RapidIO交换芯片的事件监听系统

Event Monitoring System for RapidIO Switching Chips

崔　超[1]，张明权[2]，于　洪[2]，肖　峰[1]（1. 天津市滨海新区信息技术创新中心，天津 300450；2. 解放军战略支援部队信息工程大学，河南 郑州 450002）

Cui Chao[1], Zhang Mingquan[2], Yu Hong[2], Xiao Feng[1]（1. Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin 300450, China; 2. Information Engineering University, Zhengzhou 450002, China）

摘　要：

针对RapidIO交换芯片发生的异常，设计了RapidIO交换芯片事件监听系统，以确保主控芯片及时准确检测到问题。该系统结合signal通信机制和中断机制，对交换芯片发往主控设备的port-write类型的异常事件报文进行监听和处理。试验证明，短时间内发生大量事件上报的情况下系统能监测到所有事件，并自动解析、存储、显示事件，提高了RapidIO事件监听系统的实时性和自动化程度。

关键词：

RapidIO协议；交换芯片；事件监听

doi：10.12045/j.issn.1007-3043.2022.07.012

文章编号：1007-3043（2022）07-0061-05

中图分类号：TN915

文献标识码：A

开放科学（资源服务）标识码（OSID）：

Abstract:

Aiming at the abnormality of RapidIO switching chip, the event monitoring system of RapidIO switching chip is designed to ensure that the main control chip can detect the problem timely and accurately. Combined with signal communication mechanism and interrupt mechanism, the system monitors and processes the port write type abnormal event message sent by the switching chip to the main control device. The experiment shows when a large number of incidents are reported in a short time, the system can monitor all events and automatically analyze, storage and display events, which improves the real-time and automation of RapidIO event monitoring system.

Keywords:

RapidIO protocol; Switching chip; Event monitoring

引用格式：崔超，张明权，于洪，等. RapidIO交换芯片的事件监听系统[J]. 邮电设计技术，2022（7）：61-65.

0 引言

随着信息技术的快速发展，基于RapidIO协议的交换芯片越来越多地应用到航空、航天、国防和电信等各个领域。基于这些领域对通信系统的安全性、可靠性和稳定性的高要求，在交换芯片发生异常时，能否准确、及时地被主控系统检测到是至关重要的。交换芯片因异常引发的事件有很多种，而且有时会在短时间内发生大量事件，主控系统需要准确监测到每种异常事件，不能有漏报、错报现象，这样才能依据事件的种类来采取相应的处理措施。因此，需要一种稳定可靠的事件监听系统对基于RapidIO协议的交换芯片的异常事件进行监听。

1 背景

RapidIO规范是一种开放式标准，由RapidIO贸易协会支持开发，并指导RapidIO架构的未来发展方向。国际标准化组织（ISO）和国际电工协会（IEC）已批准RapidIO互连规范为ISO/IEC DIS 18372标准，成为第一个嵌入式互连国际标准[1]。RapidIO技术一般使用面向串行背板、DSP和相关串行控制平面应用的串

基金项目：国家科技重大专项核高基项目（2016ZX01012101）

收稿日期：2022-06-16

行接口进行互连[2],是一种高性能、低引脚数、基于报文可靠交换的互连体系结构,具有高带宽、低时延、高效率、高可靠性等优点。本文提出的事件监听系统主要针对串行RapidIO交换网络。

与以太网一样,RapidIO也是基于包交换的互联技术。RapidIO相比于其他总线的接口具有明显的优势,它支持丰富的维护和错误管理功能,支持最初的系统发现、配置、错误管理和错误恢复机制。维护操作是通过访问器件的维护寄存器(如错误检测和状态寄存器)来读取相关的器件信息,或者通过写寄存器使RapidIO器件关闭或恢复。在错误管理方面,RapidIO协议本身具有强大的容错能力,它提供了许多错误检测机制来保护数据包,进行错误恢复。

RapidIO错误管理可以分为错误检测和错误恢复。有2种向系统软件通报错误的方法:一种是交换芯片向主控系统发出一个中断,另一种是交换芯片检测到错误之后向主控系统发出维护端口写(maint port-write)报文[3]。本系统采用的是第2种方法。当逻辑层、传输层或物理层有错误,且被交换芯片检测出来时,交换芯片将主要错误信息组成port-write报文(在RapidIO协议中port-write是一种用于错误信息上报的事件报文的格式)发送到主控系统。

