第38卷 第18期
Vol.38 No.18

计 算 机 工 程
Computer Engineering

2012年9月
September 2012

·工程应用技术与实现· 文章编号：1000—3428(2012)18—0221—04 文献标识码：A 中图分类号：TP391

基于双口 RAM 的冗余架构同步技术研究

张遵伟[1]，曹宝香[1]，聂胜伟[2]

(1. 曲阜师范大学计算机科学学院，山东 日照 276826；2. 中国电子科技集团公司第三十二研究所，上海 200233)

摘 要：冗余设计可提高容错设计中的可靠性，但其存在同步和丢拍的问题。为此，以航天某型号高可靠箭载嵌入式计算机研制为背景，结合飞行控制任务的需求和特点，基于 DSP 及双口 RAM，利用中断和查询相结合的方式，提出一种“1+3”冗余架构中的信息交互方案。应用结果表明，该方案能有效地解决冗余架构中同步及丢拍问题，并且在满足实时性的同时，提高可靠性。

关键词：双口 RAM；冗余设计；容错设计；同步；可靠性

Study on Synchronization Techniques for Redundancy Architecture Based on Dual-port RAM

ZHANG Zun-wei[1], CAO Bao-xiang[1], NIE Sheng-wei[2]

(1. College of Computer Science, Qufu Normal University, Rizhao 276826, China;
2. The 32nd Institute of China Electronics Technology Group Corporation, Shanghai 200233, China)

【Abstract】Redundancy design is a kind of important means to improve the reliability of fault-tolerant design, however, the redundancy design exists two outstanding problems which are the synchronization problem and the packet-loss problem. This paper mainly takes a certain type of spaceflight high reliable rocket embedded computer development as the background and analyzes demands and characteristics of the rocket flight control task. By using interrupt and query combination mode, it puts forward a kind of information interaction scheme for “1+3” redundant structure based on DSP and dual-port RAM. Application results show that this kind of information interaction scheme solves the synchronization problem and the packet-loss problem and meets the real-time requirements, while achieves high reliability.

【Key words】dual-port RAM; redundancy design; fault-tolerant design; synchronization; reliability

DOI: 10.3969/j.issn.1000-3428.2012.18.059

1 概述

在航天系统中，高可靠性是航天器的一项非常重要的指标[1]。目前，提高可靠性主要有 2 种途径：一种是避错设计，即设计一个不包含故障的“完美”系统，要绝对做到这一点，实际上是不可能的。另一种途径是容错设计[2]，其基本思想是通过对系统结构进行容错设计，利用外加资源的冗余技术屏蔽故障的影响。由于容错设计更有效且实际可行，容错设计已成为提高可靠性的一种重要手段，然而冗余设计中也存在着几个突出问题，例如交互同步问题及丢拍问题等[3-4]。

本文以航天某型号高可靠箭载嵌入式计算机研制为背景，基于双口 RAM 及 DSP 实现的冗余系统，采用中断与查询相结合的方案有效解决冗余结构中交互同步及丢拍的问题，从而实现系统的高可靠性。

2 信息交互方案

2.1 硬件架构

为满足航天系统中高可靠性的要求，箭载计算机采用了“1+3”模式的冗余架构，如图 1 所示，一块处理器(MCU)作为控制中心执行控制任务，其余 3 块处理器(DPU)作为运算中心，执行主要的运算操作，并且这 3 块处理器执行的运算操作完全相同以达到冗余的目的。

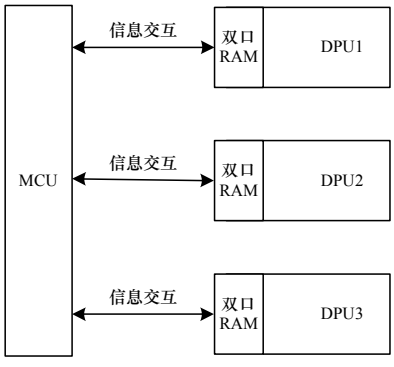

图 1 系统架构

在图 1 中，每个 DPU 对应一个独立的双口 RAM，MCU 与 DPU 通过双口 RAM 进行信息交互。MCU 作为控制中心除了执行相关的控制任务外，还要对 3 个 DPU 的数据信息进行 3 取 2 表决，并将最终表决的结果作为架构的最终输出。单 MCU 模块作为架构的关键节点，设计思

