知道当前卫星所处的空间环境和动力学状态，从 FPGA 中读取到相关数据后即向 ARM1 传送数据，ARM1 根据所读取的数据进行计算获得相应的姿态轨道参数，如果参数满足要求，卫星即进入正常模式，如果不满足要求，ARM1 向 DSP2 发送控制指令，完成对执行机构的控制，同时 DSP2 完成计算后通过 FPGA 向 ARM2、ARM3 输出相应数据，直到姿态轨道满足相应要求。

2、根据权利要求 1 所述的多处理器实时仿真平台，其特征在于：所述仿真平台还包括扩展接口，主要是便于所述的各处理器作功能扩展用，实现相应功能后通过扩展接口实现数据对外通信。

3、根据权利要求 1 或 2 所述的多处理器实时仿真平台，其特征在于：所述所有 ARM 处理器和所有的 DSP 处理器与 FPGA 的连接方式均采用硬件设计最简单的直连方式，即数据/地址/控制三总线形式。

4、根据权利要求 1 或 2 所述的多处理器实时仿真平台，其特征在于：所述的 FPGA 主要采用 FIFO 技术实现数据共享，其中一个 ARM 处理器将计算完成的数据通过地址/数据总线写入 FIFO，同时触发其他几个处理器，如果此时其他几个处理器需要相关数据则发出相关指令读取这些数据，每一个处理器均对应一个 FIFO，而每个 FIFO 又与其他几个处理器相连。

5、根据权利要求 1 或 2 所述的多处理器卫星实时仿真平台，其特征在于：所述的 ARM1 由单一芯片构成，其是整个卫星仿真平台的控制核心，其首先通过 FPGA 读取 DSP1 存放在 FIFO 中的数据，进行计算后，向 FPGA 输出相应的控制命令，同时通知 DSP2 读取相应数据，DSP2 读取相应命令后完成对执行机构的控制模拟。

6、根据权利要求 1 或 2 所述的多处理器卫星实时仿真平台，其特征在于：所述的 ARM2 由单一芯片构成，其主要完成空间环境计算，在接收到 ARM1 发出的运行指令后，ARM2 程序开始运行，同时将计算完成的数据存储在 FPGA 中相应的 FIFO 空间处；DSP1 在接到

ARM1 获取数据指令后，立即向 ARM2 发出数据获取指令，同时读取 ARM2 在 FPGA 内部存储好的数据；DSP1 同时也在 FPGA 中存储相关数据供 ARM3 读取。

7、根据权利要求 1 或 2 所述的多处理器卫星实时仿真平台，其特征在于：所述的 AMR3 主要完成卫星当前姿态轨道计算，动力学处理器等待 DSP1 通知其读取数据，即 ARM3 得到 DSP1 发出的读取指令后，从 FPGA 中所对应的 FIFO 处读取当前卫星敏感器的状态数据，同时进行计算卫星当前姿态及轨道，并将数据存储至 FPGA 中；DSP2 得到 ARM1 发出控制指令后，也进行模型计算，将结果存储至 FPGA 相应的 FIFO 中，同时通知 ARM3 读取当前数据，ARM3 接收到最新数据后计算，同时将结果存储至 FPGA 的 FIFO 中。

8、 根据权利要求 1 或 2 所述的多处理器卫星实时仿真平台，其特征在于：所述的 DSP1 是仿真敏感器模型，DSP1 首先得到 ARM1 发出读取指令，其从 FPGA 中读取由 ARM3 计算出的结果，DSP1 获得数据后进行相关计算，然后将计算结果存储到 FPGA 供 ARM1 读取。

9、 根据权利要求 1 或 2 所述的多处理器卫星实时仿真平台，其特征在于：所述的 DSP2 是仿真执行机构模型，DSP2 首先得到 ARM1 发出读取指令，其从 FPGA 中读取由 ARM3 计算出的结果，DSP2 获得数据后进行相关计算，然后将计算结果存储到 FPGA 供 ARM1、ARM3 读取。

多处理器实时仿真平台

技术领域

本发明一种用于卫星动态的仿真平台，特别是一种多处理器实时仿真平台。

背景技术

卫星动态仿真技术就是用做试验来进行科学研究的方法。卫星仿真分为数学仿真、半物理仿真、全物理仿真三种，其中卫星控制系统的仿真尤为重要。数学仿真就是在计算机上对系统的数学模型进行试验，卫星控制系统数学仿真通常与控制系统计算机辅助分析相结合，完成卫星轨道动力学和卫星姿态动力学的时域仿真及控制系统频域分析。

卫星数学仿真主要包括下列内容：（1）空间环境模型仿真包括地球引力场、磁场、空气动力、太阳辐射压力、日－月－地摄动等仿真；（2）卫星的轨道动力学及制导与导航系统仿真；（3）卫星的姿态动力学与控制系统仿真，包括三轴稳定卫星姿态确定与控制系统仿真；自旋卫星姿态确定与控制系统仿真；多刚体系统动力学分析及控制仿真；带晃动液体贮箱及大型挠性附件卫星的姿态确定与控制系统仿真。

数学仿真最关键的问题是建立能精确地反映系统性能的数学模型。建立系统数学模型有两种途径，即理论建模和试验建模。一个复杂的动力学系统建模常常需要两种方法并用。采用理论建立的数学模型不仅需要通过试验来取得有关参数，也有待于通过试验来验证和修正。复杂卫星的结构一般由一个主刚体、挠性附件（大型天线和太阳帆板）和带有晃动液体的燃料贮箱组成，系统高度复杂，很难得到精确的理论模型并求解，总是在一定的假设条件下简化，这就使得验证模型工作显得十分重要。通过测量卫星在轨运行参数来验证和修改卫星的数学模型是最为有效的方法，但难度颇大。

目前，对于卫星数字仿真通常采用的是 PC 机作核心处理机，同时采用 windows 操作系统。由于系统本身存在某些不确定因素及实时性不够，因此会造成整个仿真不确定. 如果采用 VxWorks 实时操作系统，同样仿真成本会增加。有时为了仿真相应的空间环境，都要动用几十台电脑同时工作十几天，这样一个方面造成资源浪费，另一方面花费时间太多；同时采用 PC 机实现卫星系统仿真，大多数是基于 PCI 总线卡，这样在数据传输上 PCI 总线的带宽及传输速率是整个控制系统的瓶颈，因此在这个基础上，提出多处理器实时仿真平台，对于卫星动态仿真，是一个非常必要的，而且是今后卫星发展的一个趋势，是必须的，同时集成 ARM 技术、DSP 技术及 FPGA 技术的各种优势的嵌入式系统，是卫星动态仿真中唯一能够突破 PC 机仿真瓶颈的技术途径，不仅可以大大提高卫星仿

真精度，同时也为今后卫星在处理器选型上提供可靠的依据和支持。

发明内容

本发明的技术解决问题：克服现有技术的不足，提供一种基于嵌入式高速共享存储器的多处理器实时仿真平台，它避免了在卫星设计前期的数字化仿真利用众多的计算机进行仿真，解决了模型库小型化，简化了系统的复杂性，降低了仿真难度，提高了仿真的实时性。

本发明的技术解决方案：多处理器卫星实时仿真平台，由三个或更多的 ARM 处理器和两个或更多的 DSP 处理器通过 FPGA 作为中间存储器组成，其中 ARM1 作为仿真卫星控制中心计算机，运行姿态轨道控制算法；ARM2 用于仿真卫星所处的空间环境；ARM3 用于卫星的动力学计算，完成对卫星姿态轨道参数计算；DSP1 主要是完成敏感器模型的计算；DSP2 主要是完成执行机构模型的计算；首先 ARM1 向 DSP1 发出数据获取指令，即 ARM1 要读取当前卫星敏感器数据，然后 DSP1 向 ARM2、ARM3 发出获取数据指令，即敏感器要知道当前卫星所处的空间环境和动力学状态，从 FPGA 中读取到相关数据后即向 ARM1 传送数据，ARM1 根据所读取的数据进行计算获得相应的姿态轨道参数，如果参数满足要求，卫星即进入正常模式，如果不满足要求，ARM1 向 DSP2 发送控制指令，完成对执行机构的控制，同时 DSP2 完成计算后通过 FPGA 向 ARM2、ARM3 输出相应数据，直到姿态轨道满足相应要求。

所述仿真平台还包括扩展接口，主要是便于所述的各处理器作功能扩展用，实现相应功能后通过扩展接口实现数据对外通信。

本发明与现有技术相比的优点在于：

（1）本发明采用 ARM 芯片主要是来替换当今卫星仿真所使用的计算机，由于 ARM 芯片主要完成姿态/轨道控制、动力学计算、空间环境计算，这个计算会使用相应的操作系统，因此采用 ARM 芯片可以更好地移植相应的程序，而不需要对已有的程序有太大的变动，同时目前 ARM 内核其主频已超过 1GHZ，对于星上的程序计算已经足够；采用 DSP 作为模型计算主要考虑其精度，在进行各个模型仿真时，建立的模型尽可能与真实卫星相应的部件一致，包括建立各部件的基本模型、误差模型、其他干扰模型以及故障模型等，因此主要着重于算法及计算，DSP 芯片其最主要的优点就是具有强大的计算能力，因此可以胜任对各模型的计算仿真；FPGA 技术主要是完成对各种逻辑的处理，整个卫星要实现数据共享，都要通过 FPGA 内部存储空间实现，通过在 FPGA 内部采用 FIFO 模块，可以方便地实现各个处理器之间的数据共享。

