Science Discovery
2019; 7(4): 249-256
http://www.sciencepublishinggroup.com/j/sd
doi: 10.11648/j.sd.20190704.22
ISSN: 2331-0642 (Print); ISSN: 2331-0650 (Online)

Design and Implementation of a Heterogeneous Safety Critical Computer

Shi He, Zhao Deliang

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

Email address:
17120263@bjtu.edu.cn (Shi He), 16120283@bjtu.edu.cn (Zhao Deliang)

To cite this article:
Shi He, Zhao Deliang. Design and Implementation of a Heterogeneous Safety Critical Computer. *Science Discovery.* Vol. 7, No. 4, 2019, pp. 249-256. doi: 10.11648/j.sd.20190704.22

Received: June 21, 2019; **Accepted**: August 4, 2019; **Published**: August 27, 2019

Abstract: In recent years, the rapid development of high-speed railway in our country, followed by a rail safety problem has become the most important issue, as the core of the train operation control equipment, the development of the safety critical computer should be considered, firstly. At present, the mainstream safety critical computers all adopt the method of isomorphism, which can eliminate the multiple non-common fault of the same error structure, but can do nothing for some common fault. In this paper, a heterogeneous safety critical computer design and implementation method is proposed, which adopts the method of heterogeneous hardware and software, and analyzes its reliability and security by using fault tree model.

Keywords: Heterogeneous, Safety Critical Computer, Reliability, Security

一种异构安全计算机的设计与实现

石贺，赵得亮

北京交通大学电子信息工程学院，北京，中国

邮箱
17120263@bjtu.edu.cn（石贺），16120283@bjtu.edu.cn（赵得亮）

摘要：近年来，我国的高速铁路飞速发展，随之而来的铁路安全问题成为了重中之重，作为列车运控的核心装备，安全计算机的研制首当其冲。目前主流的安全计算机均采用的是同构的实现方式，可以排除相同差错结构的多重非共因故障，但是对于某些共因故障则束手无策。本文提出了一种异构的安全计算机设计与实现的方法，采用了硬件与软件均异构的方式，并利用故障树模型分析了其可靠性与安全性。

关键词：异构，安全计算机，可靠性，安全性

1. 引言

目前，我国高速铁路通车里程已达2.5万公里，随之而来的则是与人民出行息息相关的铁路安全问题。铁路安全问题与铁路信号系统紧密关联，在当下这个信息化、智能化的时代，计算机技术与通信技术已经成为铁路信号系统的基本支撑，而基于安全计算机的列车运行控制系统则成为铁路安全的根本保证[1-3]。

现主流的二乘二取二与三取二安全计算机平台在构成上均为同构模式[4,5]，具体而言，在硬件上采取的是相同的处理器架构，在软件上使用的是相同的操作系统，它们的功能作用完全相同，这样的静态冗余结构符合故障-安全原则，可以排除相同差错结构的多重非共因故障，但是对于某些共因故障则显得束手无策。

基于上述问题，本文着手于差异化安全计算机的研究，从硬件和软件两方面采取异构的方式，即采用架构不同的处理器，在处理器上运行不同的操作系统，以此来解决某些共因故障带来的安全性问题[6-10]。

具体而言，本文在硬件上采用了两种不同的处理器架构：Coldfire MCF54455和ARM ZYNQ，软件上同样采用了两种不同的实时操作系统：VxWorks和QNX[11-14]。下表中为本文所用到的三取二安全计算机平台三系主机的软硬件组成情况：