作者简介：张遵伟(1987－)，男，硕士研究生，主研方向：系统集成，企业信息化；曹宝香，教授；聂胜伟，高级工程师

收稿日期：2011-12-15 **修回日期：**2012-02-04 **E-mail：**zhangzunwei101@163.com

路坚持简单可靠的原则即在满足可靠性和性能要求的前提下，采用的技术和方法应尽可能简单，元器件的种类和数量应尽可能少[5]。

2.2 架构的可靠性

度量系统的可靠性有 2 个主要的参数，即可靠度 $R(t)$ 和可用度 $A(t)$[6]。为方便计算，在此采用可靠度进行架构可靠性的分析。所谓可靠度 $R(t)$是指在时间 t=0 时系统正常的条件下，系统在时间区间[0,t]内能正常运行的概率。式(1)给出了可靠度的计算方法：

$$R(t)=\mathrm{e}^{-\lambda t} \tag{1}$$

其中，λ为失效率。

组合模型[7]是计算容错系统可靠性最常用的方法。由图 1 可知，“1+3”模式冗余架构作为一个典型的串并联系统，即 MCU 模块作为 1 个子系统与 3 个 DPU 模块并联构成的子系统进行的串联，完全满足组合模型的条件。

为此，本文采用组合模型的方法对“1+3”模式冗余架构进行分析得出该架构的可靠性数学模型为：

$$R_{\mathrm{sys}}=R_{\mathrm{mcu}}\times R_{\mathrm{3dpu}} \tag{2}$$

其中，R_{3dpu}为 3 个 DPU 模块并联构成的子系统的可靠性，该子系统可靠性的计算需要考虑 3 取 2 表决的存在，因此，其可靠性为至少有 2 个 DPU 模块可靠的概率，如下式所示：

$$R_{\mathrm{3dpu}}=C_3^2\times R_{\mathrm{dpu}}^2\times\left(1-R_{\mathrm{dpu}}\right)+C_3^3\times R_{\mathrm{dpu}}^3=3R_{\mathrm{dpu}}^2-2R_{\mathrm{dpu}}^3 \tag{3}$$

由式(2)、式(3)可知，要求解系统的可靠性，关键在于求解 MCU 模块及 DPU 模块的可靠性。根据系统任务的飞行要求，取飞行时间为 1 h，并且通过 GJB299/C-2006《电子设备可靠性预计手册》及集成电路失效率计算模型可预计 MCU 及 DPU 的失效率分别为 37.219 4×10^{-6}/h、96.286 8×10^{-6}/h，代入式(1)可得 MCU 及 DPU 可靠性分别为 R_{mcu}=0.999 963，R_{dpu}=0.999 904。由式(3)得到 R_{3dpu}=0.999 999，最后由式(2)可得系统架构的可靠性为 R_{sys}=0.999 962，该可靠性值完全满足控制系统任务对可靠性不小于 0.999 7 的要求。由此也可以看出，虽然 MCU 作为架构中可靠性的相对薄弱点，但是通过计算其相应的失效率最终得到的“1+3”模式冗余架构的可靠性完全满足系统的要求。

2.3 信息交互方式

MCU 与 DPU 主要有 2 种交互方式，即中断方式和查询方式。

2.3.1 中断方式

中断方式是根据双口 RAM 自身的中断仲裁方式[4-5]提出的，交互的双方主要通过硬件产生的中断信号通知对方读取数据，该方式实现简单且具有实时性高的优点，然而在交互双方交互的信息量较大并且同时存在周期性及非周期性数据时，容易产生丢失中断的现象，导致交互双方无法正常交互下去。

2.3.2 查询方式

查询方式主要是从软件的方面考虑，即对于交互双方交互的有效数据信息在设计其相应的消息结构时，增加状态标志信息，在写数据的一方将有效信息写入双口 RAM 的同时，设置其相应的状态标志位，而在取数据的一方增加一个毫秒级查询模块，该模块主要用于查询与有效信息相对应的状态标志，进而判断有效信息是否已经存在于双口 RAM 中，当取走有效信息后还要将相应的状态标志还原。该方式的主要优点是信息交互过程中不易丢失数据，从而保证交互双方可靠的通信。然而与中断方式相比，查询方式的实时性差，又由于查询方式实现的关键在于合理设计交互信息的结构，因此实现起来比中断方式复杂。