在交换芯片的设计中,port-write操作是错误管理模块(Error_management)将记录下来的错误上报给维护包处理模块(Maint_transaction)的一种方式。其主要目的是通过维护包处理模块,将错误信息封装成port-write维护包报文,告知远端的主控系统错误发生的类型和位置。它不保证递送,也没有相关的响应。软件通过清除port-write待处理状态位表明它已经发现了该操作。

RapidIO中有些错误不需要软件干预,由硬件完成错误恢复,这些错误是与协议相关的物理层的错误。还有一些错误需要错误管理软件才能成功恢复,包括逻辑层的错误和执行错误。现将多种错误归纳,可以分为可自动恢复的错误和非自动恢复的错误。

非致命传输错误(如被破坏的包或控制符,产生协议错误)都存在于物理层,是可自动恢复的错误。错误将在端口错误控制状态寄存器中适当的位域置位。可恢复错误不产生中断或请求操作,只存在于物理层中。

检测到错误时端口会进入错误停止状态。进入停止状态后端口会开始一个恢复过程以尝试从错误情况中恢复。除可能使用错误计数器来检测错误率外,系统一般不会检测到链路中的错误,也不能从错误中自动恢复。

非自动恢复的错误可产生中断,会通知主控系统,由错误管理软件恢复。对RapidIO中非自动恢复的错误,目前还没有比较成熟的解决方案,本文提出了以下3种设计思路。

a)当网络中的交换芯片发生错误或异常时,只向网络主控设备发送中断信号,再由网络主控设备主动查询交换芯片内部的错误状态寄存器,进而确定发生了哪种错误。

b)当网络中的交换芯片发生错误或异常时,会向网络主控设备发送port-write类型的异常事件报文,该异常事件报文被存储到预先分配好的存储空间。网络主控设备定时轮询上述存储空间,若发现有异常事件报文,则提取并解析该异常事件报文。

c)采用signal通信机制结合中断处理机制的方式,对b)进行了改进。当port-write报文到达预先分配的临时接收空间后,立即被拷贝到用于存储和解析的存储空间。

设计思路a)和b)具有如下缺点。

a)定时轮询会占用较多的网络主控设备CPU的资源,影响其他进程对CPU资源的使用。

b)定时轮询检测异常事件的实时性较差,检测到异常事件的时间与异常事件实际发生的时间会存在时间差。

c)在短时间内发生大量事件的情况下,由于主控处理器的处理能力限制,网络主控设备不能监测出所有事件,会丢失部分事件。

设计思想c)可以实时监听到交换芯片上报的事件报文,通过signal通信机制,在短时间内发生大量事件的情况下能监测到所有事件;并可对事件进行存储、解析和显示,提高了事件监听系统的实时性和自动化程度。