（2）整个处理模块之间采用中断机制进行相互协调，可方便地实现仿真的数据通

信，避免了在采用计算机仿真时出现的非常难解决同步机制问题。

（3）整个处理模块，各处理器之间采用三总线，即数据/地址/控制总线连接方式实现数据传输，避免了采用计算机仿真时通过 PCI 总线卡实现对下端设备的控制，采用计算机会出现 PCI 总线速率及带宽的限制，同时通过 PCI 总线传输其总线传输时延非常大，是提高系统响应时间的一个瓶颈，并且 PCI 总线不具备实时传输机制，本发明处理模块采用三总线直连方式提高数据传输效率及通信时间，同时还极大地提高了系统响应时间及实时性，提高了卫星的控制精度。

（4）整个处理模块，主要逻辑控制由 FPGA 来实现，因此解决了采用计算机繁杂的握手信号的处理，同时还会消耗大量计算机资源的问题，采用 FPGA 完成逻辑处理方便地实现了各处理器之间的协调。

（5）整个处理模块，所有数据都通过 FPGA 实现数据共享，采用 FIFO 技术实现数据融合，提高了数据的互访效率，避免了采用计算机仿真时使用昂贵的反射内存网实现数据共享。

（6）整个处理模块采用通用化、小型化及模块化设计，不仅适用于单颗卫星控制系统仿真，还可通过扩展接口实现多星仿真。避免了采用计算机进行多星仿真时要使用大量的计算机和浪费大量的空间资源，同时整个处理模块的扩充极大地提高了组建卫星星座仿真的效率。

（7）整个处理模块不仅可以作为一个仿真平台来实现单颗卫星控制系统的仿真，同时也可以将其作为其中一个部件，如将其中的 ARM 处理器作为空间环境单个模型的处理，同时通过 FPGA 实现数据共享，可以方便地实现更为复杂模型的计算，可以通过计算对其他工作提供数据支持，如重新配置整个处理模块的部分程序可以用来进行空间碎片建模计算。

（8）整个处理模块可以进一步扩展，如增加更多的 ARM 处理器，其内部运行实时操作系统用于替换当前仿真中所使用的计算机，如增加更多的 DSP 处理器可以实现更为复杂的模型计算，提高整个卫星控制精度；增加更多的处理器只要按照与之前的处理器连线方式相同即可，同时再按相应的程序配置 FPGA 的管脚就可实现处理器模块的增加。

（9）整个处理模块实现过程容易，对于之前已有的程序不需要作太大的改变，如采用操作系统，只需适当移植相应的程序即可实现与原来采用计算机仿真同样的功能，同时根据不同卫星配置不同程序，极大地提高了卫星前期仿真的效率及降低了前期仿真的运行成本。

附图说明

图 1 为本发明的动态仿真的原理示意图；

图 2 为本发明的多处理器实时仿真平台组成框图；

图 3 为本发明的仿真平台的实现流程图；

图 4 为图 2 中的 FPGA 内部功能图；

图 5 为图 2 中的 ARM1 的实现流程图；

图 6 为图 5 中姿轨控软件的耦合性分析；

图 7 为图 2 中的 ARM2 的实现流程图；

图 8 为图 2 中的 ARM3 的实现流程图；

图 9 为图 7 中由中心刚体和多个挠性附件构成的卫星坐标系；

图 10 为图 2 中的 DSP1 的实现流程图；

图 11 为图 2 中的 DSP2 的实现流程图；

图 12 为图 11 中磁力矩器控制模型程序流程图；

图 13 为图 2 中的扩展接口的实现框图；

图 14 为图 2 中扩展系统的分布式解决方案；

图 15 为本发明的具体实现框图；

图 16 为本发明的通过网络 UDP 方式向地面计算机模拟器发送的归档数据图，其中图 16a 为卫星姿态控制示意图，16b 为敏感器数据显示图，16c 为三维 STK 显示示意图。

具体实施方式

如图 1 所示，本发明整个卫星动态仿真原理。卫星处在不同工作模式下进行不同的姿态/轨道控制，在进行姿态控制时，卫星控制中心处理器首先从敏感器读取相应敏感器的数据，获得当前卫星所处状态即：当前卫星星历、太阳位置、月球位置、地球位置等一系列参数，获得参数后，中心控制器计算当前卫星所需姿态角，同时发出相应控制命令控制执行机构动作；此时卫星本体在卫星自身的执行机构动作后，加上空间摄动的影响，卫星姿态会发生相应的改变，同时敏感器会获得相关参数提供给控制中心处理器进行下一步姿态控制，直至敏感器获得的参数符合卫星所需的数据时为止。

如图 2 所示，本发明由三个或者更多的 ARM 处理器和两个或更多的 DSP 处理器通过 FPGA 作为中间存储器组成，整个处理模块架构简单，全部由单芯片组成，不涉及复杂的外扩芯片及接口或者复杂的总线技术；其中 ARM1 用于仿真卫星控制中心计算机，运行姿态轨道控制算法，按卫星从星箭分离开始进行模式分解，具体模式可根据不同卫星会有不同的区别，但基本上包含速率阻尼模式、太阳捕获模式、地球搜索模式、正常模式、应急模式等。对于不同的模式采用不同的控制方法，一般典型的算法都是基于 PID 控制

律设计；ARM2 用于仿真卫星所处的空间环境，主要是各种摄动力对卫星本身的影响，比较典型的如低轨道卫星考虑较多的为地球非球形摄动、地球引力摄动、大气阻力影响等，对于中高轨道卫星来说，较多考虑太阳光压摄动、月球引力摄动等；根据不同轨道卫星运行轨道不同考虑空间环境仿真会不一样；ARM3 用于卫星的动力学计算，完成对卫星姿态轨道计算，其主要根据当前获得的卫星测量部件信息和控制部件信息，同时根据卫星本身的一些信息如卫星重量、惯量等计算出卫星当前的姿态和轨道数据给控制中心计算机和敏感器及执行机构；两个 DSP 主要用来计算各种不同的模型，可以根据不同的卫星配置不同敏感器和执行机构来配置不同的部件模型，由于地面仿真要尽可能做到与真实部件一致，因此需要芯片支持浮点运算能力和强大的计算能力，DSP 正好可以胜任这一要求。DSP1 主要是敏感器模型的计算，主要计算敏感器相关信息，目前敏感器基本包括：太阳敏感器、地球敏感器、星敏感器、陀螺和磁强计等。根据不同轨道卫星可选择不同的敏感部件，如低轨道小卫星一般选择太阳敏感器、陀螺、磁强计等，因此仿真时只需添加相应的模型即可。整个模型在进行计算时，需要从外部获得相应的信息，因此 DSP1 首先得从外部读取当前卫星姿态轨道信息，这些数据需要从动力学计算机获取，动力学计算机给出信息后，敏感器根据卫星目前的状态，以及敏感器安装的位置，同时根据敏感器本身的工作原理计算出敏感器当前应该输出的数据，DSP2 主要是执行机构模型的计算，目前执行机构主要包括：磁力矩器、推力器、动量轮和控制力矩陀螺。与 DSP1 一样，不同轨道的卫星会选择不同的执行机构部件。其具体情况与 DSP1 相同。

如图 3 所示，本发明的实现过程如下：首先上电启动后，各芯片首先自己运行初始化程序，初始化各种变量、堆栈空间等，完成初始化后，芯片开始等待程序运行指令，星载计算机运行姿态/轨道控制算法，启动程序运行指令，即各处理器接收到一个信号变化后，程序启动运行；ARM1 首先向 DSP1 发出数据获取指令，即控制中心处理器 AMR1 读取当前卫星敏感器数据，DSP1 向 ARM2、ARM3 发出获取数据指令，即敏感器要知道当前卫星所处的空间状态，从 FPGA 中读取到相关数据后即向 ARM1 传送数据；ARM1 根据所读取的数据进行计算获得相应的姿态轨道参数，如果参数满足要求，卫星即进入正常模式，如果不满足要求，ARM1 向 DSP2 发送控制指令，完成对执行机构的控制，同时 DSP2 完成计算后通过 FPGA 向 ARM2、ARM3 输出相应数据，直到姿态轨道满足相应要求。

如图 4 所示，本发明中的 FPGA 作为整个系统仿真中间存储器，主要采用 FIFO 技术实现数据共享，其中一个 ARM 处理器将计算完成的数据通过地址/数据总线写入 FIFO，同时触发其他几个设备，如果此时其他几个设备需要相关数据则发出相关指令读取这些数据，因为每一个设备都对应一个 FIFO，而每个 FIFO 又跟其他几个处理器相连，由于

采用的是地址/数据直接连接，整个读写过程的时间几乎可以忽略不记，同时也避免了同一时刻向同一 FIFO 写入很多数据。FPGA 同时还完成对整个系统仿真的同步控制，避免仿真出现混乱，因为卫星动态仿真过程对同步要求极高，本发明采用 FPGA 完成协调控制，及匹配各处理器的时序逻辑。

如图 5 所示，本发明的 ARM1 由单一芯片构成，其是整个卫星仿真平台的控制核心，其首先通过 FPGA 读取 DSP1 存放在 FIFO 中的数据，进行姿态轨道控制算法计算后，向 FPGA 输出相应的控制命令，同时通知 DSP2 读取相应数据，DSP2 读取相应命令后完成对执行机构的控制模拟。由上可知 ARM1 对外直接与 FPGA 连接，所有数据采用 FPGA 内部 FIFO 模块共享。

ARM1 内部涉及姿态轨道控制算法，其具体实现方法如下：

（1）首先根据系统论原理，对姿态控制算法进行耦合性分析，主要从系统功能分解、硬件组成结构分解、算法分解、参数分解几个层面上进行。一般情况下，由于需实现的功能不同，完成任务所需要的敏感器和执行机构也不同；在硬件组成结构相同的情况下，不同的工况，姿态估计和姿态控制算法也可能不同；在算法相同的情况下，不同的工况，控制参数也不同。所以越接近底层，可分解出来的不同模块越多，呈金字塔形，如图 6 所示。以下说明每个分解层次的具体含义。