2.4 技术解决方案

MCU 与 3 个 DPU 交互的数据信息量较大，主要包括各类遥测信息及控制命令信息，同时由于任务本身对实时性的较高要求，MCU 与 DPU 执行的任务时序较为紧密。为保证 MCU 与 DPU 之间可靠的通信，结合中断和查询的优缺点，方案采用中断方式和查询方式相结合的模式实现 MCU 与 DPU 的通信。

根据数据传输的方向，可以将数据分成输入数据和输出数据 2 类。下面具体针对不同的数据设计相应的消息格式从而满足中断或者查询的要求，提高通信的效率。

2.4.1 数据输入

MCU 传给 DPU 的数据称为输入数据，主要包括 A 控制数据、B 控制数据、姿态数据。其中，A、B 控制数据周期为 5 ms，作为 DPU 的基准输入，具有很高的实时性要求，并且要保证 3 个 DPU 获取到的数据一致，即 3 个 DPU 采用同一拍数据，为此对于 A、B 控制数据采用中断的方式进行传输，而对于姿态数据其周期为 20 ms，采用查询的方式即可保证数据的传输。

(1)A、B 控制数据帧结构

对于 A、B 控制数据，为满足中断传输要求。采用如表 1 所示的帧结构，其中“数据标志”字段指明结构中的数据属于哪类数据；“有效数据个数”字段用于查询时确定帧结构中有效数据的个数。根据数据的相应周期，MCU 负责将准备好的 A、B 数据分别顺序放入 3 个双口 RAM 中相应的地址，待 3 个双口 RAM 都放好数据后，MCU 通过中断信号通知 3 个 DPU 分别读取各自双口 RAM 中当前拍的数据。

表 1 A、B 数据帧结构

相对地址	内容
0	数据标志
1	有效数据个数
2	数据 1
3	数据 2
…	…
n	数据 n

(2)姿态数据帧结构

对于姿态数据，MCU 负责将收到的指令数据放置到双口 RAM 中相应的位置，而 DPU 为防止姿态数据的漏

查，对于该数据的查询，采用 1 ms 周期服务程序查询，即每毫秒查询双口 RAM 中相应位置的姿态数据。

为了提高查询模块的执行效率并保证输入数据的实时有效性，对于姿态数据均采用如表 2 所示的帧结构。

表 2 陀螺数据帧结构

相对地址	内容
0	数据标志
1	有效数据个数
2	数据 1
3	数据 2
…	…
n	数据 *n*
n+1	YES/NO
n+2	ON/OFF

在表 2 中，“数据标志”字段指明结构中的数据属于哪类数据；“有效数据个数”字段用于查询时确定帧结构中有效数据的个数，提高查询的效率；“数据 1-数据 *n*”作为有效数据；“YES/NO”标志，用于 MCU 的 1 ms 周期服务程序和主程序；“ON/OFF”标志，用于 MCU 的 1 ms 周期服务程序和 DPU 的 1 ms 周期服务程序；标志“YES/NO”，与标志“ON/OFF”协同工作的目的在于输入数据同步。

2.4.2 数据输出

DPU 传给 MCU 的数据称为输出数据，包括各类遥测信息、串口查询返回帧信息、姿态控制信息(20 ms 周期)、3 个关键指令信息、惯导辅助设计帧信息、调零结果信息。其中，遥测信息又包括 14 个普通遥测信息和 7 个特征点遥测信息。与输入数据相比，输出数据信息量较大，并且除姿态控制信息为 20 ms 周期数据外其余的信息均不是周期性数据，因此传输过程中 MCU 可能无法及时处理 DPU 传来的数据，为防止数据丢失对于输出数据中的非周期性数据采用如图 2 所示的循环队列，MCU 端每毫秒查询一遍循环队列。