下面主要介绍该事件监听系统的详细实现流程,并搭建试验应用场景对该系统进行验证分析。

2 事件监听系统的具体设计与实现

2.1 主要组成模块

基于RapidIO交换芯片的事件监听系统运行于网络主控设备,监听来自交换芯片的事件,监听到异常事件时由网络主控设备做出响应。本系统主要包含

如下模块：系统初始化模块、signal初始化模块、中断处理模块、单次事件存储模块、事件接收模块和事件处理模块。其主要组成如图1所示。

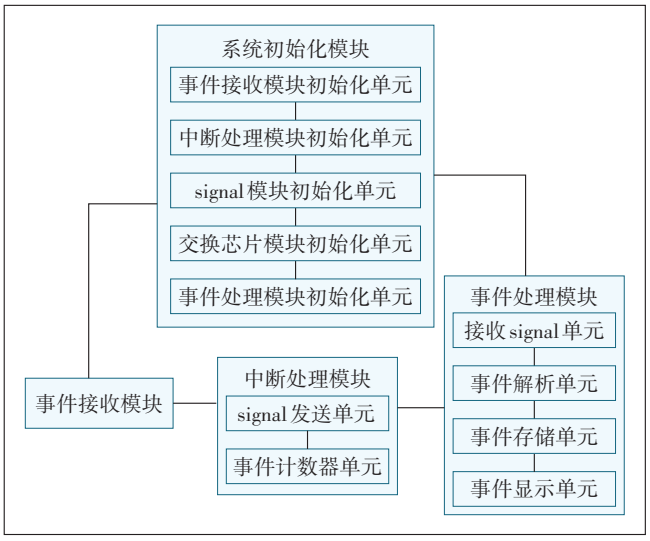

图1　事件监听系统模块组成

系统初始化模块用于开启事件监控系统时的初始化，主要包括以下5个初始化单元。

事件接收模块初始化单元：申请接收port-write报文的临时存储队列，默认为总空间大小为64 KB、起始地址64 B对齐的数组；使能主控芯片的port-write报文的接收功能；清除主控设备中port-write接收相关的标志位，并且开启snoop功能。

中断处理模块初始化单元：注册RapidIO中断服务程序；使能RapidIO中断向量。

事件处理模块初始化单元：申请事件处理存储空间，默认为struct errorInfo类型的数组，可存储64K个事件报文。

交换芯片模块初始化单元：使能交换芯片的CFG、LT、IIC、JTAG、PORT等类型事件上报功能，并且选择上报方式为向主控处理器发送port-write报文；设置port-write报文的默认目的ID和源ID。设置交换芯片的默认路由，使生成的port-write报文可以自动路由到主控处理器。

signal模块初始化单元：选择signal类型，注册signal处理函数。

中断处理模块：发送signal至事件处理模块，启动事件处理；清除中断标志位，清除接收队列标志位；更新port-write报文计数器；更新事件接收模块的存储队列指针，用于接收新的port-write报文。

事件接收模块：主要用于存储来自交换芯片的事件报文到预先申请的临时存储队列。

事件处理模块：接收到中断处理模块发送的signal，开始处理事件；从事件接收模块的临时存储队列提取事件报文，并按照port-write报文格式解析关键信息；将关键信息存储到struct errorInfo类型的数组中。

2.2 事件监听流程

事件监听流程如图2所示。交换芯片事件监听系统启动后，首先进行系统初始化，分别进行事件接收模块初始化、中断处理模块初始化、事件处理模块初始化、交换芯片模块初始化和signal模块初始化。信号（signal）是在多任务操作系统中进程间通信的一种机制，是在软件层次上模拟中断机制，而且信号是进程间唯一的异步通信机制[4]，由某个进程或者外部特殊事件触发之后发送给预先注册到这个signal的进程，使该进程启动。本系统中signal类型选择SIGRES1。因为SIGRES1是VxWorks系统中的一种signal类型，可用于用户自定义的signal通信机制，避