表1 三取二安全计算机平台三系主机软硬件构成。

	A系主机	B系主机	C系主机
硬件构成	ARM	ARM	Coldfire
软件构成	VxWorks	QNX	VxWorks

另外，本文对所用到的三取二异构安全计算机进行建模，并分析了其可靠性与安全性[15-16]。

2. 系统总体架构设计

本文所采用的三取二异构安全计算机平台主要可分为以下五大单元：供电单元、主机单元、内部通信单元、外部通信单元、输入/输出单元。系统总体架构设计图如图1所示：

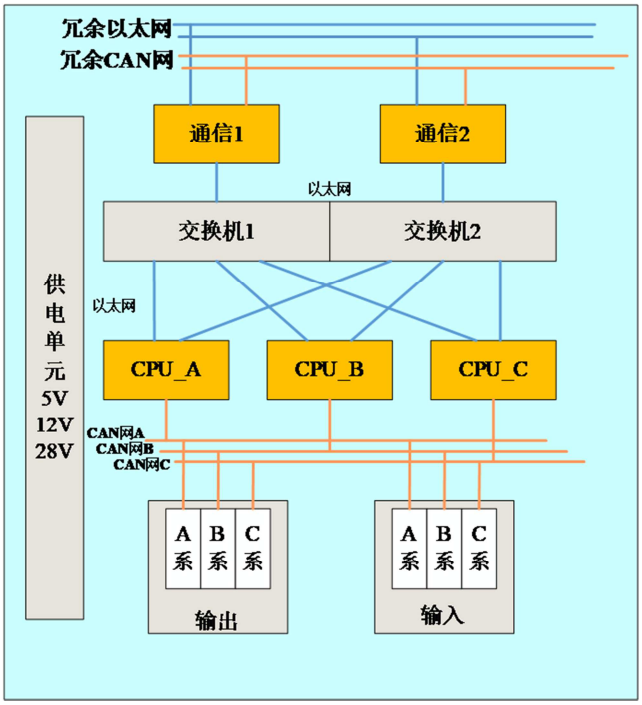

图1 三取二异构安全计算机平台系统架构图。

2.1. 供电单元

本平台的供电单元部分提供三种不同的电平，分别为5V、12V和28V，该单元由11个电源板组成，分别是5个5V电源，3个12V电源，3个28V电源。供电单元的总输入是外部的直流24V，供电单元采用DC-DC模块分别将24V转化为5V、12V和28V输出。

其中，第一块5V电源板向第一套交换机和通信机提供电源，第二块5V电源板向第二套交换机和通信机提供电源，第三块5V电源板向A系统的主机及IO板提供一路5V电源，第四块5V电源板向B系统的主机及IO板提供一路5V电源，第五块5V电源板向C系统的主机及IO板提供一路5V电源。

第一块12V电源板向A系统的I/O板提供一路12V电源（采集和驱动公用），第二块12V电源板向B系统的I/O板提供一路12V电源（采集和驱动公用），第三块12V电源板向C系统的I/O板提供一路12V电源（采集和驱动公用）。

第一块30V电源板向驱动板提供一路30V电源，第二块30V电源板向驱动板提供一路30V电源，第三块30V电源板向驱动板提供一路30V电源。

由于采用了上述供电方式，所以三取二异构安全计算机处理单元的各个模块之间实现了很好的电源隔离，避免了某个单元发生故障时对其他单元产生影响。

所用电源板卡的实物图如图2所示：

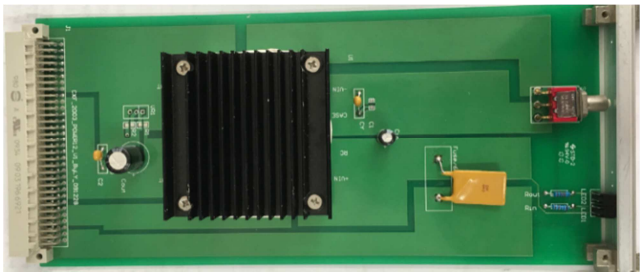

图2 电源板卡实物图。

2.2. 主机单元

主机单元由A、B、C三系组成，主要负责三取二平台的逻辑部分，对输入的数据首先进行同步，然后进行表决，最后输出。主机单元的结构图如图3所示：

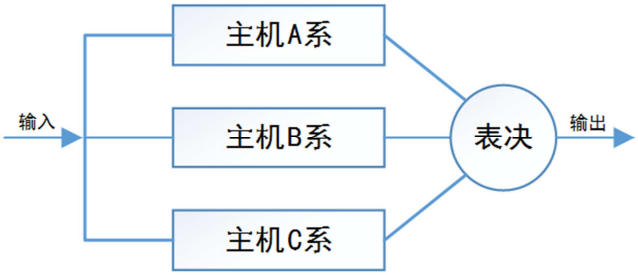

图3 主机单元结构图。

2.3. 内部通信单元

由图1的系统架构图可知，系统的内部通信单元分为以太网与CAN总线网络。

首先，A、B、C三系主机每系都有两路以太网，分别都接到了交换机1与交换机2上以构成冗余通道，这样就实现了系统内部的以太网通信。

其次，A、B、C三系主机与输入/输出单元之间采用了CAN总线通信的方式，A系主机与输入/输出单元中的A系组成了CAN网A，B系主机与输入/输出单元中的B系组成了CAN网B，C系主机与输入/输出单元中的C系组成了

CAN网C。通过CAN总线通信的方式，实现了主机单元与输入/输出单元数据的交互。

所用到的交换机板卡的实物图如图4所示：

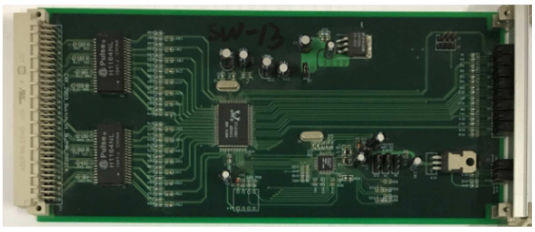

图4 交换机板卡实物图。

2.4．外部通信单元

由图1的系统架构图可知，交换机1与交换机2都通过以太网的方式连接到了通信机1与通信机2上，然后由通信机对外实现以太网通信。另外，通信机1与通信2上均有两路CAN总线网络，对外形成了一对冗余的CAN总线网络，实现外部的CAN总线通信。