功能层分解是根据特定时间或特定工况姿轨控系统完成的任务不同进行分解，分解后得出的功能模式是描述姿轨控系统能力的规范用语。

硬件组成结构层分解是根据姿轨控系统采用的敏感器与执行机构不同进行分解，分解后得出的组成结构模式是描述姿轨控系统工作状况的规范用语。星上电子系统的一些设计工况与此相关。

算法层分解是根据姿轨控系统姿态确定与姿态控制算法的不同进行分解。算法分解不仅是区分不同处，也可以归纳出相同处。分解出的规范算法可以在不同的卫星间共享，增强星载软件系统的可靠性，降低研制风险。

参数层分解是根据设定的修正、反馈和其它工作参数不同进一步对算法进行分解。分解出的核心参数是遥测遥控信号设计的依据之一。

（2）按以上分解完成后，根据不同的模式要求，确定不同的姿态确定算法和控制算法。大体上姿态确定算法有星光与太阳光双矢量定姿、太阳与地磁双矢量定姿、日地双矢量定姿、姿态与陀螺漂移修正等等，控制算法大体上八分区相平面喷气脉宽计算、继电器型角速度喷气控制、推力器分配算法等等，可根据具体实现方式采用不同算法，同时也确定好各算法的输入输出参数。

（3）轨道计算只有两种方法，一种是利用地面站注入的数据进行推算，另一种是采用 GPS 数据，并对注入数据进行修正。

（4）轨道控制主要采用轨控发动机进行轨道修正及变轨。

完成上述步骤后，即可编程统一各种算法，实现不同模式下的调度需求，也就是整个姿态轨道控制算法。

如图 7 所示，ARM2 由单一芯片构成，其主要完成空间环境计算，在接收到 ARM1 发出的运行指令后，ARM2 开始运行，同时将空间环境计算完成的数据存储在 FPGA 中相应的 FIFO 空间处；DSP1 在接到 ARM1 获取数据指令后，立即向 ARM2 发出数据获取指令，同时读取 ARM2 在 FPGA 内部存储好的数据；DSP1 同时也在 FPGA 中存储相关数据供 ARM3 读取。

以天体位置计算为例说明 ARM2 空间环境计算方法：

太阳位置确定实现原理：已知数据注入时确定的历元时刻 τ 及一组太阳轨道根数，可以根据这组根数推算出任意时刻太阳的直角坐标。根据地球自转速率求格林维治赤经，具体实现步骤如下：

（1）确定输入参数即计算历元时刻 t

（2）确定输出参数即太阳在惯性坐标系中的空间位置矢量 $\mathbf{r}$，格林维治赤经 λ_G

（3）注入存贮数据(太阳轨道根数)：e，i，t_0，ω_0，M_0，ω_m，n，λ_{G0}

常数：地球自转角速度 $\omega_E = 7.292115\times10^{-5}\,\mathrm{rad/s}$

这几个数据有具体计算方法如下：

根据天文年历，可以查得相对于当天平春分点的太阳平轨道根数，不考虑当天平春分点与 $J2000.0$ 平春分点的偏差，则太阳在地心惯性坐标系内的平轨道根数描述为

$$e = 0.01670862 - 0.00004204T - 0.00000124T^2$$

$$i = 23^\circ26'21''.448 - 46''.815T - 0''.00059T^2 + 0''.00181T^3$$

$$\omega_0 = 282^\circ56'14''.45 + 6190''.32T + 1''.655t^2 + 0''.012T^3$$

$$M_0 = 357^\circ31'44.76'' + 129596581''T - 0''.562T^2 - 0''.012T^3$$

$$n = 1.990968752798\times10^{-7}$$

$$\omega_m = -9.510076261979\times10^{-12}$$

t_0 时的格林威治角

$$\lambda_{G0} = 6^\circ41'50.54841'' + \frac{8640184.812866^\circ}{3600}T + \frac{0.093104^\circ}{3600}T^2 - \frac{6.2^\circ}{3600}T^3$$

其中T为$J2000.0$起算的时间间隔儒略世纪数，其计算公式为

$$T=\frac{JD(t)-JD(J2000.0)}{36525.0}$$

其中$JD(t)$是计算时刻t对应的儒略日，$JD(J2000.0)$是历元$J2000.0$对应的儒略日。

（4）星上具体计算太阳位置在惯性坐标系中的位置方法如下：

t时刻轨道根数

$$M=M_0+n(t-t_0)$$

$$\omega=\omega_0+\omega_m(t-t_0)$$

$$f=M+2e\sin M+1.25e^2\sin(2M)+\omega$$

太阳在惯性坐标系中的位置为

$$\mathbf{r}=\begin{bmatrix}\cos f\\ \sin f\cos i\\ \sin f\sin i\end{bmatrix}$$

格林维治赤经

$$\lambda_G=\lambda_{G0}+\omega_E(t-t_0)$$

根据以上4步，就可以得出当前时刻t，太阳在惯性坐标系下的具体位置。

如图8所示，本发明的动力学计算处理器AMR3主要完成卫星当前姿态轨道计算，动力学处理器等待DSP1通知其读取数据，即ARM3得到DSP1发出的读取指令后，从FPGA中所对应的FIFO处读取当前卫星敏感器的状态数据，同时进行卫星当前姿态及轨道参数计算，并将数据存储至FPGA中；DSP2得到ARM1发出控制指令后，也进行模型计算，将结果存储至FPGA相应的FIFO中，同时通知ARM3读取当前数据，ARM3接收到最新数据后计算，同时将结果存储至FPGA的FIFO中。

卫星当前姿态及轨道参数计算中涉及的控制参数包括轨道、姿态及质量、惯量等参数，各种敏感器参数，执行机构参数，气动外形参数，太阳帆板参数，传输参数以及动力学模型。

轨道和姿态参数包括目标姿态参数；敏感器参数包括星敏感器、太阳敏感器、地球敏感器、陀螺和磁强计的安装角和视场角等参数；执行机构参数包括动量轮的惯量、最大角速度、噪声初始转速，推力器的平均推力、力矩矢量、安装点、噪声，磁力矩器的最大力矩、时延噪声等；外形参数包括气动系数，本体面积，几何中心等；太阳帆板参数包括几何中心，模态参数等。

卫星当前姿态及轨道参数计算方法如下：

（1）完成对卫星本体建模，得出卫星运动的线动量方程。建立带挠性附件的三轴稳定卫星动力学模型。假设卫星由一个中心刚体 B 和 m 个一端与其相连的挠性附件 $\mathbf{B}_i(i=1,2,\cdots,m)$ 构成，如图 9(由中心刚体和多个挠性附件构成的卫星坐标系)所示，并建立惯性坐标系 S_i、本体坐标系 S_b 与挠性附件坐标系。

假设用有限元法求得第 i 个挠性附件 $\mathbf{B}_i$ 相对挠性附件坐标系的悬臂振型和频率为：

$$\underline{\underline{\boldsymbol{\varphi}_{ik}}}=[\boldsymbol{\varphi}_{i1}(r_{ik}),\boldsymbol{\varphi}_{i2}(r_{ik}),\cdots\boldsymbol{\varphi}_{in}(r_{ik})],(k=1,2,\cdots n_i) \quad \omega_{i1},\omega_{i2},\cdots\omega_{in}$$

其中 $\mathbf{r}_{ik}$ 为第 i 个挠性附件的第 k 个节点相对挠性附件坐标系的原点 O_i 的矢径，n_i 为第 i 个挠性附件节点总数。设 N 为振型截断数，且假定每一附件的 N 均相等，$\underline{\underline{\boldsymbol{\varphi}_{ik}}}$ 为 $3\times N$ 矩阵。其中带有双划线的变量均表示矢矩阵。

而在第 i 个挠性附件坐标系中第 k 个节点 $\mathbf{r}_{ik}$ 的位移 $\mathbf{u}_{ik}$ 可表示为：

$$\mathbf{u}_{ik}=\underline{\underline{\boldsymbol{\varphi}_{ik}}}\mathbf{q}_i(t)$$

其中，$\mathbf{q}_i(t)=[q_{i1}(t),q_{i2}(t),\cdots q_{iN}(t)]^T$ 为第 i 个挠性附件的模态坐标。

则卫星运动的线动量方程可写为：

$$m_T\frac{d\mathbf{V}_T}{dt}+\sum_i\underline{\underline{\mathbf{B}_{trani}}}\ddot{\mathbf{q}}_i(t)=\mathbf{F}$$

其中

$$\underline{\underline{\mathbf{B}_{trani}}}=\underline{\underline{\mathbf{T}_{SiB}}}\,\underline{\underline{\mathbf{B}^i_{tran}}}$$

$$\underline{\underline{\mathbf{B}^i_{tran}}}=\sum_k m_{ik}\underline{\underline{\boldsymbol{\varphi}_{ik}}}$$

式中 $\underline{\underline{\mathbf{B}_{trani}}}$ 为 $3\times N$ 矩阵，是第 i 个挠性附件相对于本体坐标系 S_b 的平动耦合系数；$\underline{\underline{\mathbf{B}^i_{tran}}}$ 是第 i 个挠性附件在挠性坐标系中相对于 O_i 的平动耦合系数 m_T。

（2）确立卫星的角动量方程：设 $\mathbf{H}$ 和 $\mathbf{T}_{SAT}$ 为系统相对于 O 点的绝对角动量和力矩，偏置动量轮的角动量为 $\mathbf{h}$，$\boldsymbol{\omega}$ 并假设变形后质心运动速度很小，根据角动量定律和有限元分析结果可得卫星的角动量方程为：

$$\underline{\underline{\mathbf{I}_T}}\dot{\boldsymbol{\omega}}+\dot{\mathbf{h}}+\boldsymbol{\omega}\times\mathbf{h}+\sum_{i=1}^{m}\underline{\underline{\mathbf{B}_{roti}}}\ddot{\mathbf{q}}_i(t)=\mathbf{T}_{SAT}$$

