

基于 FPGA 的双 DSP 冗余设计

何广龙

院（系）：控制科学与工程系

专 业：自动化

学 号：1110410421

指导教师：王强

2015 年 6 月 17 日

哈爾濱工業大學

畢業設計（論文）

題 目 基於 FPGA 的雙 DSP 冗餘設計

專 業 自動化

學 號 1110410421

學 生 何廣龍

指 導 教 師 王 強

答 辯 日 期 2015 年 6 月 29 日

摘 要

因为外部环境中的某些因素，会造成微控制器内的程序发生跑飞、死机等异常情况。而对于一些至关重要的控制系统，这些异常必须进行处理。

为此本文提出了一种利用双模冗余技术搭建控制系统的方法，在主控制器发生故障时，完成快速、无缝的切换，以保证算法和控制的连续性；切换完成后尝试对失效的控制器进行复位，提高系统的续航能力。

首先，对硬件冗余结构进行对比，选择了双机旁联模式。然后，对备份方式进行对比，选择了热备份方式。接着，选择合适的状态检测方式，即翻转输出管脚产生特定频率的方波。再根据实际需要，DSP 选择 TMS320F2812, FPGA 选择 XCV300-4PQ240I。最后，综合考虑硬件资源、传输速率和数据量，选择基于 FPGA 进行并行通讯的双口 RAM。

接着，在已完成的硬件实物上进行各功能模块的实现。包括在 FPGA 上构建双口 RAM；在 DSP 上产生状态脉冲；FPGA 对状态脉冲进行检测；传输与存储协议的设计；注入故障的设计。

最后，注入故障，进行可靠性验证。实际测试表明，在主控制器发生故障时能完成切换和复位操作。

关键词：双模冗余；DSP；FPGA；双口 RAM

Abstract

Certain factors in the external environment will result in runaway, crashes and other anomalies in the microcontroller. For some critical control systems, these exceptions must be solved.

So this paper proposes the use of a dual redundant fault-tolerant technique to build a redundant control system. When the main controller fails, the system will complete the fast and seamless handover function to ensure continuity of control and algorithms. After the completion of the handover, the system will attempt to reset the failed controller to improve endurance capacity of the system.

Firstly, hardware redundancy structure are compared theoretically, and duplication bypass redundancy is chosen. Then, backup methods are analyzed and hot backup is chosen. Next, one appropriate state detection method is selected that produces a square wave with specific frequency by flipping the output pin. After that, according to actual needs, DSP selects TMS320F2812 and FPGA selects XCV300-4PQ240I. Finally, considering hardware resources, the amount of data and transfer rate of data, dual port RAM, a kind of parallel communication based on FPGA is selected.

Secondly, on the completed hardware the functional modules are achieved. This step includes building a dual-port RAM on FPGA, DSP's state pulse generation, FPGA's state of the pulse detection, designing transmission and storage protocols and designing of fault injection.

Finally, fault will be injected to test the reliability of the system. Practical tests show that if the primary controller fails, the system can perform switch and reset operation successfully.

Keywords: TMR,DSP,FPGA,dual port RAM

目录

摘 要	I
ABSTRACT	II
第 1 章 绪 论	1 -
1.1 课题背景及研究的目的和意义	1 -
1.1.1 课题来源	1 -
1.1.2 研究的目的和意义	1 -
1.2 国内外在该方向的研究现状及分析	2 -
1.3 本文的主要研究内容	3 -
1.4 本文组织结构安排	4 -
第 2 章 总体方案设计	5 -
2.1 冗余结构的选择	5 -
2.2 备份方式的选择	6 -
2.3 系统需求分析	7 -
2.3.1 功能需求分析	7 -
2.3.2 性能需求分析	8 -
2.4 DSP 状态检测方式的选择	9 -
2.5 硬件选型	9 -
2.6 通讯方式的选择	10 -
2.7 本章小结	11 -
第 3 章 功能模块实现	12 -
3.1 利用 FPGA 实现双口 RAM	12 -
3.2 同步信号发生	15 -
3.3 DSP 的状态脉冲发生	16 -
3.4 FPGA 的状态脉冲检测	18 -
3.5 传输与存储协议	19 -
3.6 硬件电路的改装与设计	22 -

3.7 注入故障的设计.....	- 22 -
3.8 本章小结.....	- 23 -
第 4 章 系统调试与测试.....	- 24 -
4.1 系统调试流程.....	- 24 -
4.2 系统功能测试.....	- 24 -
4.2.1 正常逻辑测试.....	- 24 -
4.2.2 切换逻辑测试.....	- 27 -
4.2.3 复位逻辑测试.....	- 30 -
4.3 本章小结.....	- 32 -
结 论.....	- 33 -
参考文献.....	- 34 -
哈尔滨工业大学本科毕业设计（论文）原创性声明.....	- 36 -
致 谢.....	- 37 -

第1章 绪 论

1.1 课题背景及研究的目的和意义

1.1.1 课题来源

课题来自伺服随动电路板设计中的控制器冗余设计部分，是研究所与高校的合作项目中的一部分。该研究所开发了一套稳定回路控制系统，并将用于我国某型号飞行器上。但该控制系统在实际使用过程中发现微处理器会偶尔出现程序跑飞和死机的情况，然后导致整个系统瘫痪。必须在飞行器上使用之前处理这个异常情况。

本课题就是针对该问题进行的探索，设计了硬件双余度热备份 DSP 冗余结构，使用双模冗余技术，用双控制器替代原有的单控制器进行工作。这样在主控制器出现故障时，能确保快速、无缝的切换到从控制器，继续完成对稳定回路的伺服控制。

1.1.2 研究的目的和意义

在军事、航空航天、工业控制等领域中，经常需要控制系统在外空间、工业现场、汽车等场所工作。在这些复杂的环境中工作时，控制系统可能会因为电磁干扰、震动、大温差等发生故障。但是这些领域又要求控制系统必须有较强的鲁棒性，能够可靠、连续运行，所以需要采取措施提高系统的容错能力。与此同时，随着微控制器的迅速发展，越来越多的微控制器在控制系统中被使用，而微控制器作为高集成度的电子器件，对上述所列的多种干扰敏感，所以对微控制器进行容错保护就很有必要，否则可能出现不堪设想的后果。例如我国在 1990 年 9 月 3 日发射的风云一号（B）气象卫星，因为发射后恰遇太阳黑子活动峰年，但主控计算机的软硬件的异常处理能力都有限，所以在受到宇宙空间内的高能粒子辐射后发生多次单粒子翻转现象，导致姿态控制失效，且不能重新捕获姿态，仅仅工作了 165 天^[1]。类似事故还有很多，造成了大量人力、物力、财力上的消耗，这也让人们意识到在控制系统中加入容错设计的重要性。

在本项目中，为了避免惯性平台的控制器在工作中遇到程序跑飞、死机、掉电等异常情况时，系统无法完成功能，选择采用硬件双余度热备份 DSP 冗余设计，增强系统的可靠性。当主控制器发生异常时，控制系统被快速、无缝的自动切换

到从控制器的控制下，保证功能的连续性；同时控制系统会尝试修复已故障的控制器，提高系统的续航能力，从而从硬件上提高系统的容错能力。因为 FPGA 的可靠性高于 DSP^[2]，故用 FPGA 实现用于检错和切换的硬件电路和控制逻辑。

本文研究的硬件双余度热备份 DSP 冗余控制系统，可以解决已有的稳定控制回路系统偶尔出现的异常问题，极大地提高现有系统的可靠性，且具有较高独立性，可以很方便的进行移植以代替其他出现类似问题的单控制器控制系统，具有一定的工程意义。

1.2 国内外在该方向的研究现状及分析

容错控制系统是指，如果在闭环控制系统运行过程中，传感器、执行器甚至部分控制器或其他部件发生故障，但是控制系统仍能保持稳定，并保持较理想的控制效果，就能将此闭环控制系统称为容错控制系统^[3]。自 1986 年正式提出容错控制(Fault Tolerant Control, FTC)的概念以来，容错控制在其后迅速发展。

冗余是实现容错控制的一种有效方式，冗余技术是一门涉及控制理论、信号处理、通讯技术等多领域的综合学科。目前冗余技术主要应用在一些对系统可靠性要求很高的场合，例如航空航天、军事、核电等，且技术日趋成熟，应用也越开越广泛。冗余技术开始兴起时主要使用三模冗余（TMR）提高系统可靠性，但是由于其需要消耗额外的硬件与功率，还会影响工作速度^[4]，所以在部分领域逐渐由其它技术所替代。国外的冗余容错技术出现较早，应用场合也较多。随着制造业的进步和技术的发展，国外现在已经完成了芯片级的 TMR 实现，在对抗单粒子翻转效应方面有了很大提升^[5]。在芯片级的冗余中最出名的是欧洲航天局数次运用到实际星载计算机系统上的 LEON 系列处理器。该系列处理器由 Gaisler Research 公司设计，在其内部采用了三模冗余技术和纠错检错技术，使控制系统不仅保证了可靠性，同时还缩小了体积和质量^[6]。而 G.Buja、J.R.Pimentel 和 A.Zuccollo 在设计基于双通道的冗余 TTCAN 总线系统中通信模块的硬件电路时，采用了双路 TTCAN 总线的冗余方式，利用模块级的冗余设计提高系统可靠性^[7]。在计算机级（板级）设计中使用冗余技术的佼佼者有 Compaq 公司的 Trucluster 系统和喜马拉雅（Himalaya，原天腾公司 Tandem Co.）系列，它们都采用了双机热备份的方式保证系统可靠性^[8]。由此可见，国外的冗余技术已经发展的比较成熟，从板级到模块级再到芯片级都有成功应用。而国内在这一领域起步较晚，特别是能用于商用的成果不多，但在航天、核能等领域都取得了较好效果。如国家海洋局的第一颗用于海洋水色探究的实验型业务卫星海洋一号，就搭载的使用双模冗余结构的

星载计算机^[9]。国内在模块级的冗余设计上也有成功先例，如哈尔滨工业大学的吴钊君利用两片 DSP 以及之间的 McBSP 进行通讯从而完成了模块级的冗余结构设计，并成功应用到惯性平台上^[10]。

从已有的研究文献看，现阶段实现双控制器冗余技术的工程主要面临以下两个难点：1、同步技术；2、无缝切换技术。

解决同步问题是进行控制器冗余设计的最基本的要求。只有实现了主从控制器之间的同步，才能保证控制算法的连续和逻辑的有序。主从控制器之间的同步按精细程度可以分为时钟级同步和任务级同步两种。时钟级同步主要通过硬件设计来实现，常用锁相环技术，确保两个控制器的驱动时钟完全同步^[11]。任务级同步则主要通过软件设计来实现，通过主从控制器之间的通讯来进行同步，目前在冗余控制被广泛采用的通讯方式主要有 CAN 总线^[12]和高速串行总线^[13]。

无缝切换技术主要是在一些对实时性要求比较高的系统中需要用到。这是因为有些算法会使用过去的的数据作为参数进行计算，如积分运算和遗传算法。为了避免控制器切换导致参数丢失进而系统发生震荡的情况，就提出了无缝切换的要求。常用的解决方法是将算法的中间数据实时保存下来，在完成控制权移交的同时也进行数据继承，这样切换后的控制器在继承的数据的基础上继续运行控制算法，保证在外界看来如同未切换一样。继承数据的方式主要有通过串行数据总线进行实时传输，或者通过双口 RAM 进行中间数据的存储和读取。这其中数据的完整性和时效性是关键，需要重点解决。

1.3 本文的主要研究内容

在深入了解冗余技术尤其是同步技术和无缝切换技术理论的基础上，结合项目的具体要求，完成双 DSP 冗余系统的设计。其硬件设计包括冗余结构和备份方式的选择、DSP 状态检测方式的选择、DSP 和 FPGA 选型、通讯方式的选择和同步信号来源。软件设计包括主从 DSP 各自的通讯程序以及协议设计、DSP 的状态脉冲发生程序以及 FPGA 的状态检测逻辑、切换逻辑和复位逻辑。