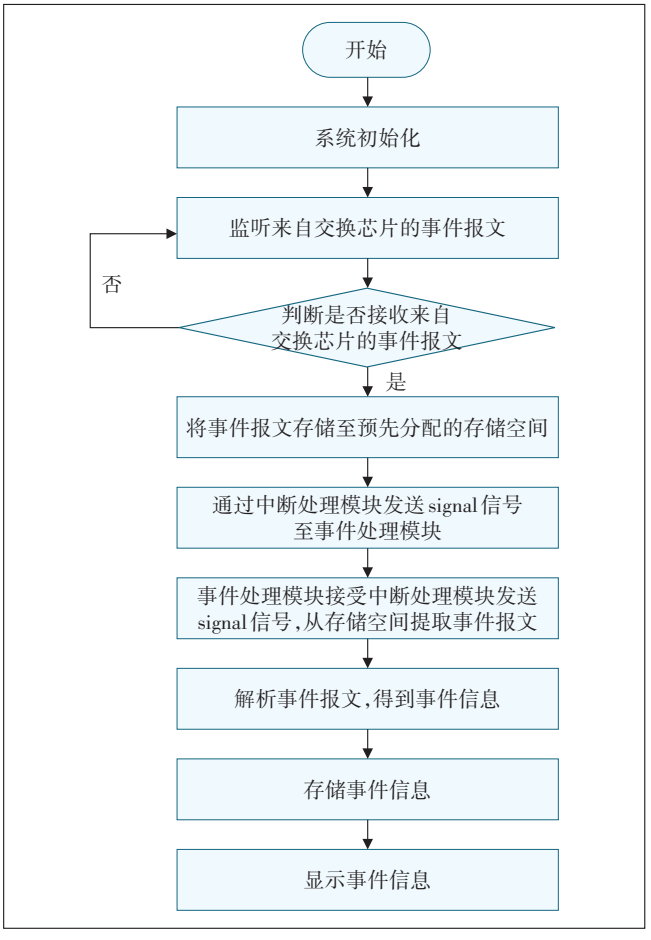

图2　事件监听流程

免和其他类型singal冲突。

系统初始化完成之后,主控处理器开始监听来自交换芯片的事件报文,当事件接收模块接收到交换芯片发来的port-write事件报文时,将事件报文存储至预先分配的临时存储队列,同时设置port-write相关状态标志位为1,可以选择增加事件计数器以记录接收到的事件报文的数量。

中断处理模块发送signal至事件处理模块,以使事件处理模块从临时存储队列提取事件报文并处理事件报文,同时清除port-write相关状态标志位。

事件处理模块解析事件报文,获取事件报文中包含的事件信息(也就是交换芯片发生的错误信息)并保存下来。可选择解析出事件信息之后立即打印出来或者只存储不打印,也可以直接根据错误信息进行错误处理或故障恢复等工作,本文中只涉及前2种操作,重点在于实时获取和存储错误信息。以上是整个事件监听系统的主要工作流程。

部分主要函数和数据结构说明如下。

RapidIOInit():初始化主控芯片的RapidIO功能。

pwInit():初始化主控芯片的port-write接收和处理功能,包括注册中断处理函数、申请接收port-write报文缓存、注册signal处理函数到指定signal等。

getPwInfo():解析、存储port-write报文信息,可以选择实时打印显示。

portWrite():初始化交换芯片的port-write功能。

portWriteGen():主动触发交换芯片若干个不同类型的错误,使交换芯片向主控处理器发送port-write报文,数量和类型自定义。

pw_counter_show():查看中断统计,port-write报文接收统计。

showAllPwInfo():查看所有接收到的port-write报文的详细内容。

struct errorInfo结构体:描述port-write的data payload详细信息定义如下。

```
typedef struct errorInformation{
UINT32 pwnum;
UINT32 cmpTagCSR;
UINT32 portErrDCSR;
UINT32 implementation_speccific_empty:8;
UINT32 implementation_speccific_err_src_code:8;
UINT32 implementation_speccific_err_code_related:8;
UINT32 portID:8;
UINT32 LTLayerErrorDetectCSR;
}errorInfo;
```

2.3 实验场景设计及分析

为了验证此系统的功能实现与可靠性,笔者搭建以下试验场景进行验证测试。

2.3.1 实验环境

硬件环境:选择Freescale的MPC8548E处理器作为主控处理器,MPC8548E是基于Freescale的PowerPC架构的PowerQuiccIII处理器,具备较强大的处理能力,较适合于高速率、低时延的业务处理。同时选择天津芯海创科技有限公司自主研发的NRS1800作为RapidIO交换芯片。NRS1800芯片满足RapidIO 2.1协议规范,支持48路通道和最多18个端口,具有低延迟、可靠数据传输和高吞吐量的特性,适合于板内互联、通过背板的板件互联与机箱互联。硬件环境拓扑连接如图3所示。

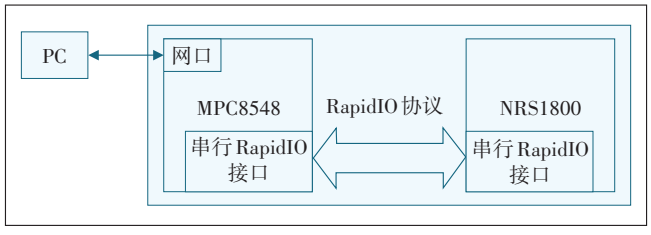

图3 测试环境硬件拓扑

软件环境:选择VxWorks作为嵌入式操作系统,VxWorks操作系统是美国公司Wind River于1983年设计开发的一种嵌入式实时操作系统(DGHI),具有良好的持续发展能力、高性能的内核以及友好的用户开发环境。IDE是Wind River公司开发的Workbench嵌入式开发平台。在PC上编码和编译,下载程序到MPC8548E处理器上运行。在ftp程序下载工具上正确设置PC机的IP地址和镜像文件在PC机上的路径,MPC8548E上电或重启就可以自动下载镜像文件。