其中：

$$\underline{\underline{\mathbf{B}_{roti}}} = \underline{\underline{\tilde{\mathbf{I}}_{pi}}}\,\underline{\underline{\mathbf{B}_{trani}}} + \underline{\underline{\mathbf{T}_{SiB}}}\,\underline{\underline{\mathbf{B}^{i}_{rot}}}$$

$$\underline{\underline{\mathbf{B}^{i}_{rot}}} = \sum_{k=1}^{ni} m_{ik}\,\underline{\underline{\tilde{\mathbf{r}}_{ik}}}\,\underline{\underline{\boldsymbol{\varphi}_{ik}}}$$

式中 $\underline{\underline{\mathbf{B}_{roti}}}$ 是第 i 个挠性附件相对于本体坐标系 S_b 的转动耦合系数；$\underline{\underline{\mathbf{B}^{i}_{rot}}}$ 是第 i 个挠性附件在挠性坐标系中相对于 O_i 的转动耦合系数。

（3）确定卫星运动方程，对于挠性附件的运动方程，应用变分原理进行推导。考虑挠性附件的结构阻尼 ζ，最后可得其运动方程为：

$$\ddot{\mathbf{q}}_i + 2\underline{\underline{\zeta_i}}\,\underline{\underline{\boldsymbol{\Lambda}_i}}\,\dot{\mathbf{q}}_i + \underline{\underline{\boldsymbol{\Lambda}_i^2}}\,\mathbf{q}_i + \underline{\underline{\mathbf{B}^{T}_{trani}}}\,\frac{d\mathbf{V}_T}{dt} + \underline{\underline{\mathbf{B}^{T}_{roti}}}\,\dot{\boldsymbol{\omega}} = 0$$

其中

$$\underline{\underline{\boldsymbol{\Lambda}_i^2}} = diag[\omega_{i1}^2, \omega_{i2}^2, \cdots \omega_{iN}^2]$$

（4）将上述三个方程归纳在一起，即得到经过线性化处理的卫星动力学方程。

如图 10 所示，本发明的 DSP1 是仿真敏感器模型，DSP1 首先得到 ARM1 发出读取指令，其从 FPGA 中读取由 ARM3 计算出的结果，DSP1 获得数据后进行相关计算，然后将计算结果存储到 FPGA 供 ARM1 读取。

如图 11 所示，本发明的 DSP2 是仿真执行机构模型，DSP2 首先得到 ARM1 发出读取指令，其从 FPGA 中读取由 ARM3 计算出的结果，DSP2 获得数据后进行相关计算，然后将计算结果存储到 FPGA 供 ARM1、ARM3 读取。

以下以磁力矩器模型为例说明 DSP1 和 DSP2 内部模型软件构成。

（1）假设条件：设定磁矩噪声 $\mathbf{M}_{noise}$，其中常值噪声为 $\mathbf{M}_{n.constant}$，随机噪声为 $\mathbf{M}_{n.random}$。

（2）定义磁力矩器不同时间段的含义：磁力矩从开始响应到规定结束响应时刻的时段称为 $T_{duration}$，其中磁力矩器从产生磁矩到达到期望磁矩值的时段成为 $T_{prepare}$；结束响应时刻到磁力矩器的磁矩撤销为零的时段称为 T_{lag}，其中磁力矩器迅速衰减的时段称为 T_{extent}，T_{extent} 末时刻磁矩占 $\mathbf{M}_T$ 的百分比为 p。计算磁矩值的某一时刻为 $t_{command}$，当前时刻为 t_{now}。

（3）确定磁力矩器控制模型中需要设置的参数，即某一时刻的磁矩 $\mathbf{M}$；期望产生的磁矩 $\mathbf{M}_T$；磁力矩器能产生的最大磁矩 $\mathbf{M}_{\max}$，最终剩磁 $\mathbf{M}_{remain}$。磁力矩器的位置矢量 $\mathbf{r} = [x \quad y \quad z]^T$。

（4）具体计算过程：

$\Delta T = t_{command} - t_{now}$

若 $\Delta T \geq T_{duration}$， $\Delta T_1 = \Delta T - T_{duration}$，则

当 $\Delta T_1 \leq T_{extent}$ 且 $T_{extent} > 0.0$ 时， $\mathbf{M} = (1 - \Delta T_1 / T_{extent}) \cdot \mathbf{M}_T \cdot (1 - p) + \mathbf{M}_{noise}$

当 $T_{extent} < \Delta T_1 < T_{lag}$ 时， $\mathbf{M} = ((\Delta T_1 - t_{extent}) / (T_{lag} - T_{extent})) \cdot \mathbf{M}_T \cdot p + \mathbf{M}_{noise}$

当 $\Delta T_1 \geq T_{lag}$ 时， $\mathbf{M} = \mathbf{M}_{remain}$；

若 $\Delta T < T_{prepare}$ 且 $T_{prepare} > 0.0$， $\mathbf{M} = (1 - \Delta T / T_{prepare}) \cdot \mathbf{M}_T + \mathbf{M}_{noise}$；

若 $T_{prepare} \leq \Delta T < T_{duration}$， $\mathbf{M} = \mathbf{M}_T + \mathbf{M}_{noise}$。

将上述的磁力矩器控制模型算法用程序流程图的形式表示如图 12 所示。

如图 13 所示，本发明的扩展接口主要是从 ARM1 扩出，因为 ARM1 主要是作控制计算机，在整个卫星大系统仿真时，主要由控制中心计算机与其他系统相连，因此整个处理模块在外扩接口时也主要选择作控制中心处理器的 ARM 芯片，同时 ARM 芯片自带相应的接口控制器，只需加载相应的接口处理芯片即可。图 14 所表示的是整个平台的整体扩展解决方案，其中在进行单星仿真时，可采用上述所示采用一个处理平台对不同部件功能分配从而实现仿真，如果要进行多星仿真，要求采用的部件众多，而且各星参数不一样，因此需要采用图 14 所示采用多个平台组合成所需要的敏感器平及执行机构平台，星载计算机平台。其中动力学计算和空间环境采用采用一个平台即可，通过 CAN 总线组成分布式实时仿真系统。这样大大缩短了卫星仿真平台组建的时间，提高了仿真效率。同时整个平台硬件完全统一，只是软件参数不一样，这样的模块化平台可以节约成本，降低风险。整个分布系统进行多星仿真可为今后卫星编队飞行及星座分布提供数据支持。

如图 15 所示，图 15 是整个模块具体实现图，ARM1 发出仿真开始指令，各处理器完成初化后报告 ARM1 初始化完成；ARM1 向 DSP1 发出读取当前敏感器数据指令，DSP1 采用外中断响应，DSP1 进入中断服务程序，同时通知 ARM3，ARM3 响应中断，并向 ARM2 发出更新数据指令，ARM2 将当前空间信息存储在 FPGA 中，ARM3 读取当前空间信息按相应控制算法计算当前敏感器数据，并将计算结果存储在 FPGA 中，DSP2 发现 FPGA 中数据更新后读取最新数据，并通过模型计算出相应参数，并将参数存储至 FPGA，ARM1 立即将最新数据读取，通过相应控制算法完成对当前卫星姿态解算，如果符合要求，则进入正常模式，如果卫星当前姿态不符合目前需求，则计算出相应执行机构该如何控制，将计

算结果存储至 FPGA，同时向 DSP2 发出控制指令，DSP2 中断响应后，立即从 FPGA 读取相应控制命令，DSP2 对命令进行解析后，按模型进行计算，将计算结果存储至 FPGA，同时通知 ARM1 和 ARM3 读取最新数据，ARM1 和 ARM3 获得指令后，读取 DSP2 存储的最新数据，并进行计算，ARM3 根据姿态/轨道算法，进行当前卫星姿态/轨道解算，同时将结果存储至 FPGA，ARM1 获得相关数据后进行相应简单计算，将计算结果存储至 FPGA，同时 ARM1 再次读取 DSP1 存储在 FPGA 中的数据，整个过程周而复始，直到卫星当前姿态/轨道符合要求为止。由于每一个处理器在 FPGA 中都有对应的 FIFO,只要其向相应的 FIFO 写入当前最新信息即可，FPGA 根据相应的控制逻辑来判定由谁读取当前哪一个 FIFO 的数据，同时在 FIFO 数据更新时通知相应的处理器做好数据接收的准备。

如图 16 所示，本发明通过网络 UDP 方式向地面计算机模拟器发送的归档数据。从图 16 可以监测当前卫星各个敏感器及执行机构的各种状态，从图 16a 上可以监测其各种姿态数据。图 16b 是用来监视整个系统敏感器的各种参数，同时接入三维显示可从 STK 显示卫星当前轨道和姿态信息，如图 16c。

图 16 整个属于地面监测系统，本发明在完成数据处理时，将相应的归档数据经以太网 UDP 方式发送至地面监测系统，图上显示是具体实现过程中地面监测系统所显示的界面。原本卫星系统仿真是采用计算机完成，由控制系统和地面监测系统组成，本发明主要是为了替代当前由计算机组成的卫星控制仿真系统，由于其与地面监测系统采用以太网连接，因此本发明在对个扩展接口时也有以太网接口，同时为了更好地实现程序移植，在计算机仿真时，动力学计算机、环境计算机、星载计算机模拟器均采用 VxWorks 操作系统，因此在移植到 ARM 处理器还同样采用实时操作系统实现，对于模型库也采用之前的编写的模型程序，因此大大加快了整个开发进度，提高了仿真效率，达到了应有的效果。将发明接入相应的地面监测系统，就可以非常直观地观测整个仿真系统的运行过程。