要实现以上功能，首先要从多种多样的冗余技术、备份技术和通讯技术中选择合适的方法，既要保证系统的可靠性，又要兼顾系统的成本和运行效率，完成系统的整体设计；其次，完成芯片选型和各个功能模块的硬件设计；然后，需要在 CCS3.3 的环境下开发 DSP 程序，在 ISE8.2 和 Modelsim 联合环境下开发 FPGA 程序，逐个将功能模块进行实现；接着，进行软硬件联调，保证软硬件协调工作；最后，进行故障注入，设计测试方案，测试系统的可靠性。

1.4 本文组织结构安排

首先，从理论上对三种常见的硬件冗余结构进行对比分析，从中选择出兼顾性能与成本的最佳结构；对三种常见的备份方式进行对比分析，从中选择出满足本系统性能要求的最佳方式；考虑 DSP 的特性，选择合适的状态检测方式；综合前 3 者，完成 DSP 和 FPGA 的硬件选型；综合考虑硬件资源、传输速率和数据量，选择一种合适的通讯方式。至此，完成系统的硬件设计。接着，在已完成的硬件实物上进行各功能模块的实现。包括在 FPGA 上构建双口 RAM；利用 AT89C52 产生同步信号；在 DSP 上产生状态脉冲；在 FPGA 上进行状态脉冲检测；传输与存储协议的设计用于数据完整性的判断；注入故障的设计。

最后，分三步，将软件逻辑与各功能模块结合起来，设计测试方案，在实物上进行测试与验证，完成最终功能的实现。

第 2 章 总体方案设计

2.1 冗余结构的选择

系统可靠度是指，系统在特定的工作环境下和特定的时间内，完成预期的功能且不出故障的概率。对于大多数微控制器来说，系统可靠度会随着时间以指数形式衰减，通常用 $R(t) = e^{-\lambda t}$ 进行表征，其中 $R(t)$ 为系统可靠度， λ 为系统失效率， t 为时间。

为了提高系统的可靠度，会在系统设计时引入冗余容错技术。这样，当某一个或某几个单元发生故障时，会利用冗余单元进行替代，从而保证系统仍旧能完成预期的功能，提高系统的可靠度。

常见的系统冗余结构有三种，分别为并联冗余方式、表决冗余方式和旁联冗余方式。这三种冗余的双机结构的示意如图 2-1。

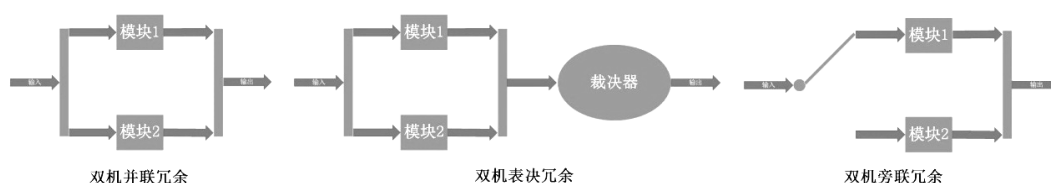


图 2-1 三种冗余结构

并联冗余的概率模型见公式（2-1）：

$$R_s = 1 - \prod_{i=1}^N (1 - R_i) \quad (2-1)$$

其中， R_s ——采用并联冗余方式的系统的可靠度；

R_i ——单个并联冗余单元的可靠度；

N ——并联系统中所有并联单元的数目。

假设 $R_i = 0.5$ ，则对于双机系统， $R_s = 0.75$ ，相对于单机系统可靠性提高 50%；对于三机系统， $R_s = 0.875$ ，相对于双机系统可靠性提升 16.67%。

表决冗余的概率模型见公式（2-2）：

$$R_s = R_m \sum_{i=1}^N C_N^i R(t)^i (1 - R(t))^{N-i} \quad (2-2)$$

其中， $R(t)$ ——单个表决冗余单元的可靠度；

R_s ——采用表决冗余方式的系统的可靠度；

R_m ——表决器的可靠度；

N ——表决系统中所有并联单元的数目。

假设 $R(t)=0.5$ ， $R_m=0.98$ ，则对于双机系统， $R_s=0.735$ ，相对于单机系统可靠性提高 50%；对于三机系统， $R_s=0.8575$ ，相对于双机系统可靠性提升 16.67%。

旁联冗余的概率模型见公式（2-3）：

$$R_s = R_1 + [(1-R_1)R_2 + (1-R_1)(1-R_2)R_3 + \dots + (1-R_1)(1-R_2)\dots R_N]R_w \quad (2-3)$$

其中， $R_2 \dots R_N$ ——离线单元的可靠度；

R_s ——采用旁联冗余方式的系统的可靠度；

R_w ——故障检测切换开关的可靠度；

R_1 ——激活单元的可靠度。

N ——表决系统中所有旁联单元的数目。

假设 $R_1=0.5$ ， $R_w=0.98$ ，则对于双机系统， $R_s=0.745$ ，相对于单机系统可靠性提高 49%；对于三机系统， $R_s=0.8675$ ，相对于双机系统可靠性提升 16.44%^[14]。

由上述计算可知，若假设故障监测及转换装置的可靠度为 0.98，单元可靠度为 0.5 时，三种冗余方式可靠性非常接近，并联最高，旁联次之，表决最差。但考虑到旁联系统可以对离线模块进行修复，从而提升系统寿命，所以选择旁联冗余。另一方面，系统从单机变成双机时，可靠性提高都在 50%左右，而从双机提升至三机，则只有 16.5%左右。所以从成本和提升性能方面考虑，双机系统的性价比要高于三机系统。

综上，在本方案中选择双机旁联方式。

2.2 备份方式的选择

备份方式按照工作方式的不同可分为冷备份、温备份和热备份。对这三种工作方式有多种定义，本文取如下定义：

冷备份是指常态下备份系统不运行，其内部也没有存储数据。一旦发生故障，需启动备份系统，并将应用数据导入后才能替代原系统工作。冷备份的优点是成本低，功耗低，但其缺点也很突出，一旦发生故障，系统从故障中恢复的时间很长，一般要几天甚至 1 周，数据完整性与一致性也没有保证。

温备份是指备份系统与工作系统同时运行，然后定期备份，将工作机的数据

同步到备份系统中。工作机故障后，备份系统直接使用之前定期备份所获得的数据进行控制。它的优点优缺点与冷备份类似，但是恢复时间较短，只需要数小时。

热备份则是指，备份系统也处于工作状态，且通讯模块与工作系统进行实时通讯，完成数据同步。工作机故障后，备份系统可利用实时同步得到的数据立刻替代已故障的系统运行，避免对系统造成大的干扰。它的优点是恢复时间短，从几毫秒到几小时，都能实现。而且因为实时同步，传输的数据也基本不会丢失，所以备份系统中的数据完整性与一致性最好。热备份的缺点在于成本高，且实现难度大。

因为随动伺服系统对系统的实时性要求很高，需要在毫秒级的水平上完成切换，所以必须选用热备份保证系统控制的连续性^[15]。

系统的总体设计框图如图 2-2 所示。

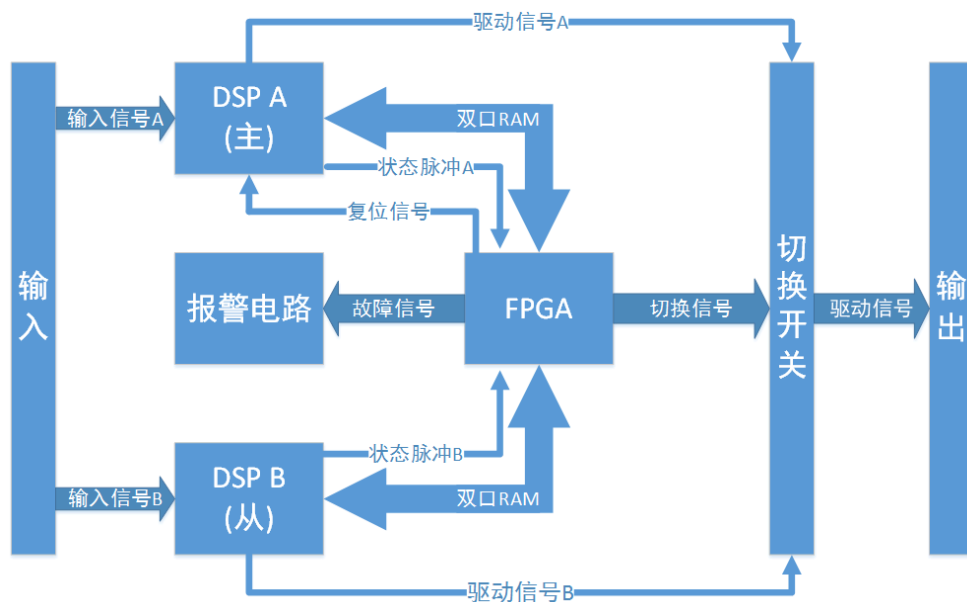


图 2-2 总体设计框图

2.3 系统需求分析

2.3.1 功能需求分析

根据项目的实际需求，结合已有的条件，对设计的双冗余系统提出下面几点功能指标：

(1) 当 FPGA 检测到 DSP A 故障时，如控制器死机、掉电或者程序跑飞等异常情况时，则将系统切换到 DSP B 控制下，DSP B 从 FPGA 中的双口 RAM 中读

取中间数据，从而获取 DSP A 的所有运行参数，然后继续运行算法进行参数计算，得到系统的输出；同时 FPGA 检测到 DSP A 故障后向 DSP A 发送复位信号

DSP A 作为主控制器，DSP B 作为其热备份，FPGA 用于故障检测与修复。系统第一次上电时，DSP A 处于实时工作地位，而 DSP B 则热备份状态。正常工作时，DSP A 通过双口 RAM 将数据实时同步给 FPGA，同时向 FPGA 发送状态脉冲表明自己正在正常工作。当 DSP A 发生故障时，如控制器死机、掉电或者程序跑飞等异常时，状态脉冲消失，由此 FPGA 知道 DSP A 发生故障。然后 FPGA 向 DSP B 发送信号，DSP B 从 FPGA 中的双口 RAM 中读取最近一次保存的完整中间数据，从而获取 DSP A 的所有运行参数，然后继续运行与 DSP A 一致的控制算法，进行参数计算，得到系统的输出，并通过双口 RAM 向 FPGA 实时同步数据；同时 FPGA 对 DSP A 进行硬复位。FPGA 在完成数据传输后对 DSP A 硬复位三次，并在每次复位之后检测状态脉冲。若重新检测到 DSP A 的状态脉冲，则停止发送复位信号，并把系统切换回 DSP A 的控制之下，即 FPGA 通过双口 RAM 将最近一次保存的完整数据传送给 DSP A，并将执行器切换给 DSP A 控制。若三次硬复位之后 DSP A 还是无法工作，则进行报警，不进行切换操作。

(2) 两片 DSP 能接收频率为 1kHz、满足 TTL 电平要求的方波信号，用来进行两片 DSP 的同步，同时触发 DSP 中的控制算法进行运算。

2.3.2 性能需求分析

双 DSP 冗余系统在完成功能指标的基础上，还必须满足项目特定的性能指标，才能投入到实际的工程使用中。其主要的性能指标要求如下：

(1) 双 DSP 冗余系统的软硬件设计，都必须满足一定的降额设计要求，即按照 80% 额度进行降额设计。

(2) 根据研究所的要求，选用的芯片必须是军品级以上，且所使有的芯片都能进行国产化。

(3) 双 DSP 冗余系统在主机切换到从机的过程中，最多能容忍丢失一拍的控制数据，即 1ms 触发的一次外部中断失效；同时必须保证每一次写入的中间结果数据都满足 CRC16 完整性校验，以实现主从控制器之间的无缝切换。