2.5．输入输出单元

输入/输出单元由输入、输出板卡构成，每块输入、输出板卡上均有完全相同的A、B、C三系，用来与主机的三系进行CAN总线通信。

另外，每个输入板卡上提供了16路的输入供系统采集外部数据使用，每个输出板卡上提供了4路输出供系统驱动外部设备使用。

3．三系异构主机软硬件设计

3.1．异构硬件设计

本文在三系异构主机硬件设计上选取了两种不同的硬件架构：Coldfire V4架构和ARM ZYNQ架构，对应的主CPU芯片分别为：MCF54455和XC7Z020CLG484I-1。

3.1.1．基于Coldfire架构的硬件设计

对应于本文的接口需求，基于Coldfire架构的硬件设计可分为以下几部分：电源部分、DDR及FLASH、4路串口、2路CAN、2路以太网、复位及实时时钟。

电源部分：基于Coldfire架构的MCF54455板卡需要三种不同电平（3.3V、1.8V、1.5V）的电源，由于输入电压为DC5V，所以需要用到电平转换芯片，本文中所用的电平转换芯片为PTH04070W，通过改变对应电阻的值，使输出电压为所需要的设定值。

DDR及FLASH：由于主机板卡需要运行操作系统，故需要DDR及FLASH来为系统提供内存及存储空间。本文中DDR部分采用了两片型号为MT47H64M8的DDR2 SDRAM，每一片DDR2的存储空间为64MB，且数据线为8位，本文将两片DDR2通过位扩展的方式得到了16位数据位；FLASH采用了Intel的TE28F256J3C，同样使用了两片，并通过级联的方式得到了64M的存储空间。

4路串口：由于系统仅自带3路串口，不满足本文的使用需求，所以使用CPU的Flex Bus总线扩展4路串口（2路RS232、2路RS422），通过外接SC16C554DB将并行总线接口转换为串行总线接口，再通过电源隔离芯片与信号隔离芯片，最后由串口协议芯片实现串口数据的收发功能。

2路CAN：2路CAN总线是通过CPU提供的SPI接口转换得来的，通过带有SPI功能的CAN控制器芯片得到CAN信号，然后经过电源隔离芯片与信号隔离芯片，最后由CAN收发器芯片完成CAN通信的功能。

2路以太网：2路以太网是由CPU提供的RMII接口实现的，通过外接PHY芯片与网络变压器实现以太网通信。

复位及实时时钟：复位功能采用了SP706芯片来实现，不仅有看门狗功能还带有电平阈值检测功能；实时时钟则采用了DS1374芯片，此芯片带有I2C通信功能。

基于Coldfire架构的MCF54455板卡实物图如图5所示：

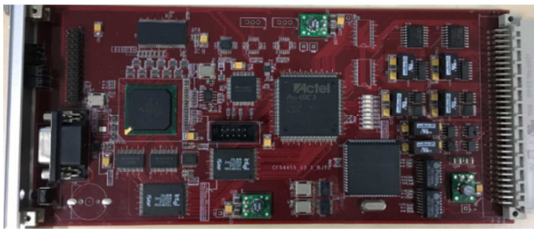

图5 基于Coldfire架构的MCF54455板卡实物图。

3.1.2．基于ARM ZYNQ架构的硬件设计

同样地，对于ARM ZYNQ架构的硬件来说接口需求仍是不变的，故其设计可分为以下几部分：电源部分、DDR及EMMC、4路串口、2路CAN、2路以太网、MIO及FPGA。基于ARM ZYNQ架构的硬件设计基本思路与Coldfire架构的类似，下面仅把两者不同之处作一简单说明：

电源部分：基于ARM ZYNQ架构的ZC702板卡需要四种不同电平（3.3V、1.8V、1.5V、1.0V）的电源，且ZC702主芯片对电源的上电顺序有着严格要求，故本文所用到的电平转换芯片为TPS62130，此芯片提供了一个PG（POWER GOOD）引脚，将上一级电源芯片的PG引脚连接至下一级电源芯片的使能引脚则可满足ZC702对电源时序的要求。

DDR及EMMC：基于ARM ZYNQ架构的ZC702板卡使用了两片DDR3，型号为MT41K128M16，每片DDR3有16根数据线，同样将两片DDR3进行位扩展得到32位数据位；EMMC采用的型号为MTFC8GACAAAM-1M。