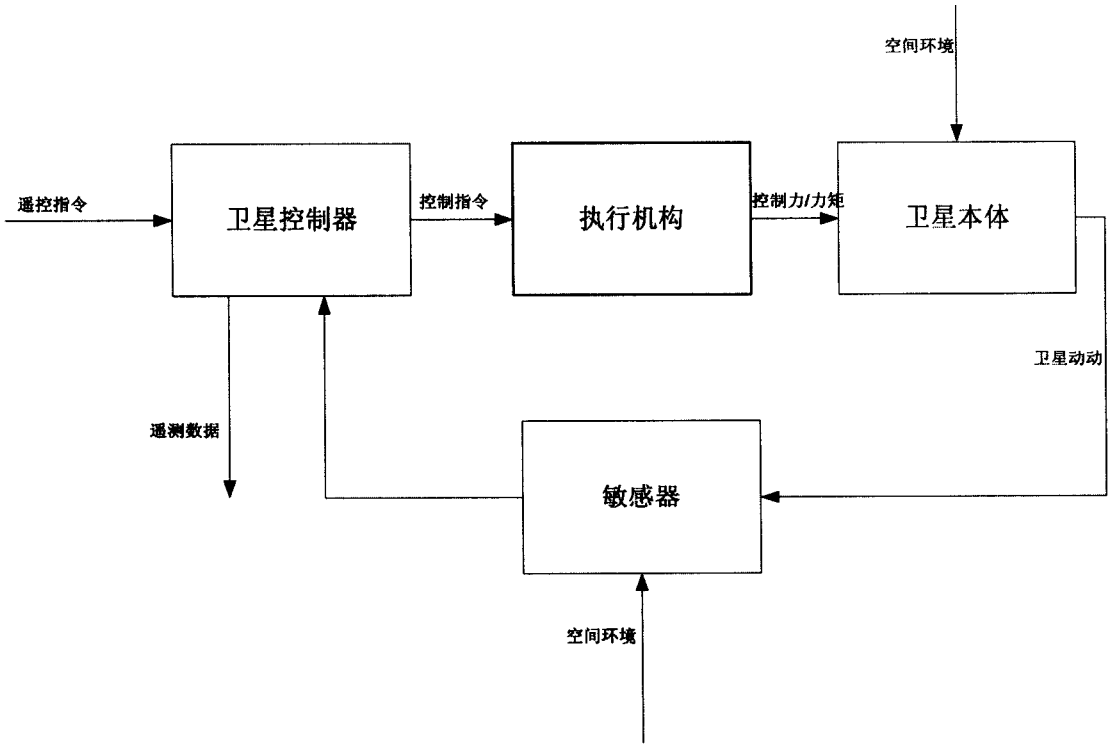

空间环境
遥控指令
卫星控制器
控制指令
执行机构
控制力/力矩
卫星本体
卫星动动
遥测数据
敏感器
空间环境

图 1

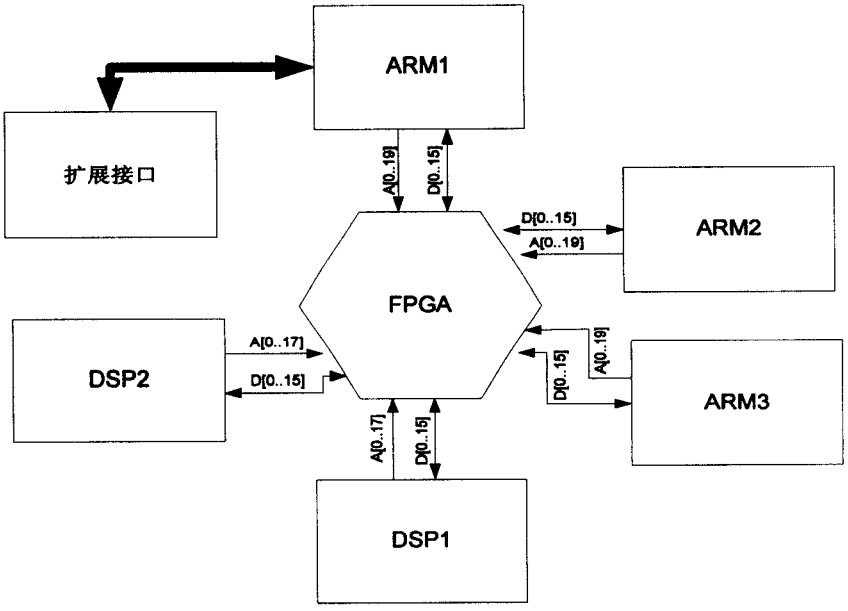

ARM1
扩展接口
A[0..19]
D[0..15]
D[0..15]
A[0..19]
ARM2
FPGA
DSP2
A[0..17]
D[0..15]
A[0..19]
D[0..15]
ARM3
A[0..17]
D[0..15]
DSP1

图 2

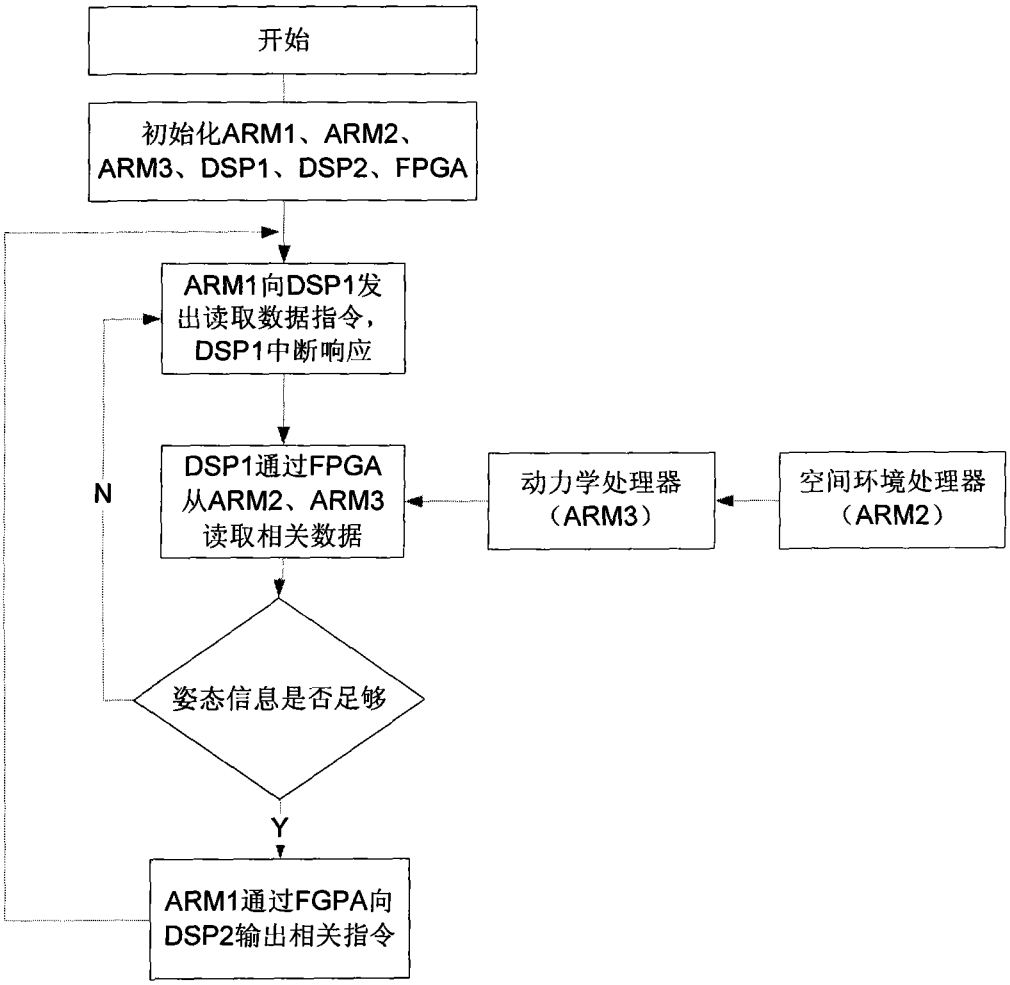

开始
初始化ARM1、ARM2、ARM3、DSP1、DSP2、FPGA
ARM1向DSP1发出读取数据指令，DSP1中断响应
DSP1通过FPGA从ARM2、ARM3读取相关数据
动力学处理器（ARM3）
空间环境处理器（ARM2）
N
姿态信息是否足够
Y
ARM1通过FGPA向DSP2输出相关指令