(4) 系统切换时间必须控制在 3ms 以内，否则会导致的控制算法的输出振荡超过百分之一度，调整时间超过 500ms，无法满足项目需求。

2.4 DSP 状态检测方式的选择

因为 DSP 在运行过程会因为掉电、程序跑飞、电磁干扰等而发生故障，所以需要寻找一种办法来检测 DSP 当前的工作状态，保证正在运行的 DSP 是正常的。

常用的状态检测方式有以下三种：

(1) 利用 DSP 自带的看门狗。看门狗 (Watch Dog) 是在大量微控制器中得到运用的一种防程序跑飞和防死机的方法。从本质上说，看门狗就是一个定时器；不同的是，程序必须在规定时间内将此定时器清零，即进行“喂狗” (Feed Dog) 操作，否则看门狗会溢出，然后复位系统。看门狗技术成熟，而且集成在微控制器内部，不需要外部电路，使用也很方便，所以很多对初值不敏感，且对复位启动时间要求不苛刻的系统选择使用此方法。但是在本系统中需要进行微控制器切换，将算法运行的中间变量和结果遗传下去，且这些变量是对初值敏感的，所以利用看门狗进行状态检测的办法并不合适。

(2) 将 DSP 的某个管脚设置为输出，并对其高低电平进行人为定义，如规定该引脚输出高电平代表 DSP 工作正常，输出低电平则代表 DSP 工作异常。这样，在上电后 DSP 对此管脚进行赋值，再由 FPGA 对此管脚的电平进行监测，即可获取 DSP 当前的工作状态。此方法实现起来相比于看门狗更为简单，也不需要外接电路。但是这种方法也有致命缺点，如 DSP 被击穿时，其引脚会输出固定的高电平与低电平，也就是说，FPGA 无法检测出此时 DSP 的故障状态，所以对本系统此方法不合适。

(3) 将 DSP 的某个管脚设置为输出，并定时对其进行翻转操作从而产生方波。FPGA 通过监测此方波的频率来判断 DSP 当前的工作状态。此种方法亦不需要外接电路，软件上可以通过 DSP 内部的定时器实现，实现难度是三种方法中最大的。

综合考虑以上三种实现方法，最终选择第三种方案。同时考虑到本系统中的输入信号会在一个 1kHz 方波触发下的外部中断中处理，所以可以利用此外部中断进行输出管脚的翻转，从而大大降低了实现难度，也减少了软件开销。

2.5 硬件选型

对于 DSP，为了满足国产化的需求，同时考虑系统中控制芯片的延续性和兼容性，以及基于维护方面的考虑，选用了 TI 公司的 TMS320F2812。

其特性包括：

1. 采用了高性能静态 CMOS 技术，最高时钟主频能达到 150MHz，且进行了低功耗设计；

- 2.支持 JTAG 边界扫描，便于仿真和单步跟踪调试，还支持分析和断电功能，提高了程序开发的效率；
- 3.高性能 32 位 CPU，能对中断进行快速响应和处理；
- 4.丰富的外部接口，三个外部中断，并带有外部接口（XINTF），方便拓展存储器空间。

对于 FPGA，在均衡成本和性能后，选择 Xilinx 公司的 XCV300-4PQ240I。

其特性包括：

- 1.采用了 0.22 μm 集成电路制造工艺以及 CMOS 和 SRAM 技术；
- 2.在利用 ISE 环境进行开发时有集成的 IP 核可供调用；
- 3.提供了 322970 门，166 个输入输出引脚；
- 4.自动平衡速度和密度的平衡构架；

2.6 通讯方式的选择

根据 TMS320F2812 芯片提供的硬件资源及系统的要求，通讯方式主要有以下两种类型：串行通信和并行通信。

串行通讯主要是利用 TMS320F2812 芯片自带的多通道缓冲串口(McBSP)，作为一个标准外设，它可以与符合工业标准的编码器和解码器以及符合相应串行通讯协议的模数转换器、数模转换器等传感器进行通讯。同时也可以和其它微处理器进行通讯，便于数据交换和控制命令的首发，但因为其串行的特性，只能在数据量不是很大的情况下使用。因为串口是静态的，所以在任意低的频率下按标准工作。使用内部串口时钟时，TMS320F2812 的标准串口的最高工作频率可达到 CLKOUT 的 1/4，即在 5ns 时其传输速率可达到 50Mbit/s。TMS320F2812 的多通道缓冲串口功能强大，能进行全双工通信。同时它还自带双缓冲数据寄存器，允许数据流进行不间断传输。数据传输可以进行配置选择是使用外部时钟还是内部的可编程时钟。

并行通信因为其能在同一时间传输多位数据，所以对于大数据量、高速率的应用大多采用并行通讯。实现并行通讯的方式主要有两种，其主要区分点在于进行数据传输的双方是否可以同时访问共享存储器。其中，DMA 是不能同时访问共享存储器的代表，它是微控制器常见的功能，因为能把数据传输和控制逻辑独立开而得到广泛应用。另一种典型的并行通讯方式是双口 RAM，但是微控制器很少内部集成双口 RAM，需要外接其它器件，同时它不支持同时访问同一单元。因为 TMS320F2812 没有 DMA 功能，所以此处只讨论双口 RAM。

双口 RAM 是一种供控制器外扩的存储器件，它有两个端口，所以支持两个处理器分别从两侧进行访问，且两个端口各有一套独立的地址、数据、控制总线，每一个处理器都可以将双口 RAM 当作自己的本地存储器，支持它们同时访问不同地址单元的任何地址单元。

综合考虑硬件资源、传输速率和数据量，此处选择利用 FPGA 搭建一个双口 RAM。

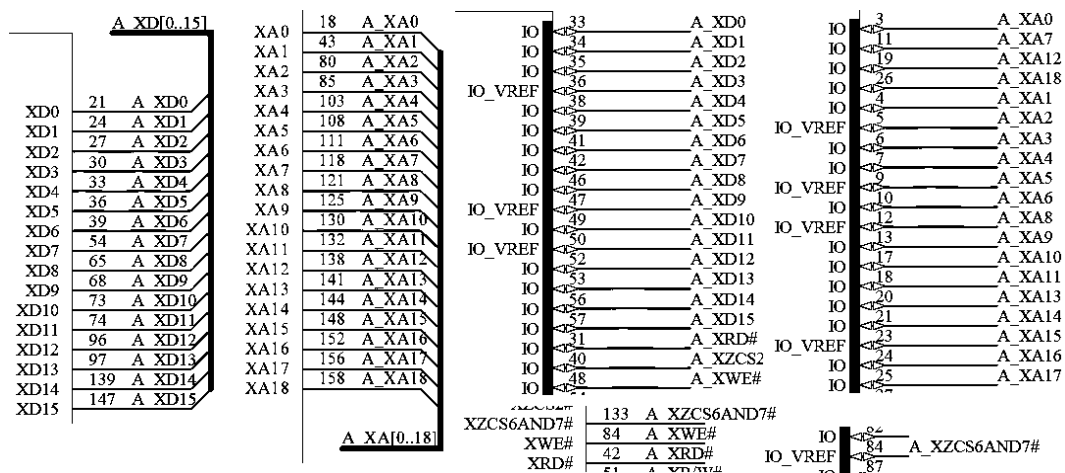
2.7 本章小结

本章从理论上对三种常见的硬件冗余结构进行对比分析，从中选择出兼顾性能与成本的双机旁联模式；对三种常见的备份方式进行对比分析，从中选择出满足本系统性能要求的热备份；考虑 DSP 的特性，选择合适的状态检测方式，即翻转输出管脚产生特定频率的方波；根据实际需要，完成 DSP 和 FPGA 的硬件选型，DSP 选择 TMS320F2812, FPGA 选择 XCV300-4PQ240I；综合考虑硬件资源、传输速率和数据量，选择一种合适的通讯方式，即基于 FPGA 进行并行通讯的双口 RAM。至此，完成系统的硬件设计。

第 3 章 功能模块实现

3.1 利用 FPGA 实现双口 RAM

双口 RAM 是一种供控制器外扩的存储器件，它有两个端口，所以支持两个处理器分别从两侧进行访问，且两个端口各有一套独立的地址、数据、控制总线，每一个处理器都可以将双口 RAM 当作自己的本地存储器，支持它们同时访问不同地址单元的任何地址单元。DSP 与 FPGA 的电路连接，以 DSP A 为例，如图 3-1 所示。



下的 XRD#、XWE#、XZCS6AND7#自动根据程序中的读写操作完成电平变化，所以只需要在初始化阶段配置好其建立时间、有效时间和保护时间即可。

同时两者之间还有一组 16 位的数据总线 XD[0..15]和一组 19 位的地址总线 XA[0..18]，故 DSP 可读写区域达 512K×16 位，完全满足中间变量的存储要求。

因为存储空间足够，故将其等分成两个区域，即区域一：0x100000 到 0x13FFFF，区域二：0x140000 到 0x17FFFF。在写入时，奇数次写入区域一，偶数次写入区域二，这样保证了数据的完整性，即使写入到一半时产生突发情况也可以保证中间数据完整。

利用 FPGA 的 IP 核生成的双口 RAM 如图 3-2 所示。

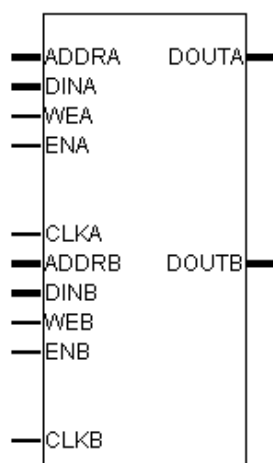


图 3-2 双口 RAM 的 IP 核示意图

从图 3-2 中可以发现，每个端口的数据输入和输出是分开的，数据通过 DINX 总线输入，通过 DOUTX 总线输出。而如图 2-1 所示，DSP 与 FPGA 连接时数据总线是双向的，既是输入又是输出，并没有分成输入总线和输出总线。所以在实例化双口 RAM 的 IP 核时，需要对 IP 核的数据总线作特殊处理，将 inout 类型的数据总线从一个三态的总线拆分成两个独立 wire 型的总线，然后输入到 IP 核中去。

FPGA 程序的仿真波形如图 3-3 所示。

从图 3-3 可以看出，仿真设计的 testbench 先拉低 A 侧的片选线和写入线，从 A 侧向双口 RAM 的地址 0x100001 写入 0x2，然后拉高 A 侧的片选线与写入线，完成写操作。等待一段时间后，再拉低 B 侧的片选线与读取线，在 B 侧从双口 RAM 的地址 0x100001 读出数据，发现 B 侧数据总线输出 0x2，与之前 A 侧写入的一致，说明 A 写入 B 读取的逻辑正确。然后与之类似，从 B 侧向双口 RAM 的地址 0x100002 写入 0x3，然后在 A 侧从此地址读出数据，发现 A 侧数据总线输出 0x3，与之前 B 侧写入的一致，说明 B 写入 A 读取无误。

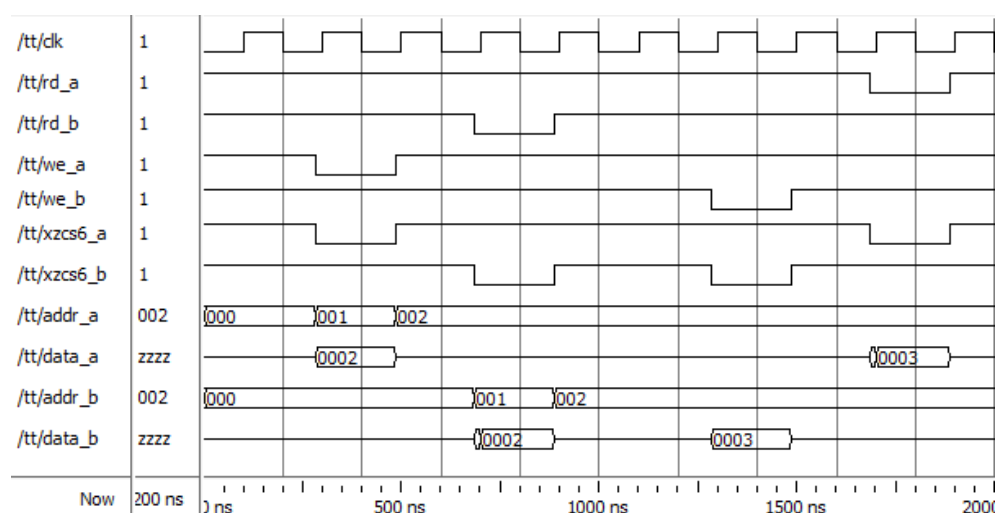


图 3-3 双口 RAM 的仿真波形

需要说明的是，此处的地址总线是 19 位的，即对于地址 0x100000 到 0x17FFFF，地址线只能区分后 19 位，但是地址有变化的只有后 19 位[18..0]，前 5 位[23..19]固定为 00010，所以在 testbench 中地址线为 0x02 时即代表对地址为 0x100002 的空间进行操作。