MIO及FPGA：此部分均为一些接口。

基于ARM ZYNQ架构的ZC702板卡实物图如图6所示：

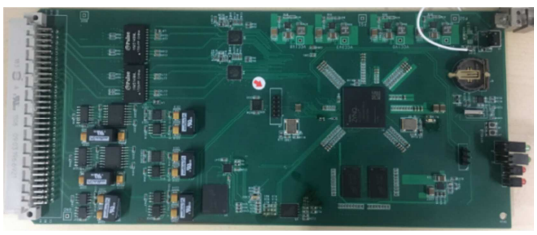

图6 基于ARM ZYNQ架构的ZC702板卡实物图。

3.2．异构软件设计

3.2.1．C系板卡基于VxWorks的系统移植

基于Coldfire架构的MCF54455板卡移植VxWorks操作系统需要两个文件：bootrom.bin和VxWorks镜像。

首先，bootrom.bin为Coldfire架构所需的Bootloader，用于启动引导加载内核，一般来说，在Wind River Workbench当中打开workbench development shell，进入目标板所在目录，直接make编译即可生成bootrom.bin。

然后，配置VxWorks BSP包中的时钟、网卡等信息，按需求修改Kernel Configuration中的模块，最后编译工程即可生成VxWorks镜像。

最后，将bootrom.bin通过BDI2000烧写到某个固定的地址处，通过tftp网络的方式加载VxWorks镜像，移植成功后的界面如图7所示：

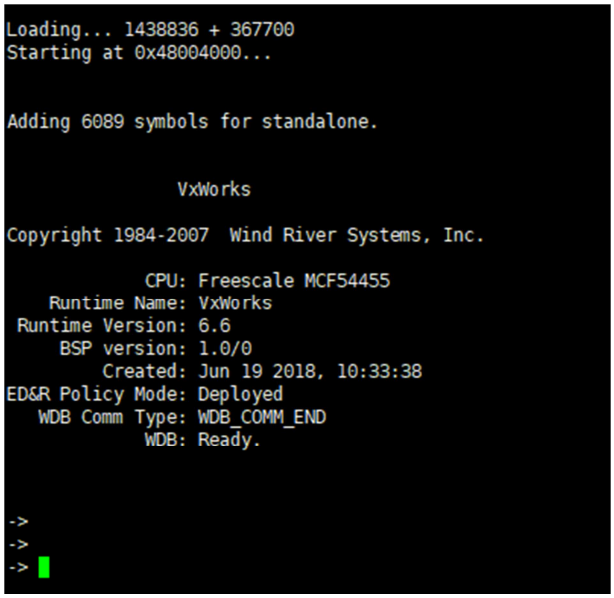

图7 C系板卡基于VxWorks的系统移植界面。

3.2.2．B系板卡基于QNX的系统移植

基于ARM ZYNQ架构的ZC702板卡移植QNX操作系统需要两个文件：BOOT.bin和QNX镜像。

XC7Z020CLG484I-1芯片为双ARM核，分为PS（Processing System）和PL（Programming Logic）两个部分，故制作BOOT.bin文件需要PL部分的比特流文件以及PS部分的.elf文件，此外，还需要处理这两个文件的代码，即FSBL.elf文件。得到这三个文件之后利用Xilinx公司的Vivado软件创建启动镜像，得到BOOT.bin。

打开QNX配套的开发环境QNX Momentics IDE 5.0，导入QNX工程，修改相关的时钟、网卡、串口时钟等配置，编译生成QNX镜像。

对SD卡进行分区并制作文件系统后，将BOOT.bin和QNX镜像拷贝到SD卡当中，然后ZC702板卡通过SD卡启动，与系统进行交互，正确加载对应的内核，启动成功的界面如图8所示：