图 3

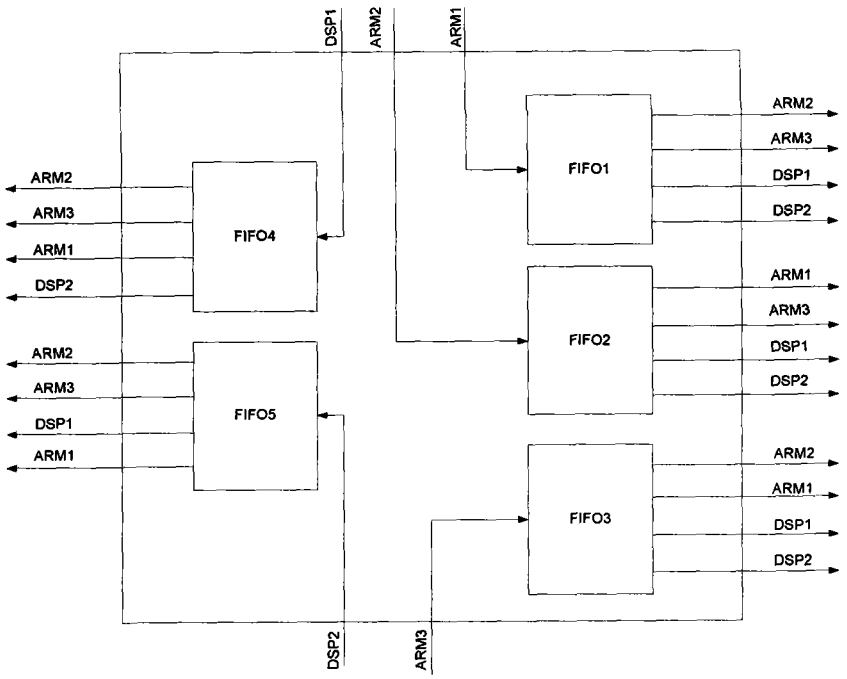

DSP1
ARM2
ARM1
FIFO1
FIFO2
FIFO3
FIFO4
FIFO5
ARM2
ARM3
DSP1
DSP2
ARM1
DSP2
ARM3

图 4

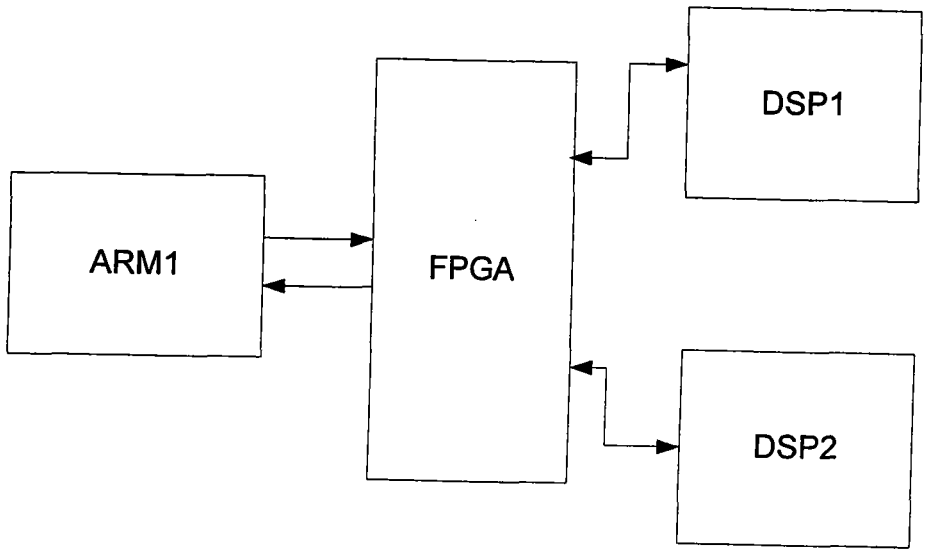


图 5

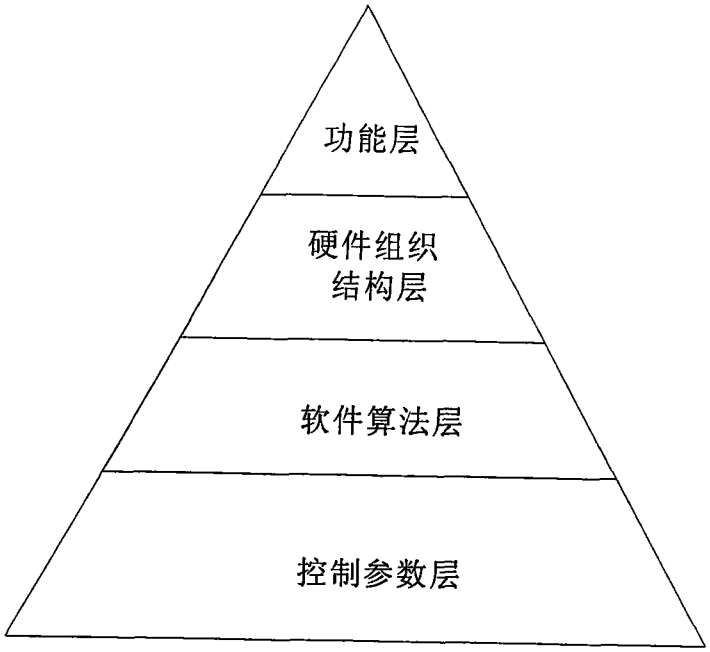


图 6

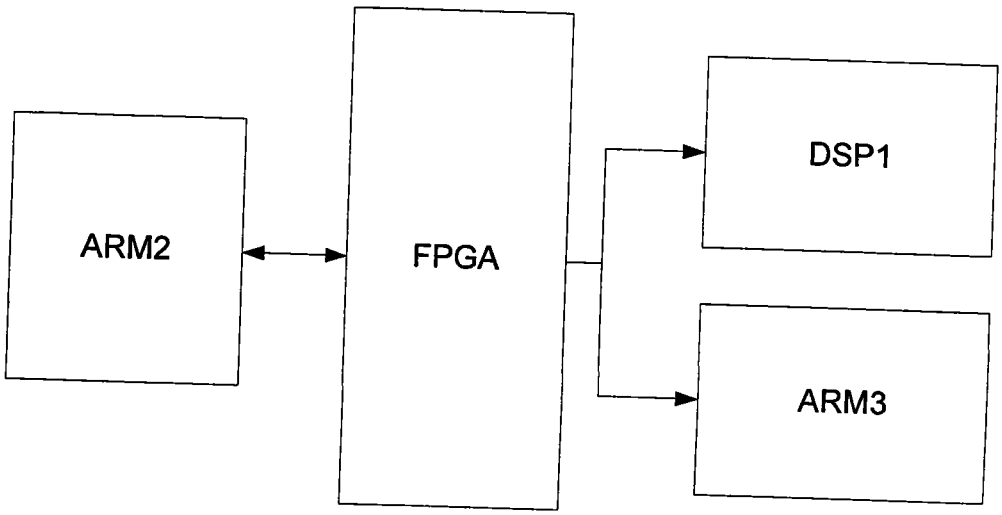


图 7

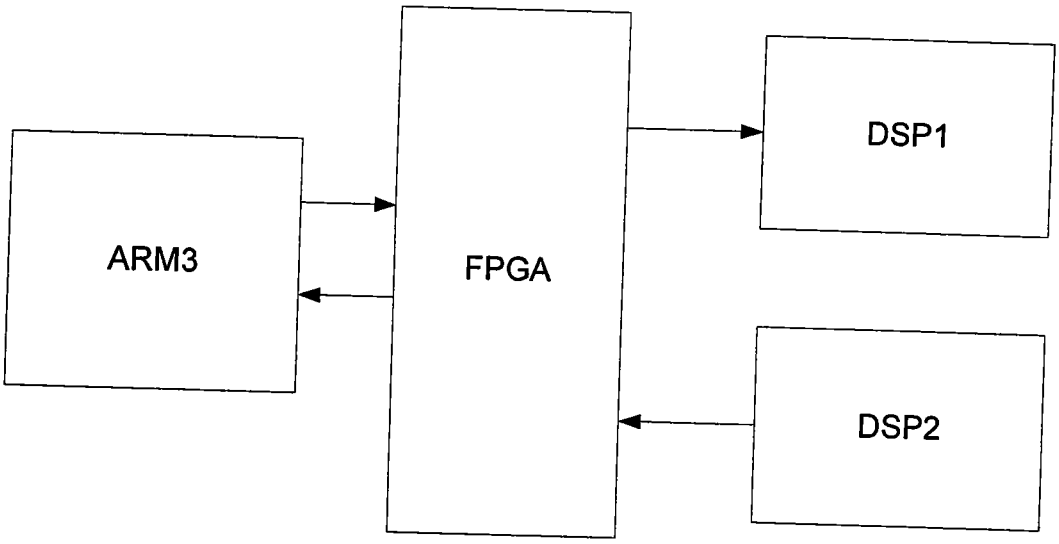


图 8

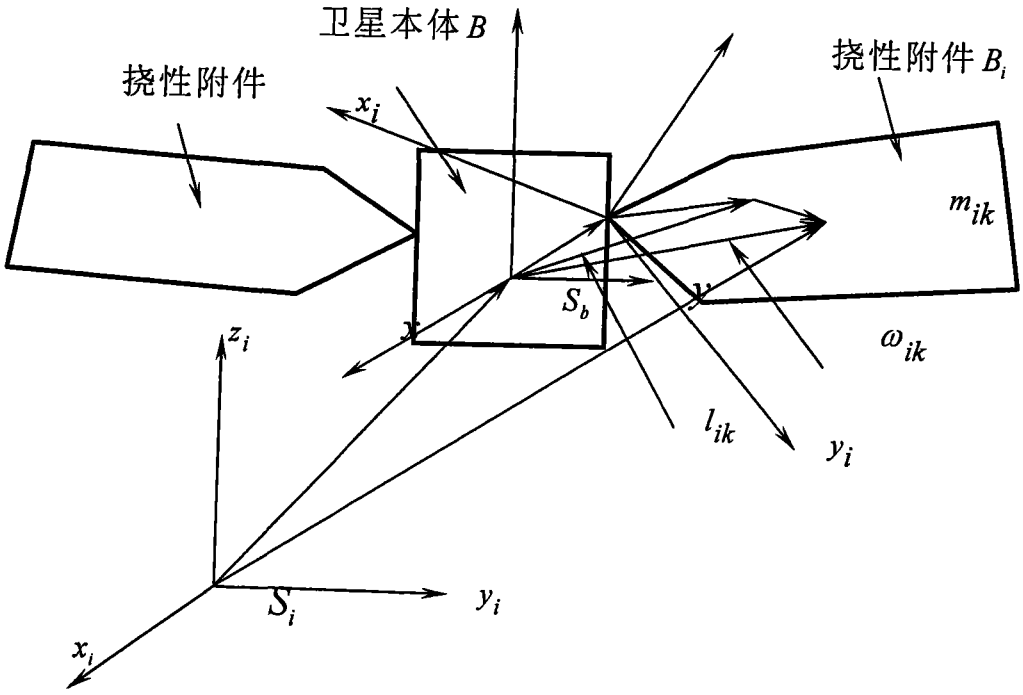

卫星本体B
挠性附件
挠性附件B_i
x_i