3.2 同步信号发生

为了两片 DSP 能够进行同步，以合理的时序和谐的对数据进行读取和处理，所以需要产生同步信号提供给两片 DSP。

在本设计中系统的输入是 1ms 进行一次的 A/D 信号输入，所以以此为基准，产生 1kHz 的方波用于触发 DSP 的外部中断。此模块用于调试时使用，没有特殊要求，所以利用成本较低、易获得、程序实现较容易的 AT89C52 实现。利用 AT89C52 的定时器对输出管脚 P1.0 进行翻转操作以获得 1kHz 的方波。

同步信号发生的电路如图 3-4 所示（仅给出控制器部分，电源和程序下载部分省略）。

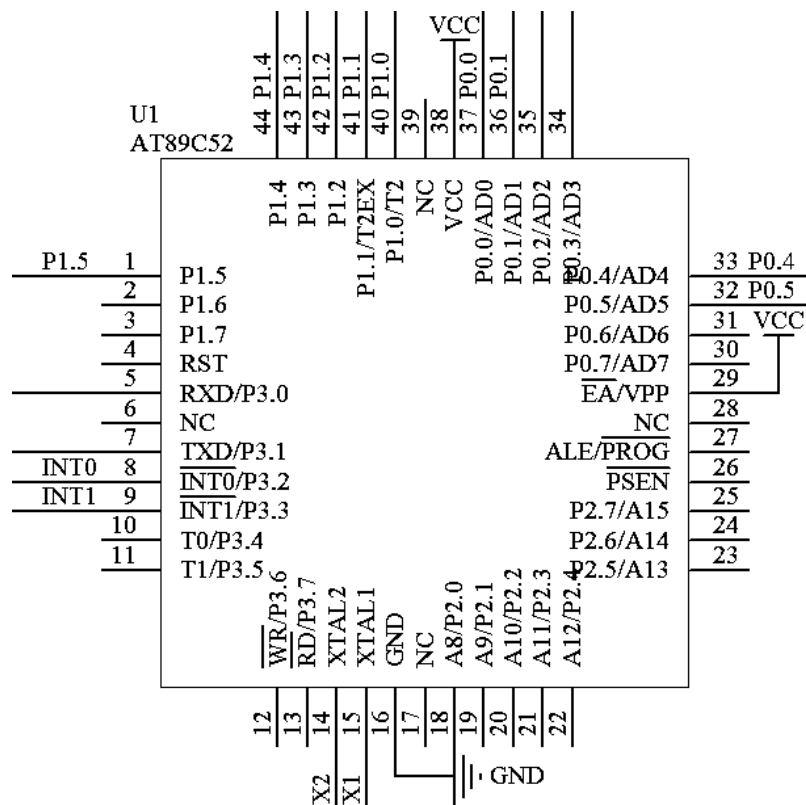


图 3-4 同步信号发生电路

从图 3-4 可以看出，仅仅提供了 AT89C52 的最简工作环境，其产生的同步信号由管脚 P1.0 输出。因为 51 系列的单片机在 P1 端口的 8 个管脚都在内部自带上拉电阻，所以外部不用再接上拉电阻，简化了电路。

将编写的程序烧写到 AT89C52 中，然后用示波器观察相应的管脚产生的同步信号如图 3-5 所示。

从图 3-5 可以看出，输出为频率为 1.00001kHz 的方波，与目标 1kHz 差距不大，且满足 TTL 电平的要求，可认为符合要求。

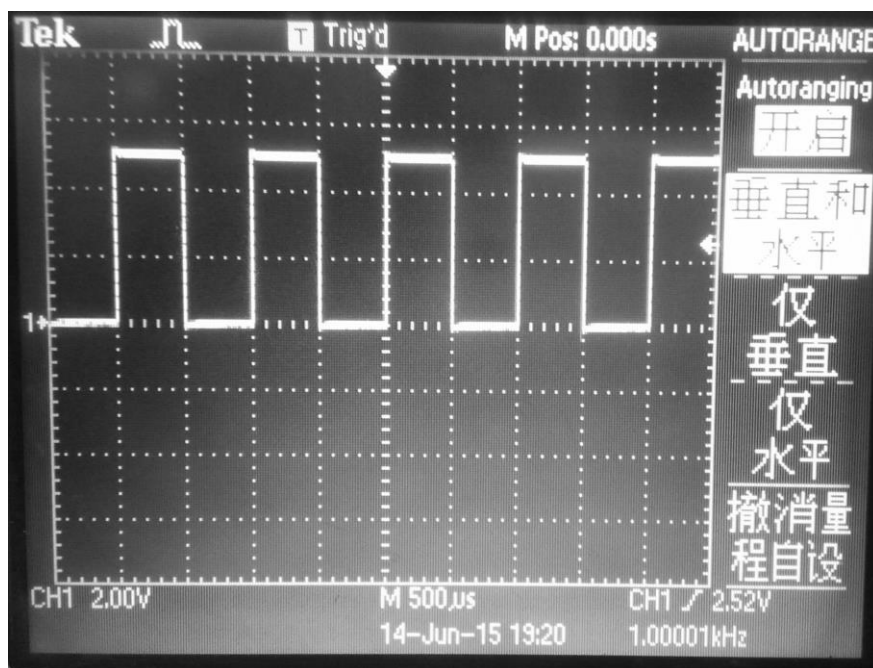


图 3-5 AT59C52产生的同步信号

3.3 DSP 的状态脉冲发生

此部分电路设计如图 3-6 所示。

从图 3-6 中可以看出，DSP A 和 DSP B 的状态脉冲都是通过同一个 I/O 管脚输出，即 62 管脚（C5TRIP#/GPIOB14），所以两者产生状态脉冲的程序相同。

A M4 DIS	61	TCLNIND
A GPIO CHECK	62	C4TRIP#
A TX422 EN63		C5TRIP#
		C6TRIP#
B M4 DIS	61	TCLNIND
B GPIO CHECK	62	C4TRIP#
B TX422 EN63		C5TRIP#
		C6TRIP#

图 3-6 DSP的状态脉冲发生电路

因为需要产生的状态脉冲是方波，所以将 62 管脚设置为通用输入输出 GPIO 的输出模式即可，即 GPIOB14 为通用输出口；因为方波频率为 500Hz，所以需要精确延时，每 1ms 故进行一次翻转操作。同时考虑到本系统中的输入信号会在一个 1kHz 即周期 1ms 的方波触发下的外部中断中处理，所以可以利用此外部中断进行输出管脚的翻转，既降低了实现难度，也减少了软件开销。

利用外部中断而不是 for 循环延时产生，既保证了延时的精确性，也保证了翻转的实时性，同时也确保了两个 DSP 的状态脉冲的同步。

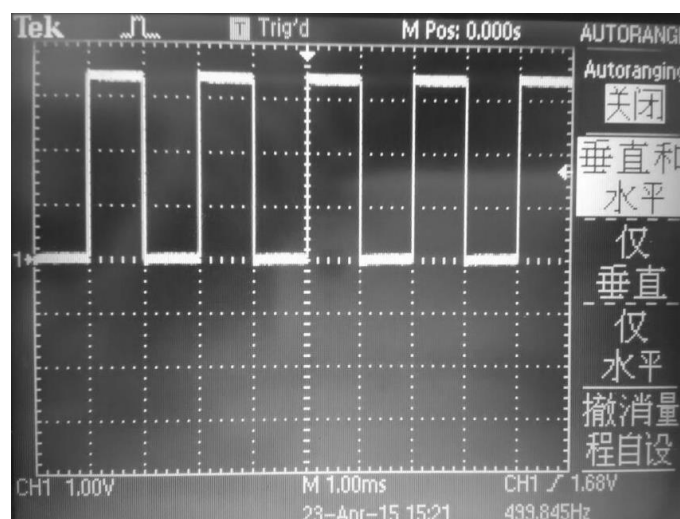


图 3-7 DSP发生的状态脉冲

将编写的程序烧写到 DSP 中，然后用示波器观察相应的管脚，可以看到生成的波形如图 3-7 所示。

此方波的频率为 499.845Hz，与目标 500Hz 差距不大，可认为符合要求。

3.4 FPGA 的状态脉冲检测

此部分电路设计如图 3-8 所示。

在 FPGA 上进行状态脉冲检测程序设计，其状态切换逻辑如图 3-9 所示。

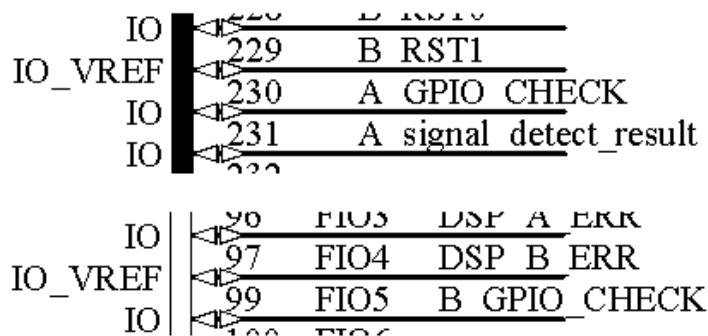


图 3-8 FPGA 的状态脉冲检测电路

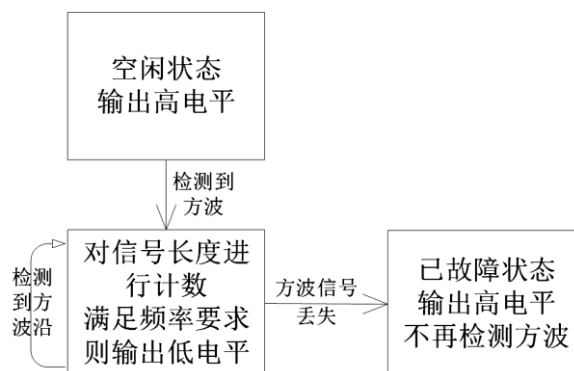


图 3-9 FPGA 状态脉冲检测逻辑

在进行状态脉冲检测的设计时需要重点注意的是，因为 DSP 和 FPGA 的启动不

可能完全同步，所以在刚上电时可能会有延迟，即 DSP 发出的第一个方波沿到来的时间会大于 1ms，故需要在程序设计时注意这点，将这个延迟的影响消除掉。

最终程序的仿真波形如图 3-10 所示。

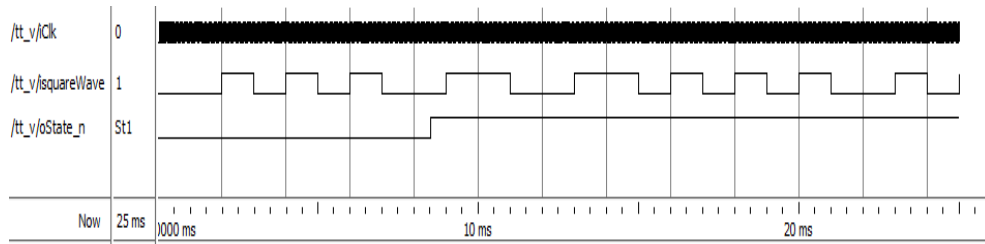


图 3-10 FPGA的状态脉冲检测仿真波形

其中 iClk 为 30MHz 的采样时钟，iSquareWave 用于模拟方波信号，oState_n 为检测结果，低电平代表正常，高电平代表方波异常。

从仿真波形可以看出，启动不同步造成的延迟影响已经被消除，FPGA 并没有误报。而除此以外造成的方波异常都能检测出来，且故障之后即使方波信号恢复为 500Hz 的正常信号也不会改变检测结果。