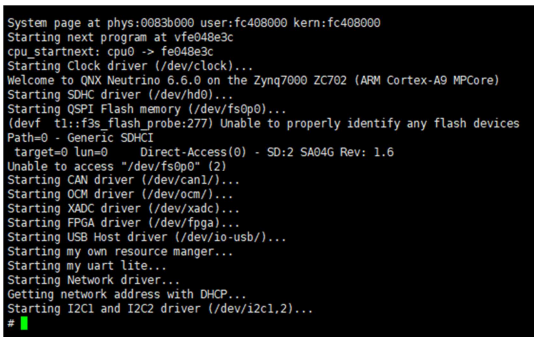

图8 B系板卡基于QNX的系统移植界面。

3.2.3．A系板卡基于VxWorks的系统移植

基于ARM ZYNQ架构的ZC702板卡移植VxWorks操作系统同样也需要两个文件：BOOT.bin和VxWorks镜像。

生成上述两个文件的方法与前两系板卡类似，在此不再赘述，基于ARM ZYNQ架构的ZC702板卡移植VxWorks操作系统成功的界面如图9所示：

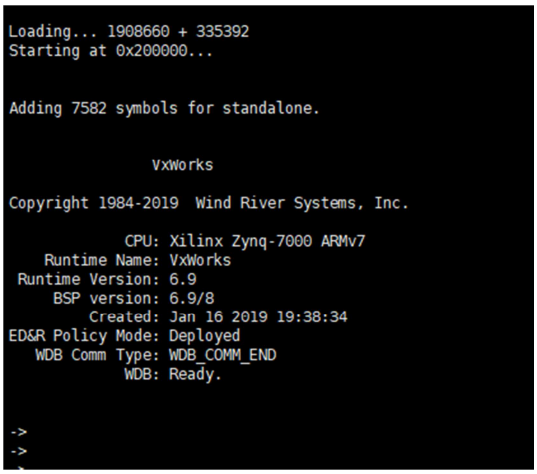

图9 A系板卡基于VxWorks的系统移植界面。

4．异构安全计算机系统可靠性与安全性分析

4.1．系统可靠性与安全性模型

为对本文的异构安全计算机进行可靠性与安全性建模，现假设如下条件对于本系统成立：

（1）由相同结构单元组成的系统存在共因失效，由不同单元组成的系统不存在共因失效；

（2）采用特殊维修策略，系统失效才进行维修，且系统的修复时间服从参数为 μ 的指数分布；

（3）不考虑硬件表决及平台软件的可靠性问题，假设硬件表决器和平台软件是完全可靠的；

（4）采用相似模型，组成系统的冗余单元结构完全相同，承受相似的共因失效冲击，相同分布的单元同时故障的失效率相同；

基于上述假设及异构三取二安全计算机基本单元的相互逻辑关系，以平台故障作为顶事件，可得到不考虑共因失效下系统的故障树模型，如图10所示，图中故障树的

顶事件、中间事件、以及基本事件的符号及含义如表2所示。

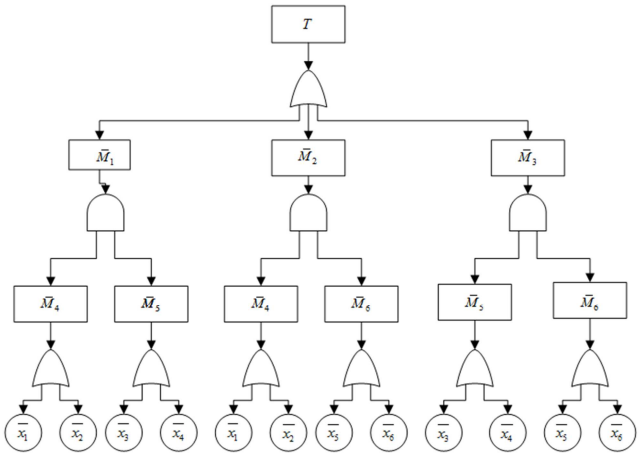

图10 异构安全计算机故障树。

表2 异构安全计算机故障树事件及含义。

事件分类	事件符号	符号表示含义
顶事件	T	异构安全计算机发生故障
中间事件	$\overline{M_1}$	AB两系发生故障
中间事件	$\overline{M_2}$	AC两系发生故障
中间事件	$\overline{M_3}$	BC两系发生故障
中间事件	$\overline{M_4}$	A系发生故障
中间事件	$\overline{M_5}$	B系发生故障
中间事件	$\overline{M_6}$	C系发生故障
基本事件	$\overline{x_1}$	A系处理器故障
基本事件	$\overline{x_2}$	A系软件故障
基本事件	$\overline{x_3}$	B系处理器故障
基本事件	$\overline{x_4}$	B系软件故障
基本事件	$\overline{x_5}$	C系处理器故障
基本事件	$\overline{x_6}$	C系软件故障

将图10中的与门改成或门，或门改成与门，底事件和顶事件改为其对应的逆事件，则可得原异构安全计算机故障树的对偶树，其结构如图11所示：

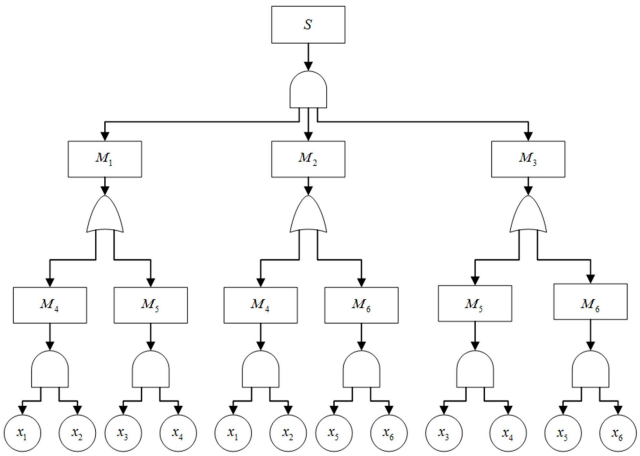

图11 异构安全计算机故障树的对偶树。

利用上行法求解异构安全计算机平台的对偶故障树的最小割集，从而求得原故障树的最小路集。

对偶树的最底层为：

$$\begin{cases} M_4 = x_1 x_2 \\ M_5 = x_3 x_4 \\ M_6 = x_5 x_6 \end{cases} \tag{1}$$