m_{ik}
S_b
z_i
ω_{ik}
l_{ik}
y_i
S_i
y_i
x_i

图 9

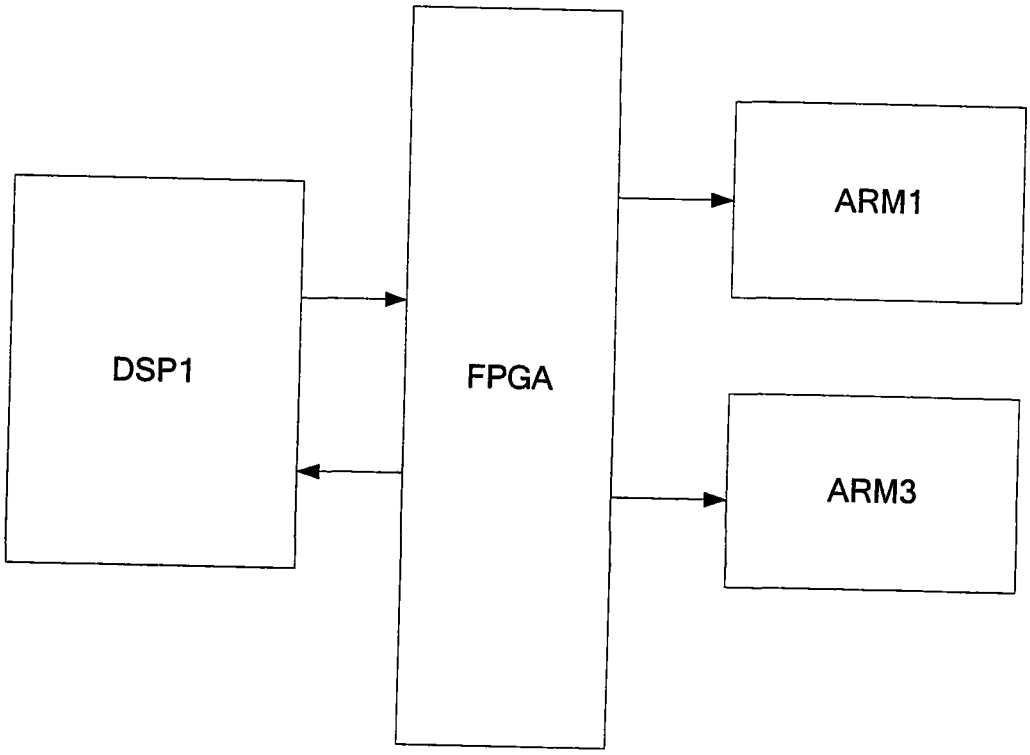

DSP1
FPGA
ARM1
ARM3

图 10

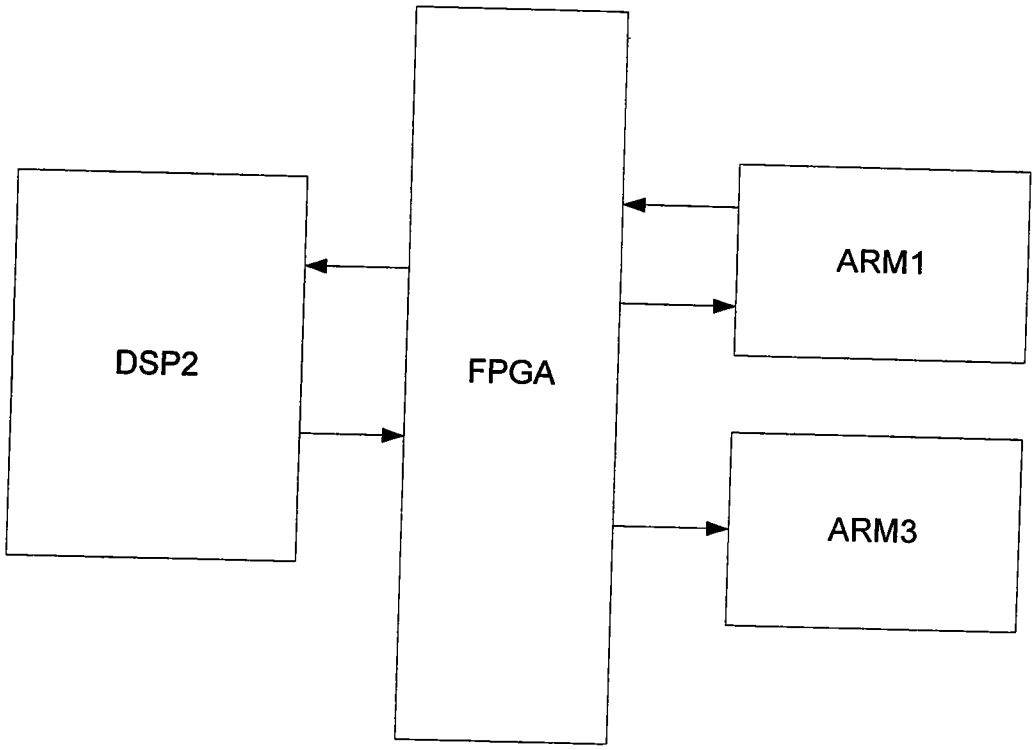


图 11

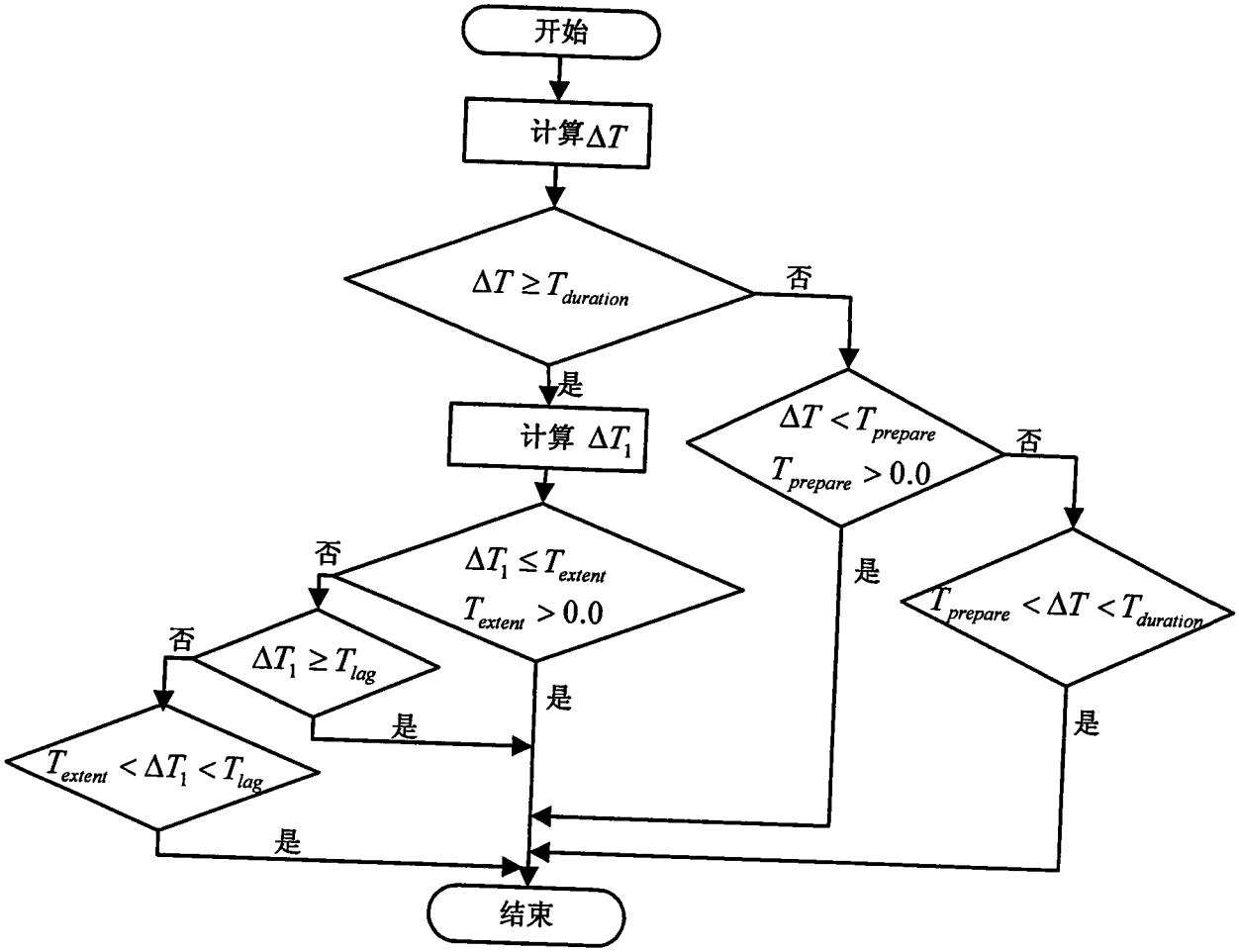

开始
计算ΔT
ΔT ≥ T_duration
否
是
计算 ΔT_1
ΔT < T_prepare
T_prepare > 0.0
否
是
ΔT_1 ≤ T_extent
T_extent > 0.0
否
T_prepare < ΔT < T_duration
否
是
ΔT_1 ≥ T_lag
是
是
T_extent < ΔT_1 < T_lag
是
结束