3.5 传输与存储协议

传输与存储协议如图 3-11 所示：

帧头	用户定义帧头	数据长度	数据内容	校验
32B	16B	16B	不定	16B

图 3-11 传输与存储协议

开始的两个字均为 0x007E。这 32Bits 代表帧头，只有当检测到连续两个

0x007E 的时候才代表此帧数据传输开始。用户定义帧头占 16Bits,即第三个字 DES。第三个字 DES 代表数据来源。当此协议为存储协议时, DES=0 时代表当前存储空间此前尚未有器件写入过; DES=1 时代表当前存储空间此前由 DSP A 写入; DES=2 时代表当前存储空间此前由 DSP B 写入; DES 取其他值时均为非法。当此协议为传输协议时, DES=1 时代表当前存储空间此前由 DSP A 写入; DES=2 时代表当前存储空间此前由 DSP B 写入; DES 取其他值时均为非法。数据长度占 16Bits,即第四个字 LEN。第四个字 LEN 代表此帧中有效数据的数量。从第五个字开始的连续 LEN 个字为有效数据。数据内容之后的最后 16Bits,即最后一个字为校验位,此处采用 CRC-16 校验。

之所以这么设计协议,是基于以下几个原因。

首先,第三个字存在的意义是便于 DSP A 上电后判断此次上电是第一次上电还是修复后复位上电。因为 FPGA 中的程序是从 EEPROM 中导入,所以每次上电之后其内部的双口 RAM 都会遗失掉电之前的数据,即自动清零。DSP 从存储器中依据存储协议读出数据后,则可通过对第三个字的判断,若 DES=0,则代表此次为第一次上电;若 DES=1 或 2,则代表此次为修复后上电。

CRC 校验则是选择的 CRC-16 校验。其生成多项式见公式 (3-1) :

















$$G(x) = x^{16} + x^{15} + x^2 + 1 \quad (3-1)$$

DSP 端对即将发送的有效数据根据协议进行打包,其 CRC 计算是根据 CRC 的原始定义进行亦或和移位操作得出。利用 RAM 版本的 DSP 程序进行跟踪,在加入断点后的跟踪结果如图 3-12 所示。

从图 3-12 可以观察到当 DSP A 对有效数据 0-9 这 10 个数字进行打包发送时,其校验位为 0xDAAD。此结果与手算结果一致。

FPGA 端对接收到的数据根据协议进行解包。在检测到帧头后,将帧头后的数据按格式进行解析,然后存放到相应的寄存器或寄存器组中。最后再进行 CRC 计算,将计算结果与收到的校验位进行比较,若结果一致则代表此帧信息有效,且

发送此帧信息的 DSP 并没有损坏。

-  buf	0x003F9100	unsigned int[100]	hex
 [0]	0x007E	unsigned int	hex
 [1]	0x007E	unsigned int	hex
 [2]	0x0001	unsigned int	hex
 [3]	0x000A	unsigned int	hex
 [4]	0x0000	unsigned int	hex
 [5]	0x0001	unsigned int	hex
 [6]	0x0002	unsigned int	hex
 [7]	0x0003	unsigned int	hex
 [8]	0x0004	unsigned int	hex
 [9]	0x0005	unsigned int	hex
 [10]	0x0006	unsigned int	hex
 [11]	0x0007	unsigned int	hex
 [12]	0x0008	unsigned int	hex
 [13]	0x0009	unsigned int	hex
 [14]	0xDAAD	unsigned int	hex



 Watch Locals	 Watch 1
--	---

图 3-12 DSP的CRC计算结果

其仿真波形如图 3-13 所示。

相同的数据流，FPGA 计算出来的 CRC 校验位也是 0xDAAD。

故 DSP 与 FPGA 的 CRC 校验程序相互匹配。

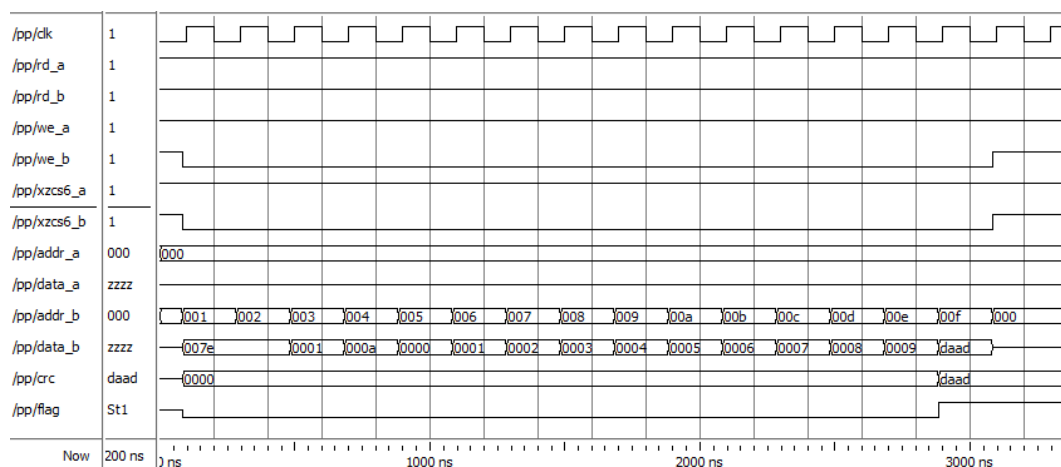


图 3-13 FPGA的CRC校验仿真波形

3.6 硬件电路的改装与设计

硬件电路的改装与设计如图 3-14 示。

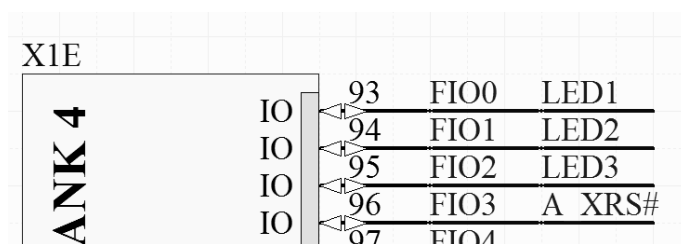


图 3-14 硬件电路的改装与设计

电路的改装主要利用了 FPGA 多余的端口。其中，A_XRS#为 DSP A 的复位信号；LED1、LED2 和 LED3 用于调试过程中对 FPGA 一些中间状态的显示。

3.7 注入故障的设计

在完成系统的设计后，需要通过故障注入技术，对系统的可靠性进行验证。

因为 FPGA 对 DSP 正常/故障的判断主要依靠对 DSP A 的状态脉冲检测结果，而 DSP A 的状态脉冲来自于其外部中断 ISR 中的翻转输出引脚操作。有多种情况可能导致翻转操作的消失，大致可以分为硬件和软件两种，其中最典型的是硬件上掉电和软件上因程序跑飞等导致无法执行外部中断 ISR。基于以上两种典型情况，设计两种故障进行注入，分别从硬件和软件上模拟 DSP A 失效。

第一种注入故障是最常见的掉电。第二种注入故障是人为在程序中关闭外部中断 ISR，并在复位后清除关闭外部中断 ISR 的操作。

第一种故障对应的预期现象为：FPGA 检测到状态脉冲消失而进行切换和复位操作，但是因为 DSP A 掉电而复位失败，即 DSP A 工作一段时间后失效，然后 DSP B 进行接管并一直工作。

第二种故障对应的预期现象为：FPGA 检测到状态脉冲消失而进行切换和复位操作，DSP A 复位成功，即 DSP A 工作一段时间后失效，然后 DSP B 进行接管，接着 DSP A 复位成功后再次接管并一直工作。

3.8 本章小结

本章主要介绍了在已完成的硬件实物上进行各功能模块的实现。包括在 FPGA 上构建双口 RAM；DSP 状态脉冲发生；FPGA 对状态脉冲的检测；传输与存储协议的设计；注入故障的设计。

第 4 章 系统调试与测试

4.1 系统调试流程

系统调试分成三步。

第一步：调试系统的正常逻辑。具体来说，就是完成系统在正常、未切换之前 DSP 和 FPGA 的功能。具体来说就是，FPGA 中构造的双口 RAM 能够正常工作；DSP 能对 FPGA 中的双口 RAM 进行正常读写；DSP 能利用外部中断 1 进行数据处理与状态脉冲的输出；FPGA 能检测到 DSP A 的状态脉冲符合要求。

第二步：调试系统的切换逻辑。具体来说，就是在第一步的基础上，人为的使 DSP A 进入死循环，然后 FPGA 能检测到 DSP A 的状态脉冲异常，并利用外部中断向 DSP B 发送信号；DSP B 能成功接收信号并从双口 RAM 中取数，并在此基础上继续运行算法，同步数据。

第三步：调试系统的复位逻辑。具体来说，就是在第二步的基础上，加入 FPGA 向 DSP A 发送的复位信号，并根据复位的成败进行控制；若失败，则不操作；若成功，则向 DSP B 发送信号，停止 DSP B 中的算法，改由 DSP A 接管。

4.2 系统功能测试

根据调试流程，功能测试也分三步。因为本文不涉及具体的 DSP 内部的处理算法，仅仅是搭建一个具有可移植性的基于 FPGA 的双 DSP 冗余的原型，所以在测试过程中 DSP 内部搭载的算法被简化成每 1ms 自动加 1，这样在验证时可以通过两次数据之间是否连续来进行判断。

4.2.1 正常逻辑测试

该步骤主要为了验证两片 DSP 和 FPGA 上构建的双口 RAM 之间的通讯。其实际电路连接如图 4-1 所示。

各组件功能也在图 4-1 中进行了标明。

软件上，FPGA 上仅搭载双口 RAM；DSP A 上电后先灭灯，然后通过同步信号触发的外部中断向双口 RAM 中按要求写入数据；DSP B 同样由同步信号触发外部中断，但是在 ISR 中每次都会先延时 0.5ms，保证 DSP A 写入完成后再进行读取，然后与前一次数据进行比较，若连续则点灯，否则灭灯。