往上一层为:

$$\begin{cases} M_1 = M_4 + M_5 = x_1 x_2 + x_3 x_4 \\ M_2 = M_4 + M_6 = x_1 x_2 + x_5 x_6 \\ M_3 = M_5 + M_6 = x_3 x_4 + x_5 x_6 \end{cases} \tag{2}$$

顶层为：

$$S = M_1 M_2 M_3 = (x_1 x_2 + x_3 x_4)(x_1 x_2 + x_5 x_6)(x_3 x_4 + x_5 x_6) \tag{3}$$

根据布尔代数化简得：

$$S = x_1 x_2 x_3 x_4 + x_1 x_2 x_5 x_6 + x_3 x_4 x_5 x_6 \tag{4}$$

故系统的最小路集为：$\{x_1 x_2 x_3 x_4\}$，$\{x_1 x_2 x_5 x_6\}$，$\{x_3 x_4 x_5 x_6\}$，最小路集不交化得：

$$S = x_1 x_2 x_3 x_4 + x_1 x_2 \overline{x_3} x_5 x_6 + x_1 x_2 x_3 \overline{x_4} x_5 x_6 + \overline{x_1} x_3 x_4 x_5 x_6 + x_1 \overline{x_2} x_3 x_4 x_5 x_6 \tag{5}$$

则系统的可靠度表达式为：

$$\begin{aligned} R(\mathrm{t}) = {} & R_1(t)R_2(t)R_3(t)R_4(t) + R_1(t)R_2(t)F_3(t)R_5(t)\mathrm{R}_6(t) + \\ & R_1(t)R_2(t)R_3(t)F_4(t)R_5(t)R_6(t) + F_1(t)R_3(t)R_4(t)R_5(t)R_6(t) + \\ & R_1(t)F_2(t)R_3(t)R_4(t)R_5(t)R_6(t) \end{aligned} \tag{6}$$

由公式（7）对（6）式进行化简：

$$R_i + F_i = 1 \tag{7}$$

化简结果为：

$$R(t) = R_1 R_2 R_3 R_4 + R_1 R_2 R_5 R_6 + R_3 R_4 R_5 R_6 - 2R_1 R_2 R_3 R_4 R_5 R_6 \tag{8}$$

底事件 x_1 和 x_3 为同硬件架构处理器，且服从相同分布、承受相同的共因失效冲击，底事件 x_2 和 x_6 采用相同的操作系统，服从相同分布、承受相同的共因失效冲击。假设硬件Zynq-7020的一阶失效率为 λ_{b1}，二阶失效率为 λ_{b2}，操作系统VxWorks的一阶失效率为 λ_{c1}，二阶失效率为 λ_{c2}，操作系统QNX的失效率为 λ_4 ,硬件MCF54455的失效率为 λ_5。

利用隐式替代法，令

$$R_1(t) = R_3(t) = B(t)$$

$$R_2(\mathrm{t}) = R_6(\mathrm{t}) = C(\mathrm{t})$$

则系统可靠度表达式可进一步化简为：

$$R(t)=-2R_4(t)R_5(t)B^2(t)C^2(t)+R_4(t)B^2(t)C(t)+R_5(t)B(t)C^2(t)+R_4(t)R_5(t)B(t)C(t) \quad (9)$$

其中，

$$\begin{cases} R_4(t)=e^{-\lambda_4 t} \\ R_5(t)=e^{-\lambda_5 t} \\ B(t)=e^{-(\lambda_{b1}+\lambda_{b2})t} \\ B^2(t)=e^{-(2\lambda_{b1}+\lambda_{b2})t} \\ C(t)=e^{-(\lambda_{c1}+\lambda_{c2})t} \\ C^2(t)=e^{-(2\lambda_{c1}+\lambda_{c2})t} \end{cases} \quad (10)$$

因此考虑共因失效时异构安全计算机平台的可靠度表达式为：

$$R(t)=-2e^{-(\lambda+\lambda_{b1}+\lambda_{c1})t}+e^{-(\lambda+\lambda_{b1}-\lambda_5)t}+e^{-(\lambda-\lambda_4+\lambda_{c1})t}+e^{-\lambda t} \quad (11)$$

其中，

$$\lambda=\lambda_4+\lambda_5+\lambda_{b1}+\lambda_{b2}+\lambda_{c1}+\lambda_{c2} \quad (12)$$