图 12

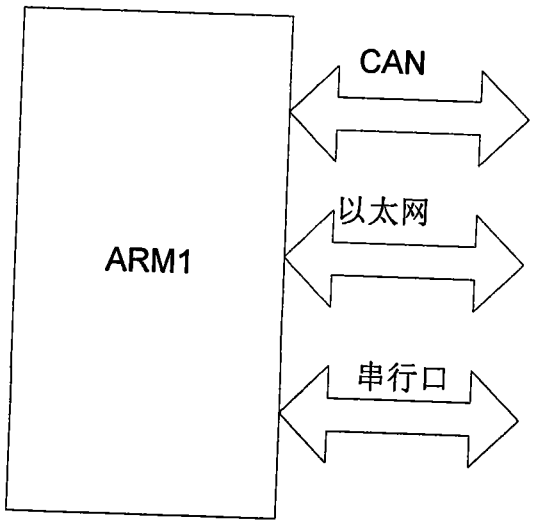

CAN
以太网
ARM1
串行口

图 13

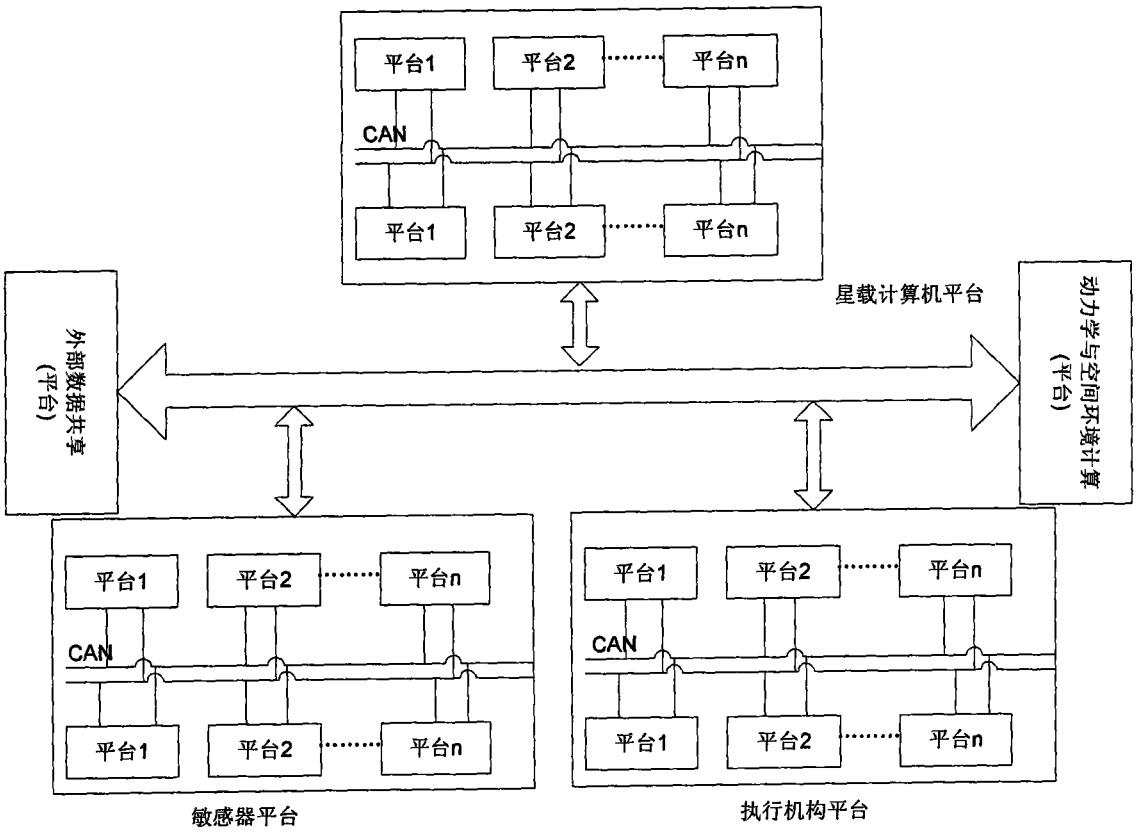

平台1
平台2
平台n
CAN
平台1
平台2
平台n
星载计算机平台
外部数据共享(平台)
动力学与空间环境计算(平台)
平台1
平台2
平台n
CAN
平台1
平台2
平台n
敏感器平台
平台1
平台2
平台n
CAN
平台1
平台2
平台n
执行机构平台

图 14

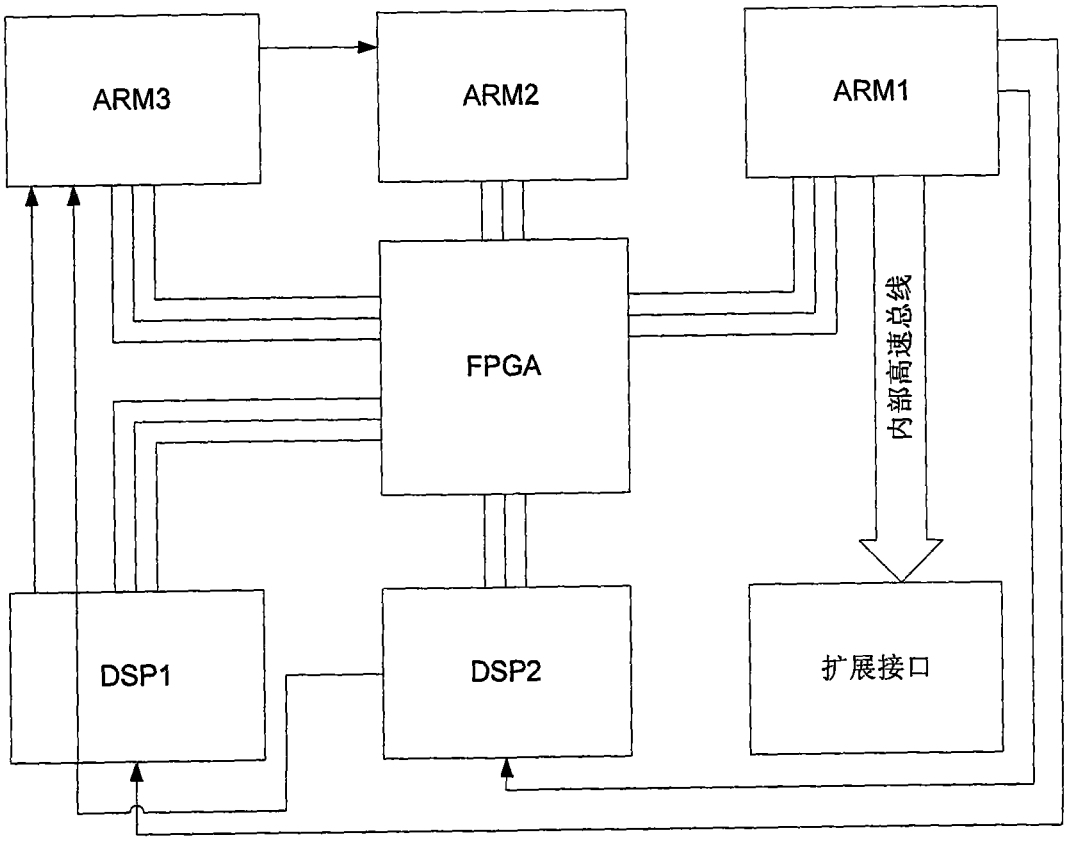

ARM3
ARM2
ARM1
FPGA
内部高速总线
DSP1
DSP2
扩展接口

图 15

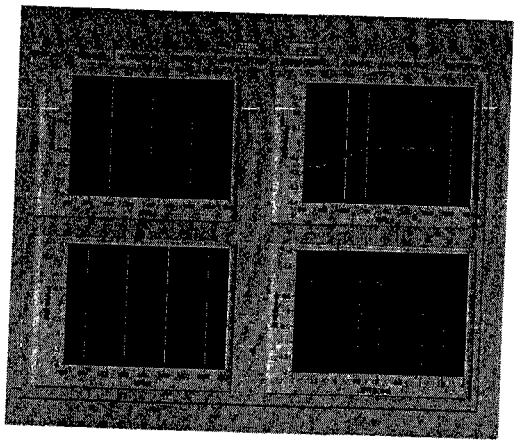


图 16 a

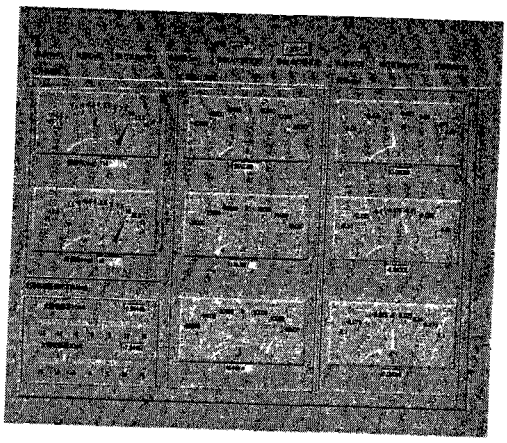


图 16b

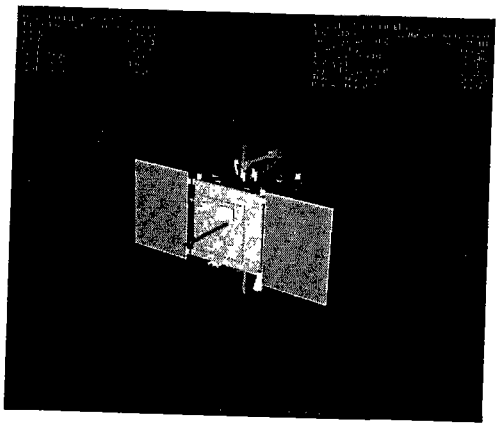


图 16c