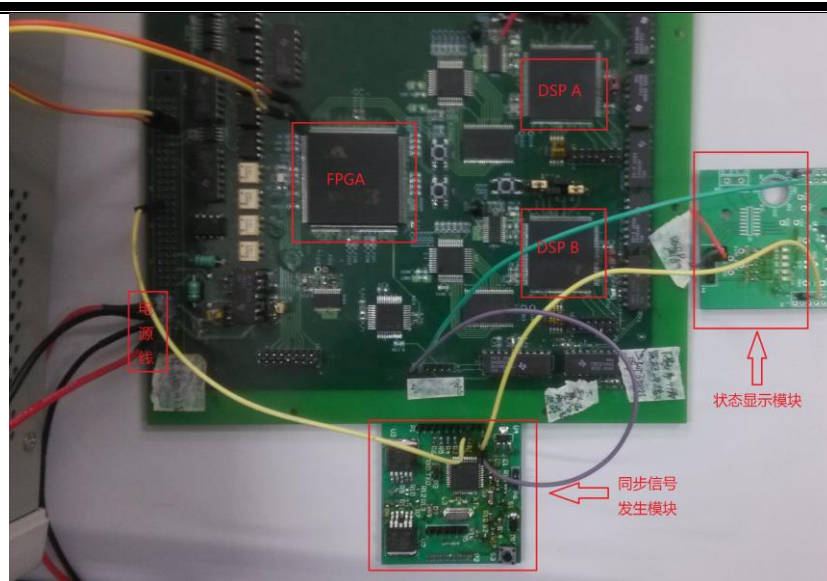


图 4-1 正常逻辑测试的硬件连接

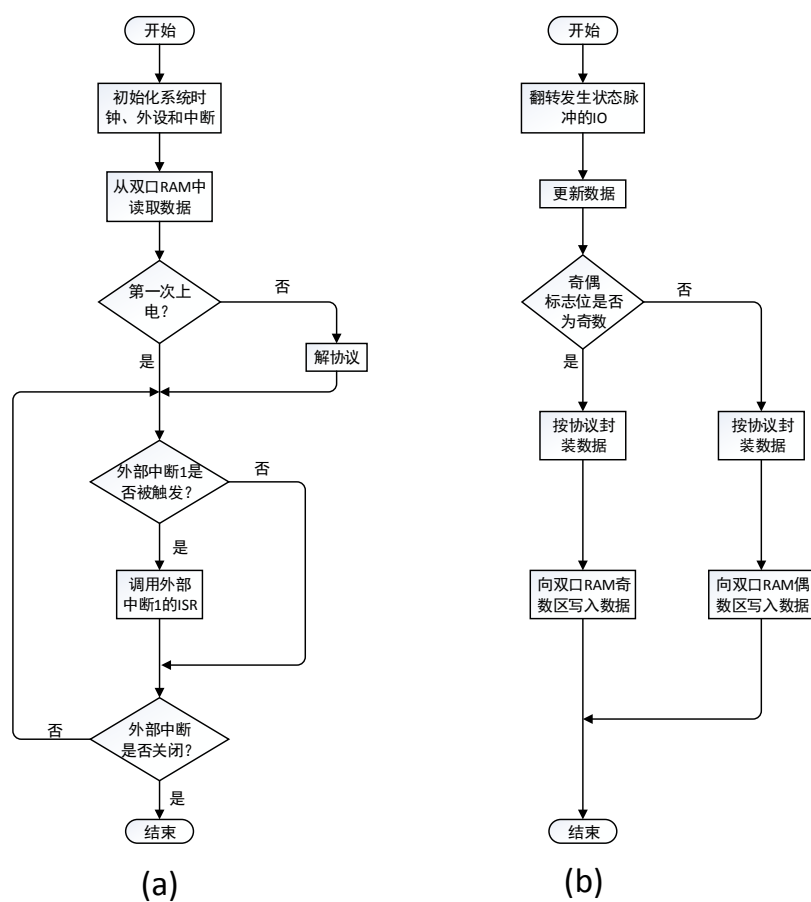


图 4-2 a) DSP A主程序流程图 b) DSP A外部中断1的ISR程序流程图

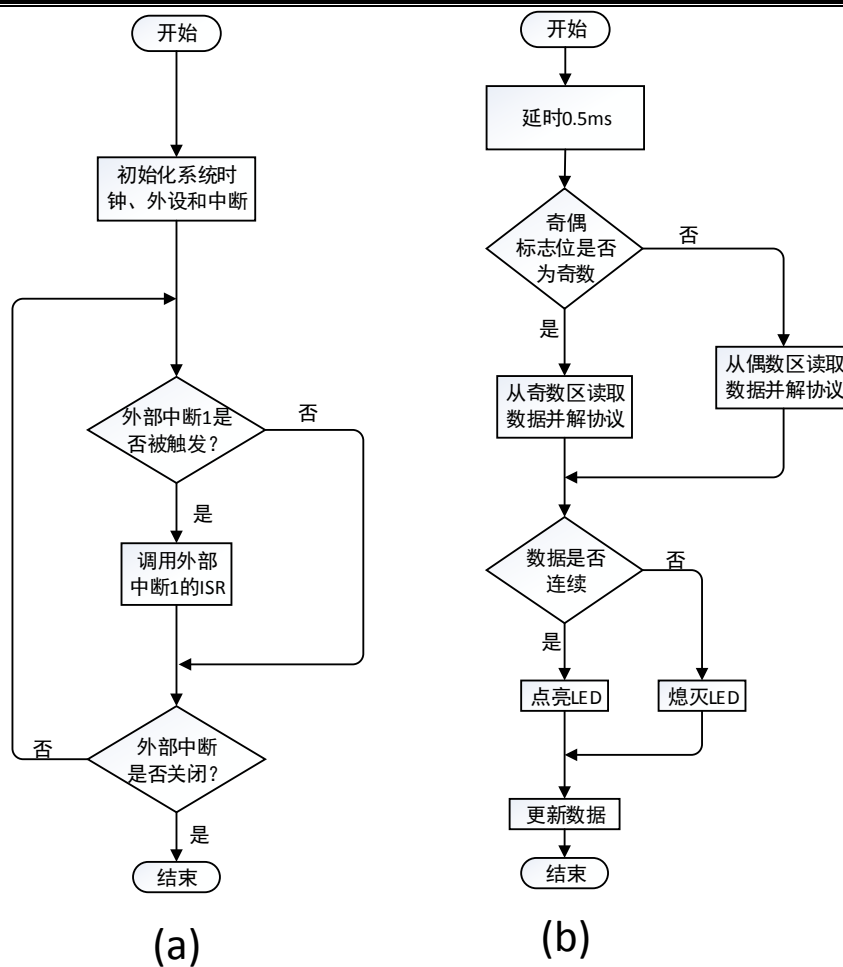


图 4-3 a) DSP B主程序流程图 b) DSP B外部中断1的ISR程序流程图

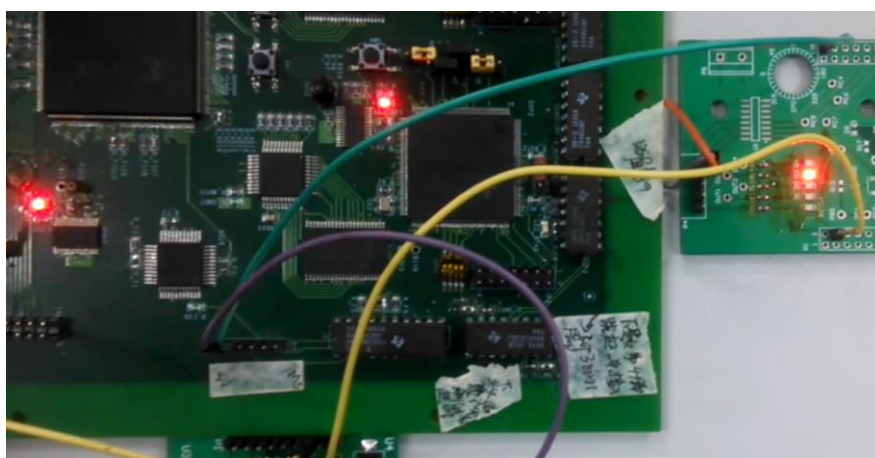


图 4-4 正常逻辑测试现象

DSP A 中的程序流程图如图 4-2 所示，DSP B 中的程序流程图如图 4-3 所示。上电之后的现象如图 4-4 所示。

从图 4-4 中可以看到，显示模块上侧的第一个 LED 灯不亮，代表 DSP A 正常工作，正在向 FPGA 中的双口 RAM 写入数据。上侧第二个 LED 灯亮着，代表 DSP B 双口 RAM 里读取到了数据，且读到的数据是连续的。

4.2.2 切换逻辑测试

该步骤主要为了验证 DSP B 能完整、及时的从双口 RAM 中继承数据并继续算法，保证切换操作是无缝的。其实际电路连接如图 4-5 所示。

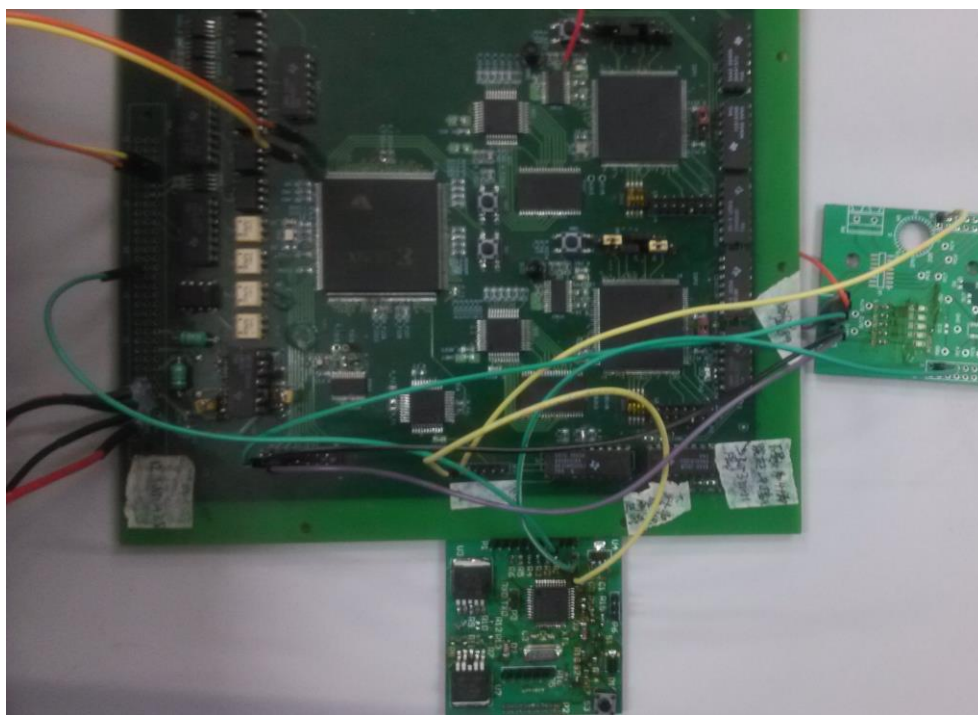


图 4-5 切换逻辑测试的硬件连接

与图 4-1 相比，图 4-5 主要多出了 FPGA 拓展的三根状态显示线，因为在此步骤中，对数据连续性的判断由 FPGA 进行。这三根状态显示线也是连接 LED 灯，分别代表 FPGA 接收到的数据连续就亮，否则灭；接收到的数据为 9 时点亮；接收到的数据为 10 时点亮。其排序如图 4-5 所示为上侧第 3-5 个 LED 灯。上侧的两个分别由两片 DSP 控制，上侧的第一个灯用于确认 DSP 的 FLASH 中的程序能够正常工作；第二个灯用于确认 DSP B 成功的从双口 RAM 中继承了完整的中间数

据。

软件上，DSP A 依旧负责向双口 RAM 中写入数据，但是在写入 0-9 之后就停止。DSP B 则在上电后等待，直到收到 FPGA 的信号后从双口 RAM 中读取数据，验证数据完整性之后继续运行算法，并将结果写回到 FPGA 中。FPGA 除了搭载双口 RAM 之外，还对数据的连续性进行判断和显示，同时也对数据是否为 9 和 10 这两个关键性数据进行检测，方便观察测试效果。