系统的故障概率密度函数为：

$$f(t)=-\frac{d[R(t)]}{dt}=-2(\lambda+\lambda_{b1}+\lambda_{c1})e^{-(\lambda+\lambda_{b1}+\lambda_{c1})t}+(\lambda+\lambda_{b1}-\lambda 5)e^{-(\lambda+\lambda_{b1}-\lambda 5)t}+(\lambda-\lambda_4+\lambda_{c1})e^{-(\lambda-\lambda_4+\lambda_{c1})t}+\lambda e^{-\lambda t} \quad (13)$$

经拉普拉斯变换后得：

$$F(s)=L[f(t)]=-\frac{2(\lambda+\lambda_{b1}+\lambda_{c1})}{s+\lambda+\lambda_{b1}+\lambda_{c1}}+\frac{(\lambda+\lambda_{b1}-\lambda_5)}{s+\lambda+\lambda_{b1}-\lambda_5}+\frac{(\lambda-\lambda_4+\lambda_{c1})}{s+\lambda-\lambda_4+\lambda_{c1}}+\frac{\lambda}{s+\lambda} \quad (14)$$

由于系统的修复时间服从参数为 μ 的指数分布，则系统的修复概率密度函数为：

$$m(t)=\mu e^{-\mu t} \quad (15)$$

其拉氏变换为：

$$G(s)=L[m(t)]=\frac{\mu}{s+\mu} \quad (16)$$

联立公式：

$$Q(s)=\frac{F(s)[1-G(s)]}{s[1-F(s)G(s)]} \quad (17)$$

得到系统得不可用度表达式得拉氏变换为：

$$Q(s)=\frac{\left[-\frac{2(\lambda+\lambda_{b1}+\lambda_{c1})}{s+\lambda+\lambda_{b1}+\lambda_{c1}}+\frac{(\lambda+\lambda_{b1}-\lambda_5)}{s+\lambda+\lambda_{b1}-\lambda_5}+\frac{(\lambda-\lambda_4+\lambda_{c1})}{s+\lambda-\lambda_4+\lambda_{c1}}+\frac{\lambda}{s+\lambda}\right]\left(1-\frac{\mu}{s+\mu}\right)}{s\left\{1-\left[-\frac{2(\lambda+\lambda_{b1}+\lambda_{c1})}{s+\lambda+\lambda_{b1}+\lambda_{c1}}+\frac{(\lambda+\lambda_{b1}-\lambda_5)}{s+\lambda+\lambda_{b1}-\lambda_5}+\frac{(\lambda-\lambda_4+\lambda_{c1})}{s+\lambda-\lambda_4+\lambda_{c1}}+\frac{\lambda}{s+\lambda}\right]\frac{\mu}{s+\mu}\right\}} \quad (18)$$

则系统的可用度为：

$$A(t)=1-L^{-1}[Q(s)] \quad (19)$$

4.2．系统可靠性与安全性分析

由于异构安全计算机相对于同构安全计算机参数差异很大，为了使其具有可比性做如下假设：

（1）异构三取二安全计算机三系的一阶失效率是相同的，且对于异构安全计算机的每一系其硬件架构和软件系统构成的整体串联系统的独立可靠性与原同构系统的一阶失效率参数相同，即一阶失效率满足：

$$\begin{cases} \lambda_1=\lambda_{b1}\lambda_{c1} \\ \lambda_1=\lambda_{b1}\lambda_4 \\ \lambda_1=\lambda_5\lambda_{c1} \end{cases} \quad (20)$$

（2）由于本文设计的异构安全计算机只有AB两系（硬件架构不同）与AC两系（操作系统不同）存在共因失效的问题，对于二阶失效率来说，可认为与原同构系统的二阶失效率相同，此时有：

$$\begin{cases} \lambda_2=\lambda_{b2} \\ \lambda_2=\lambda_{c2} \end{cases} \quad (21)$$

基于上述假设，在不可维修条件下，按照上述规则，选取同构安全计算机系统的一阶失效率为 $\lambda_1=10^{-5}/h$，二阶失效率为 $\lambda_2=10^{-6}/h$，三阶失效率为 $\lambda_3=0$，维修率 $\mu=0$，选取异构安全计算机的参数 $\lambda_{b1}=5*10^{-6}/h$，$\lambda_{b2}=10^{-6}/h$，$\lambda_{c1}=5*10^{-6}/h$，$\lambda_{c2}=10^{-6}/h$，$\lambda_4=5*10^{-6}/h$，$\lambda_5=5*10^{-6}/h$，维修率 $\mu=0$。