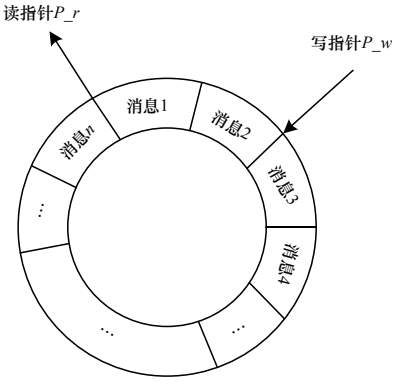

图 2 循环队列

在图 2 中，*P_r* 表示读指针；*P_w* 表示写指针；*P_r* 等于 *P_w*，表示队列空；*P_w* 加 1 等于 *P_r*，表示队列满；一个消息的数据放入结束后，*P_w* 加 1；读取一个消息后，*P_r* 加 1；*P_r* 不等于 *P_w*，表示有数据，一直进行取数据操作，直到数据取空为止；为了方便指针的移动队列中每个消息的长度是固定的，根据经验值每个消息中包含的数据个数不超过 32 个字(1 个字为 16 bit)，在此将消息长度定义为 50 bit×16 bit。每个消息的帧结构如表 3 所示。

表 3 循环队列帧结构

相对地址	内容
0	数据标志
1	有效数据个数
2	数据 1
…	…
n+1	数据 *n*
n+2	无效数据
…	无效数据
49	无效数据

循环队列帧结构中“数据标志”字段指明该消息属于哪类消息；“有效数据个数”字段用于明确帧结构中有效的数据的个数，便于查询时较快的获取有效数据；“无效数据”字段用于填充帧结构，在此使用零填充。

为了防止循环队列在传输过程中溢出，循环队列的大小定义为：循环队列长度=(DPU 传给 MCU 的最大消息数量×2+1)×每个消息帧的长度。其中“每个消息帧的长度”为表 2 所示的帧结构长度作为循环队列中读写指针移动的基本单位，大小固定不变；“DPU 传给 MCU 的最大消息数量”为输出数据中包含的所有消息的个数，然而实际传输的过程中所有的输出数据按照一定的时序被传输，不会同时被传输，因此上面定义的循环队列的大小完全满足传输过程中的需求。

对于输出数据中的姿态控制信息，由于其具有 20 ms 周期特性，并要求 MCU 每 20 ms 的第 3 ms 将 DPU 传来的姿态信息读取出去，为了提高查询效率，姿态控制信息不采用循环队列，而是采用如表 4 所示的帧结构，并将该结构对应到双口 RAM 的固定位置。

表 4 姿控信息帧结构

相对地址	内容
0	数据标志
1	有效数据个数
2	数据 1
…	…
n+1	数据 *n*
n+2	ON/OFF

在表 4 中“数据标志”字段标明信息帧的类别，“有效数据个数”字段标明帧结构中有效数据的个数，“ON/OFF”字段主要用于 DPU 通知 MCU 姿控信息是否为最新。MCU 每 20 ms 的第 3 ms 从双口 RAM 中读取姿控信息作为当前周期的数据，如果此时标志为“OFF”，则取上一周期的姿控信息；如果此时标志为“ON”，则取

最新数据发送并将标志置为“OFF”。

3 技术方案实现

3.1 开发环境

方案具体实现采用如表 5 所示的开发环境。

表 5 开发工具及语言

DSP 模块	IDE 工具	编程语言
MCU	CCS2000	C 语言，汇编语言
DPU	CCS3X-4X	C 语言，汇编语言

3.2 MCU 关键代码

MCU 关键代码如下：

```
while(1) {                  //ms 中断
if(((*index_p_r)!=(*index_p_w)) && (DPU_ok==ON)){
(*index_p_r) = ((*index_p_r)+1)%CNT_INF;     //读指针索引
if(No_DPU == DPU_1){
DPU_ID_cx =p_CHIP1_doubleRAM_baseAddr[0x1088+(*index _
p_r)*LEN_INF]&0xFFFF ;
DPU1_ID_Parse(DPU_ID_cx, (*index_p_r));     //DPU1 循环队列
//ID 解析    }
else if(No_DPU == DPU_2)       {
DPU_ID_cx = p_CHIP2_doubleRAM_baseAddr[0x1088+(*index_
p_r)*LEN_INF]&0xFFFF ;
DPU2_ID_Parse(DPU_ID_cx, (*index_p_r));     //DPU2 循环队列
//ID 解析    }
else if(No_DPU == DPU_3)       {
DPU_ID_cx = p_CHIP3_doubleRAM_baseAddr[0x1088+(*index_
p_r)*LEN_INF]&0xFFFF ;
DPU3_ID_Parse(DPU_ID_cx, (*index_p_r));     //DPU3 循环队列
//ID 解析    }
} //END of   IF
 else{break;}
}//END of WHILE
```