DSP A 中的程序流程图如图 4-6 所示，DSP B 中的程序流程图如图 4-7 所示。

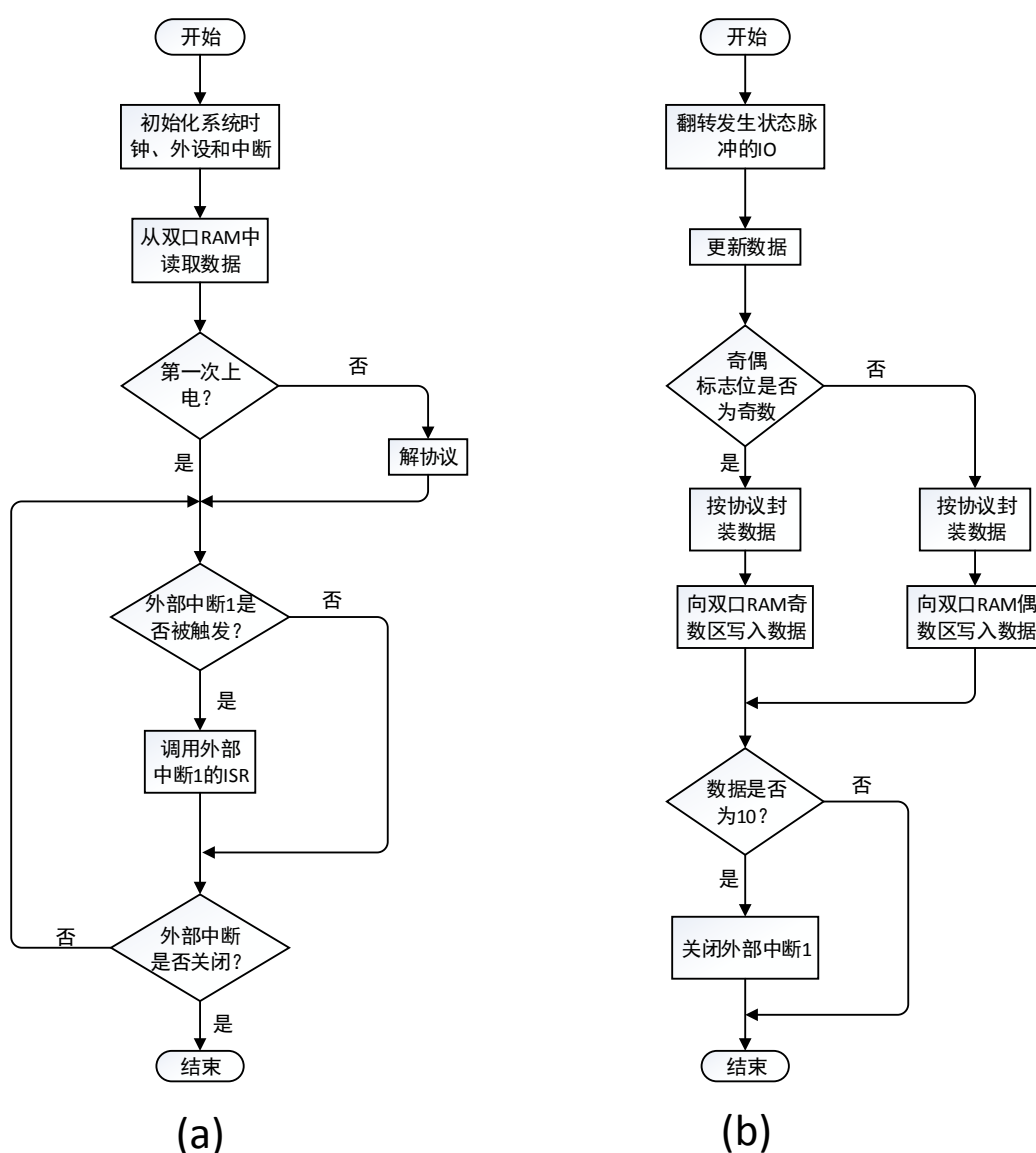


图 4-6 a) DSP A 主程序流程图 b) DSP A 外部中断1的ISR程序流程图

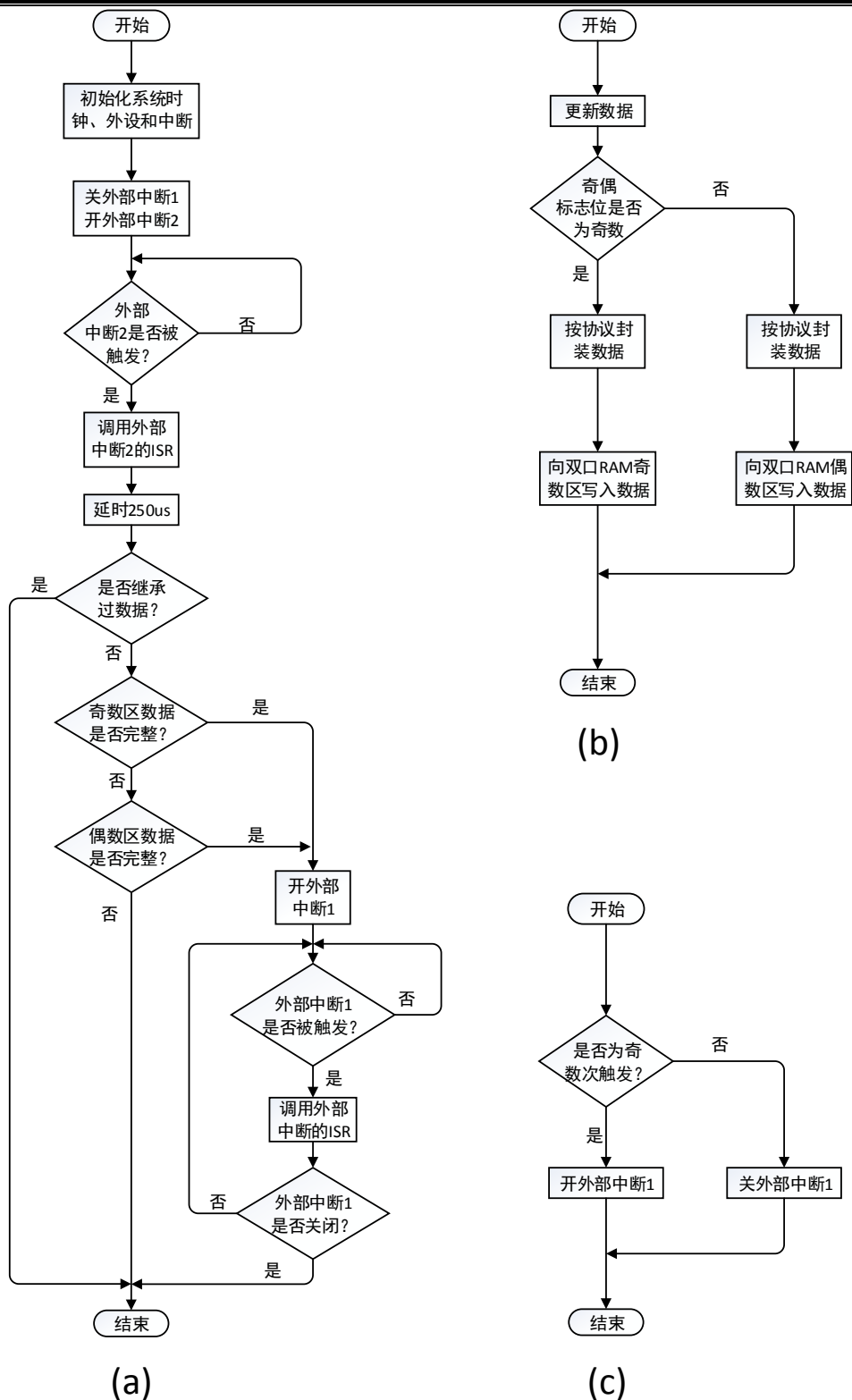


图 4-7 a) DSP B主程序流程图 b) DSP B外部中断1的ISR程序流程图 c) DSP B外部中断2的ISR程序流程图

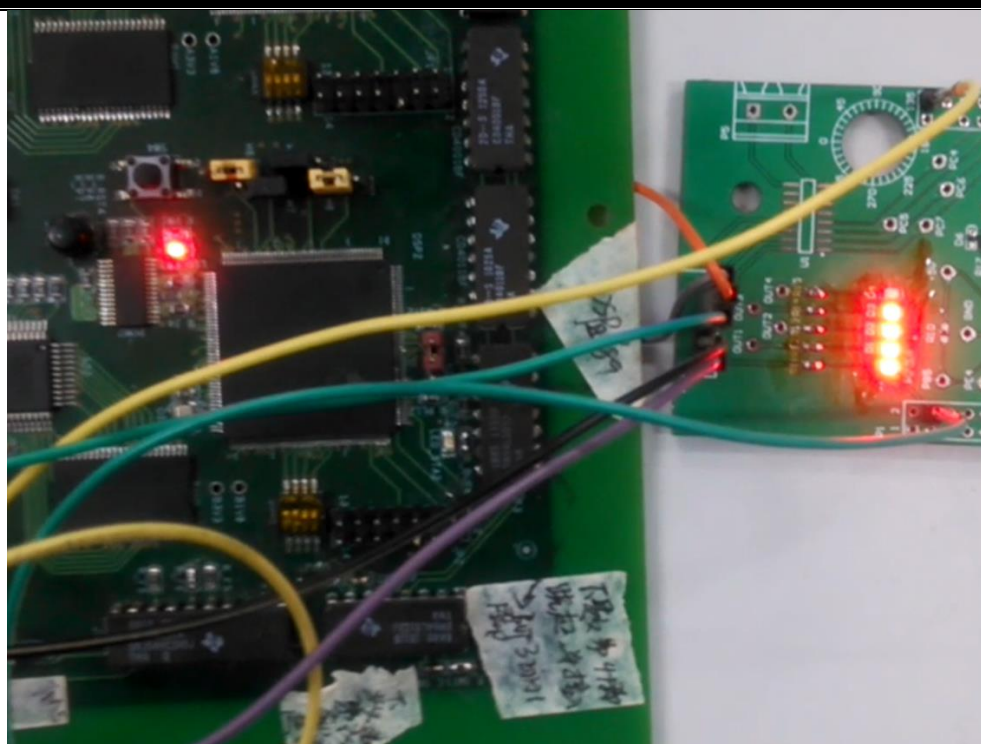


图 4-8 切换逻辑测试现象

上电之后的现象如图 4-8 所示。

从图 4-8 可以看出，上侧第一个 LED 不亮，代表 DSP A 的 FLASH 版本程序正常运行；第二个灯亮，代表 DSP B 从双口 RAM 中继承了完整的中间数据；第三个灯亮，代表向双口 RAM 中写入的数据一直是连续的，即 DSP 中的算法即使经过了切换还是连续的；第四个灯亮代表 FPGA 成功接收到了数据 9，第五个灯亮代表 FPGA 成功接收了数据 10，即切换过程是无缝的。

4.2.3 复位逻辑测试

该步骤在第二步的基础上加入复位逻辑，主要是为了在 DSP A 故障后对其进行复位操作，提高系统的续航能力。其实际电路连接如图 4-9 所示。

图 4-9 中央的红色飞线是外加的 DSP A 的复位信号线，从 FPGA 富余的第四个 IO 引出接到 DSP A 的复位引脚上，用于 FPGA 对 DSP A 的复位操作。右侧状态显示板有两个 LED 灯，分别与一个 DSP 相连。当 DSP 进行写入操作时，代表写入操作的 LED 灯才会亮，否则熄灭。

软件上，与第二步相比，多了复位功能，即 FPGA 在向 DSP B 发送接管指令

的同时，会给 DSP A 发送复位信号和复位自身的状态检测模块，然后在 1.5ms 之后检测 DSP A 的状态脉冲。若仍未收到，则放弃；若收到，则再次进行切换，将控制器切换回 DSP A 的控制之下。