仿真得出异构安全计算机平台与同构安全计算机平台可靠度对比曲线，如图12所示：

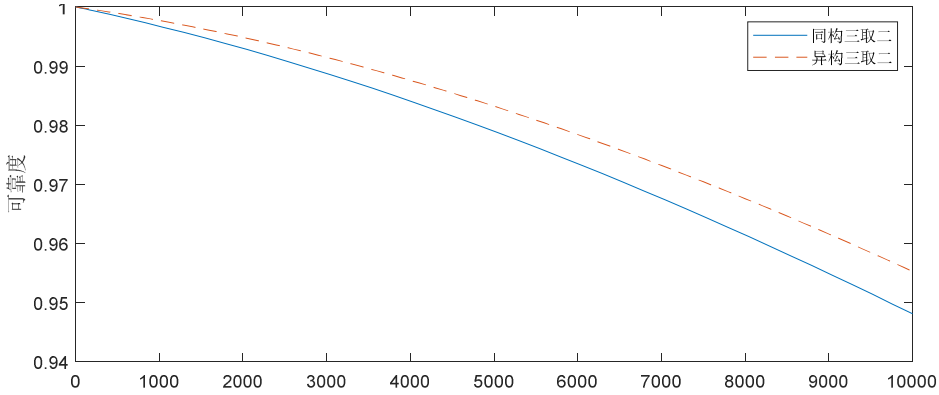

图12 同构与异构安全计算机可靠度对比图。

从图12中可以看出，在不可维修的情况下，随着时间的推移，同构安全计算机和异构安全计算机的可靠度都呈下降趋势，但异构三取二安全计算机平台在与同构三取二安全计算机平台参数近乎相同的情况下，可靠度略高于同构安全计算机系统。

5. 结论

本文介绍了一种异构三取二安全计算机平台的设计与实现。首先，由现在主流安全计算机均采取同构模式，其对某些共因故障不能有效并及时排除引出本文的研究意义。其次，本文介绍了系统的总体架构，大致可分为五个部分。然后，分别介绍了异构三取二安全计算机平台硬件与软件的具体设计方法与流程。最后，利用故障树的方法对系统的可靠性与安全性进行建模，并分析得出异构安全计算机在参数近乎相同的情况下可靠度略高于同构安全计算机的结论，为异构安全计算机的进一步研究提供了便利。

致谢

本文为国家“十三五”重点研发计划《中速磁浮运行控制系统关键技术研究及装备研制》(2016YFFB200602-26)的阶段性成果之一。

参考文献

[1] 唐俊同. 轨道交通信号系统安全计算机浅谈[J]. 机车电传动, 2011(6):73-75.

[2] Kim H, Lee H, Lee K. The design and analysis of AVTMR (all voting triple modular redundancy) and dual–duplex system[J]. Reliability Engineering & System Safety, 2005, 88 (3): 291-300.

[3] Hwang J G, Jo H J, Jeong R G. Analysis of safety properties for vital system communication protocol [C]// International Conference on Electrical Machines and Systems. IEEE, 2007: 1767-1771.

[4] 黄涛, 陈祥献, 黄海. 基于三取二冗余结构的安全计算机系统[J]. 计算机工程, 2011, 37(18):254-257.

[5] 刘真. 一种三取二安全计算机系统的设计与实现[J]. 铁路计算机应用, 2016, 25 (11):49-52.

[6] Ferdous R, Khan F, Sadiq R, et al. Fault and event tree analyses for process systems risk analysis: uncertainty handling formulations. [J]. Risk Analysis, 2011, 31 (1): 86–107.

[7] 马婷. 二乘二取二安全计算机内部安全通信机制的设计与实现[D]. 西南交通大学, 2016.

[8] 刘晨阳. TYJL-ADX型二乘二取二计算机联锁系统的优越性[J]. 技术与市场, 2012(10):26-27.

[9] 张海波. 分布式异构三取二安全控制单元的设计[D]. 浙江大学生物医学工程与仪器科学学院 浙江大学, 2010.

[10] Oster D, Kumada M, Zhang Y. Evacuated tube transport technologies (ET3) tm: a maximum value global transportation network for passengers and cargo[J]. Journal of Modern Transportation, 2011, 19 (1): 42-50.

[11] 邹玉龙, 刘彬, 田小莉,等. 基于VxWorks新型映像的三模冗余启动机制研究[J]. 计算机测量与控制, 2017, 25(8):120-122.

[12] 盛华, 刘书刚, 葛树俊. 基于QNX与Cortex-A8的CAN通信[J]. 计算机应用, 2015(a02):20-23.

[13] Qian Z, Huang H. Design and implementation of Linux network computer system based on Loongson Mipsel architecture [C]// International Conference on Computer Science and Service System. IEEE, 2011: 1209-1212.

[14] Sun L, Peng X, Zhu J, et al. A TRM Control System Designed by Loongson MCU [C]// Control Conference, 2008. CCC. IEEE, 2008: 777-779.

[15] 卢宏康, 曹源, 马连川. 基于动态故障树的异构安全计算机系统共模故障分析研究[J]. 铁路计算机应用, 2017(9).2

[16] 陈仁龙. 故障树分析计算方法[J]. 科技创新与应用, 2018, No.244(24):114-115.