3.3 DPU 关键代码

DPU 关键代码如下：

```
asm(" push st ");              //保存状态寄存器
asm(" and 1FFFh,st ");      //关中断
g_index = (g_index+1) % CNT_INF ;
for(j=0;j<len_data;j++){
pg_VC33_doubleRAM_baseAddr[0x1088+g_index*LEN_INF+j]=
p_data[j] ;    //将信息写入循环队列}
pg_VC33_doubleRAM_baseAddr[0x1080] = g_index ;    //写写
//指针的索引号
asm(" pop st ");        //恢复状态寄存器
```

4 技术方案分析

对技术方案分析如下：

(1)循环队列空间为总消息个数的 2 倍加 1，可以避免消息的覆盖，从而不会出现丢拍问题。

(2)输入数据 A 及输入数据 B 通过双口中断通知 3 个 DPU，实时性高，并能解决 3 个 DPU 的输入数据同步问题。

(3)循环队列消息长度固定，读写指针只需做加 1 操作，即可指向下一条消息，从提高了队列查询的效率。

(4)循环队列中的数据标志和数据的实际长度一一对应，不需每次从内存中取“消息长度”个数据，而只是将实际有效数据取出，可以节省时间，在实际长度都很少的情况下，效率很高。

(5)DPU 将遥测数据通过不同的时间段均匀地传给 MCU，使得 MCU 在 ms 查询中的时间耗时均匀，从而保证 MCU 端任务时序的可靠性。

5 结束语

背景项目中箭载嵌入式计算机是作为航天器的飞行控制计算机存在的，在其工作的过程中，一旦控制中心(MCU)与运算中心(3DPU)之间的交互出现问题，则整个飞行控制流程会出现停步，最终导致飞行任务失败。由此可以看出控制中心与运算中心之间的通信对于整个飞行控制任务的成败起着至关重要的作用[8-9]。针对 MCU 与 3 个 DPU 的通信，本文提出的适用于“1+3”冗余架构的信息交互方案在满足输入数据同步的同时，能够有效地解决输出数据的丢拍问题，并且在实现高可靠性的同时，兼顾了执行效率问题。目前该方案已应用于背景项目的飞行控制软件中。

参考文献

[1] 盖玲兴. 航天分布式实时容错平台研究[D]. 西安: 西北工业大学, 2004.

[2] 李新明, 李 艺, 王 鹏, 等. 分布、实时、容错一体化设计方法研究[J]. 计算机工程, 2007, 33(18): 262-264.

[3] 王丽丽. 无人机余度飞行控制计算机关键技术研究[D]. 南京: 南京航空航天大学, 2009.

[4] 王群伟, 吴成富. 基于 SCADE 的无人机三余度飞控系统设计及实现[J]. 测控技术, 2007, 26(4): 52-54.

[5] 黄永勤, 金利峰, 刘 耀. 高性能计算机的可靠性技术现状与趋势[J]. 计算机研究与发展, 2010, 47(4): 589-594.

[6] 杨 雅. 星载嵌入式计算机系统可靠性技术研究[D]. 长沙: 国防科学技术大学, 2005.

[7] 李烈彪, 李 仙. 计算机系统的可靠性技术[J]. 计算机技术与发展, 2007, 17(11): 142-144.

[8] 杨 政, 华 伟, 张宗麟. 基于双口 RAM 的 ARINC429 航空数据总线收发系统设计[J]. 弹箭与制导学报, 2005, 25(2): 273-277.

[9] 李小青. 双口 RAM 在多 CPU 计算机测控系统中的应用[J]. 微计算机信息, 1999, 15(1): 54-56.

编辑 索书志