两片 DSP 的程序流程图与第二步相同。

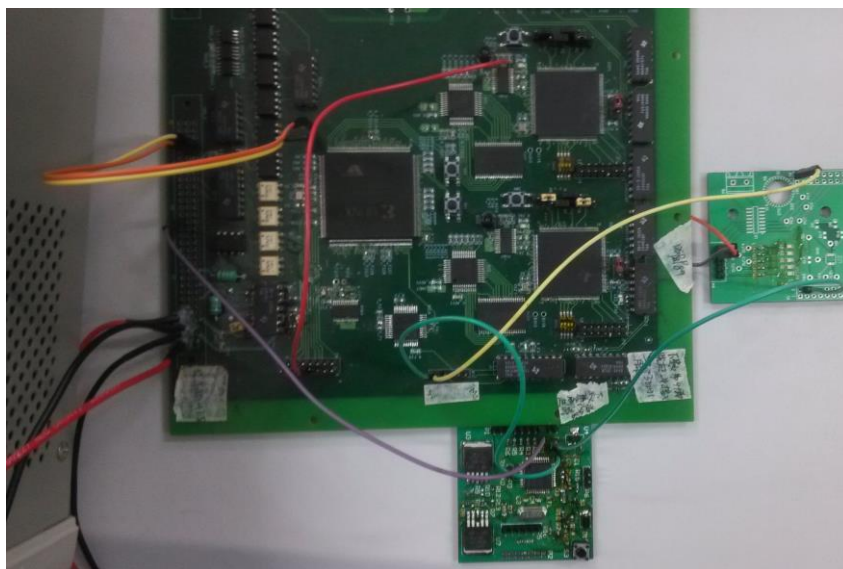


图 4-9 复位逻辑测试的硬件连接

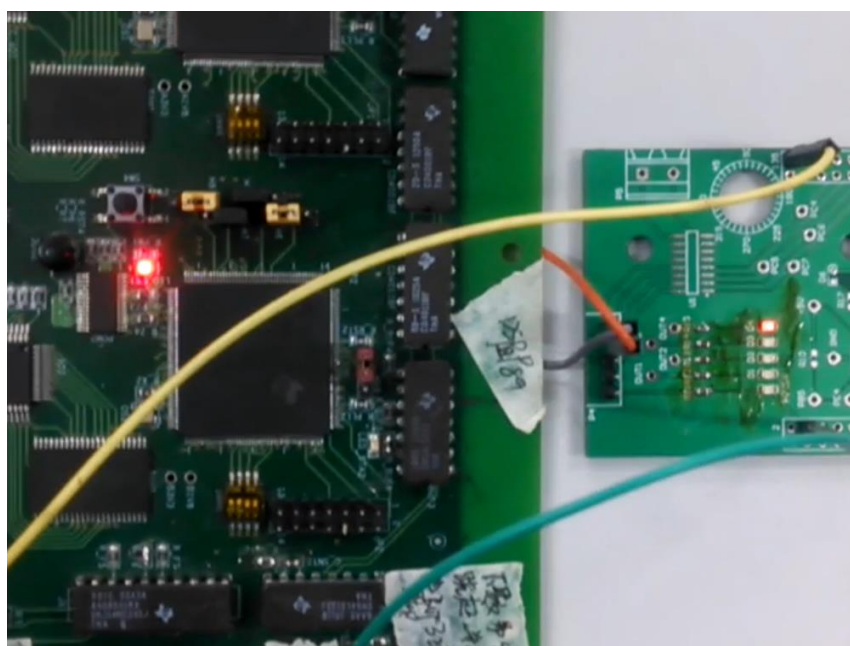


图 4-10 复位逻辑测试的复位前现象

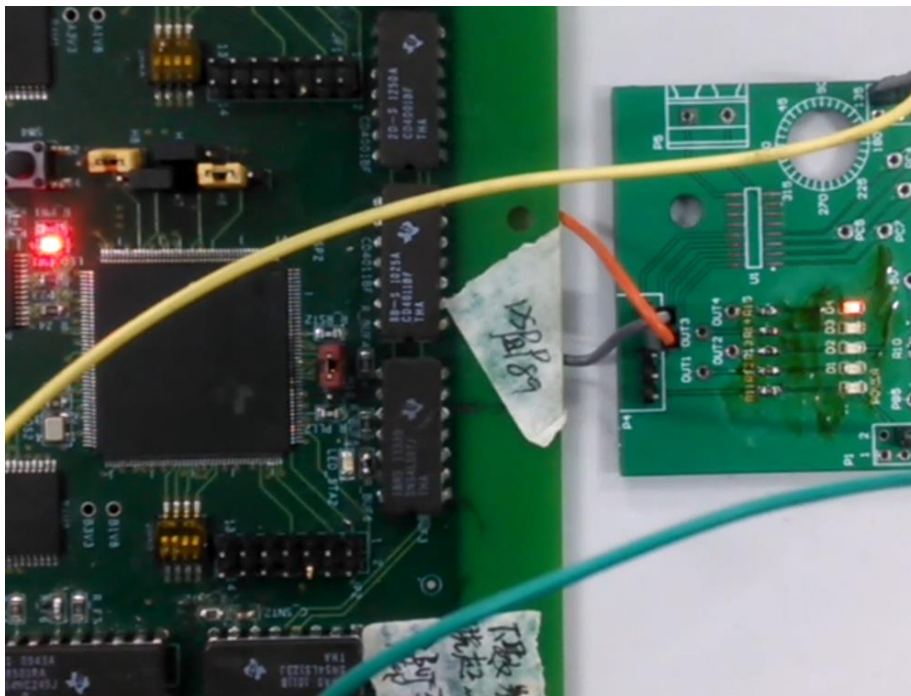


图 4-11 复位逻辑测试的复位后现象

现以故障注入的第二种情况为例，即人为在程序中关闭外部中断 **ISR**，并在复位后清除关闭外部中断 **ISR** 的操作，展示测试结果。复位逻辑测试的复位前现象如图 4-10 所示，复位逻辑测试的复位后现象如图 4-11 所示。

综合图 4-10 和图 4-11 可知，复位前和复位后都是 **DSPA** 进行写入操作，复位成功。因为切换过程太过迅速，所以无法得到理想的图片来展示效果。

4.3 本章小结

本章主要根据设计好的故障注入方式对系统进行调试与测试。测试结果表明，正常情况下，主控制器能把需要备份的中间数据写入到双口 **RAM** 中。当主控制器发生故障时，**FPGA** 能及时的将系统切换到从控制器的控制之下。当主控制器发生的不是不可恢复的灾难性故障时，**FPGA** 能重启原主控制器，原主控制器也能重新接管控制系统；当主控制器发生的是灾难性故障时，从控制器能很好的完成对控制系统的连续控制。

结 论

本文按照设计——实现——调试与测试的步骤完成了基于 FPGA 的双 DSP 冗余设计，为从单 DSP 系统拓展成双 DSP 冗余系统提供了一个原型。本课题主要完成了一下几项工作：

（1）从理论上对常见的三种硬件冗余结构的可靠性以及三种备份方式进行了分析和对比，从中选取了双模冗余热备份的总体结构搭建系统；根据项目需求和实际条件进行硬件选型后确定了 DSP 状态检测方式和通讯方式。

（2）在原有印制电路板的基础上进行重定义和改装，并自行设计同步信号发生模块和显示模块用于调试与测试，自行设计了调试和测试方案。

（3）利用外部的同步信号解决了同步问题；利用 FPGA 设计出严格的时序以实现无缝切换。

由于毕业设计的时间，本人能力和实验环境的限制，本课题中还有许多细节值得商榷和改进，主要包括：

（1）目前 FPGA 对 DSP 工作状态的判断仅限于对 500Hz 状态脉冲的检测，还可以通过 FPGA 上对写入双口 RAM 的数据进行 CRC16 校验，进行完整性测试，以增加一种判断 DSP 工作状态的方式。在进行模块实现的时候也使用 Verilog 语言完成了 FPGA 上 CRC16 校验的代码并测试通过了，也与 DSP 的 CRC16 校验相匹配，但是在整合过程中发现校验过程是寄存器版本的伪双口 RAM，功能上与利用 IP 核实现的真双口 RAM 重合了，性价比上也因为浪费了大量寄存器而比不上 IP 核实现的真双口 RAM。若能将真双口 RAM 和 CRC16 校验结合起来就能兼顾性能与可靠性了。

（2）在系统的复位逻辑设计与实现时，仅进行了一次复位信号的发送，而通讯领域的一般原则是复位三次来更好的提供续航能力。

（3）目前的功能是当 DSP A 故障后切换到 DSP B 并同时复位，但是考虑到 DSP A 故障的原因并不知道，所以考虑在 DSP B 失效之后再进行复位更合理，能避免因为 DSP A 故障原因未排除而造成的复位——失效——切换——复位的死循环。

（4）未对整个系统做电磁兼容性（EMC）测试。

参考文献

- [1]徐福祥. 用地球磁场和重力场成功挽救风云一号 (B) 卫星的控制技术[J]. 宇航学报, 2001, 22(2): 1-11.
- [2] 童杰文. 高可靠皮卫星综合电子系统研究[D]. 浙江大学, 2014.
- [3] Patton R. Robustness issues in fault-tolerant control[C]//Fault Diagnosis and Control System Reconfiguration, IEE Colloquium on. IET, 1993: 1/1-125.
- [4] Pratt B, Caffrey M, Carroll J F, et al. Fine-grain SEU mitigation for FPGAs using partial TMR[J]. Nuclear Science, IEEE Transactions on, 2008, 55(4): 2274-2280.
- [5] Clark L T, Patterson D W, Hindman N D, et al. A dual mode redundant approach for microprocessor soft error hardness[J]. Nuclear Science, IEEE Transactions on, 2011, 58(6): 3018-3025.
- [6] Cordero F, Mendes J, Kuppusamy B, et al. A cost-effective Software Development and Validation environment and approach for LEON based satellite & payload subsystems[C]//Recent Advances in Space Technologies (RAST), 2011 5th International Conference on. IEEE, 2011: 511-516.
- [7] Buja G, Zuccollo A, Pimentel J. Overcoming babbling-idiot failures in the FlexCAN architecture: a simple bus-guardian[C]//Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on. IEEE, 2005, 2: 8 pp.-468.
- [8] 多处理器实时系统的容错研究与实现[D]. 成都:电子科技大学, 2006.
- [9] 曾涛, 伍保峰. 中国海洋一号卫星星务管理系统设计与在轨性能评估[J]. 航天器工程, 2004, 12(3): 71-78.
- [10]吴钊君. 惯性平台稳定回路的双冗余设计与实现[D]. 哈尔滨工业大学, 2014.
- [11] 强宁. 基于双 DSP 的航空发动机电子控制器设计及仿真验证[J]. 现代制造

工程, 2008 (5): 113-117.

[12] 余同正, 徐龙祥. 基于双 DSP 的磁轴承数字控制器容错设计[J]. 电子技术应用, 2005, 31(1): 27-29.

[13] 上官廷杰, 闫鸿慧, 王国锋. 串行通信控制器在双 DSP 系统中的应用[J]. 装甲兵工程学院学报, 2002 (1): 63-67.

[14] 张汭. 双机系统的冗余及仲裁策略研究[D]. 电子科技大学, 2010.

[15] 陆阳, 王强, 张本宏. 计算机系统容错技术研究[J]. 计算机工程, 2010, 36(13): 230-235.

哈尔滨工业大学本科毕业设计（论文）原创性声明

本人郑重声明：在哈尔滨工业大学攻读学士学位期间，所提交的毕业设计（论文）《基于 FPGA 的双 DSP 冗余设计》，是本人在导师指导下独立进行研究工作所取得的成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明，其它未注明部分不包含他人已发表或撰写过的研究成果，不存在购买、由他人代写、剽窃和伪造数据等作假行为。

本人愿为此声明承担法律责任。

作者签名：

日期： 年 月 日

致 谢

时光荏苒，本科四年的时间匆匆而过，觉得还有很多东西没有学到，就已经到了离开这里的时间。回想这四年，大一大二在努力的学习基础知识，还参加了很多社团，认识了很多朋友；大三开始学习专业知识，并开始进入实验室，尝试将所学的知识运用到实际中。犹记得初基础单片机时候的迷茫；记得一个 TCP/IP 协议调了两周；更记得这所学校的老师和同学给予的帮助。

首先要感谢的是我的毕业设计指导老师王强老师。从毕业设计的选题直到最后的完成，王强老师始终给予我指导和激励，是他给了我机会，在大学的最后阶段还能接触到之前一直缺失的 DSP 和 FPGA 编程，完善了我的知识体系；他独特的人格魅力更是对我的生活产生了影响。

其次要感谢实验室的吴钊君师兄。我的毕业设计是在他的硕士毕业设计的基础上进行的，硬件平台是由他设计，而我只是在其上根据需要进行改装。同时他在编程思路和实际操作细节上作为先行者给我提供了很多建议和帮助。

然后要感谢是计算机硬件实验室的陈惠鹏老师和深空探测实验室的姜成平师兄。陈老师是我硬件开发方面的启蒙老师，是他帮我找到了自己的兴趣和特长之所在，并给了独立完成一个硬件开发项目的机会；而在日常的交流中，他对大学教育、创新以及未来的展望也让我耳目一新。姜成平师兄在我初学单片机时，帮我指明了方向，让我少走了很多弯路；他的严格要求让我对单片机的学习更系统全面；他还让我对本专业有了更深刻的了解，对走出学校后的社会生活不再畏惧。

最后要感谢的是同学和父母。他们为我的生活增添了欢声笑语，也一直是我坚强的后盾。