2.3.2 测试流程及分析

系统初始化过程已经使交换芯片处于可上报port-write报文的状态,由于需要统计port-write报文数量和信息,所以采用写交换芯片错误探测寄存器的方式主动创造错误进而引起事件。交换芯片触发事件之后,统计主控处理器上的中断触发次数和接收到的port-write报文数量,结合主动触发的事件数量判断是否发生漏报问题,并查看解析出来的data payload信息

判断是否发生接收错误问题。

由于打印信息会造成延时，影响事件接收能力和实际统计结果，所以测试分以下2个层次。

a）在较慢事件触发速率的情况下，实时打印显示出事件的详细信息。

b）在较快事件触发速率的情况下，只存储和统计事件的详细信息，不再实时打印显示，待一次完整测试完成之后再全部打印显示和验证是否正确。

如果触发错误数量、触发中断数量和收到的port-write报文数量都相同，且port-write报文内容无误，则可以认为在当前的错误事件触发频率下所有的port-write报文都可以成功接收。

如果触发中断数量和收到的port-write报文数量相同，但是小于触发错误数量，且port-write报文内容无误，则可以认为只要在主控处理器中断承受能力之内的port-write报文都可以成功接收，接收能力瓶颈主要受主控处理器的中断处理能力影响。

使用本监听系统时，触发错误的间隔时间过短，会导致监听机制不能监听并解析到所有实际发出的port-write报文，将触发错误的时间间隔增加taskDelay（1）（在VxWorks系统中默认为1/60 s）延时之后，可以监听并解析到全部port-write报文（测试0x100个port-write包），随后将错误触发的时间间隔缩短到1/6 000 s（167 μs）左右时为不丢包的临界值，测试记录如表1所示。

表1　事件监听测试结果

触发错误时间间隔/s	触发错误次数	触发中断次数	收到port-write包次数	data payload是否正确(Y/N)
1/60	256	256	256	Y
1/1 000	256	256	256	Y
1/2 000	256	256	256	Y
1/3 000	256	256	256	Y
1/4 000	256	256	256	Y
1/5 000	256	256	256	Y
1/6 000	256	256	256	Y
1/6 100	256	255	255	Y
1/6 200	256	254	254	Y
1/7 000	256	254	254	Y
1/8 000	256	251	251	Y

上述实验结果表明本系统将signal这种进程间的通信机制与中断结合应用到RapidIO交换芯片的事件监听解决方案中，可以实时地监听到上报的事件；在触发错误事件时间间隔不小于1/6 000 s的时候触发大量事件的情况下能监测到所有事件，可以自动解析、存储、显示上报的事件；当触发错误事件时间间隔小于1/6 000 s的时候就会发生丢包现象，但是触发中断的数量和收到的port-write报文数量相同。

经过分析，主控处理器丢包原因主要有2个：一是MPC8548E处理器的中断处理能力和运算能力相对于高速率触发错误事件的应用场景较低，在高速率触发中断的情况下承受能力有限，高速率接收port-write报文会发生丢包问题；二是事件监听系统软件架构应该还有进一步优化的空间，可以在一定程度上提高port-write报文接收速率。

3　结束语

本文提出的RapidIO交换芯片的事件监听系统是基于VxWorks系统，采用signal通信机制结合中断的设计方案，可以实时监听到交换芯片上报的事件报文，在短时间内发生大量事件上报的情况下能监测到所有事件，并且可以通过事件处理模块自动解析事件报文、存储事件信息、显示事件信息，提高了RapidIO事件监听系统的实时性和自动化程度。而且，该方法无需网络主控设备的处理器定时轮询事件报文的存储空间或者主动查询交换芯片的错误状态寄存器，很大程度地节约了网络主控处理器的资源，有利于网络主控处理器更高效、更稳定地维护RapidIO网络。

同时，由于在交换芯片高速率事件触发的场景下存在性能瓶颈，所以优化工作需要继续开展。后续计划采用性能更强的处理器作为主控设备，缓解中断处理能力瓶颈，并且优化软件设计方案，缩短能承受的事件触发时间间隔。

参考文献：

[1]　邓豹，赵小冬．基于串行RapidIO的嵌入式互连研究[J]．航空计算技术，2008，38(3)：123-126.

[2]　FULLER S，COX T. Anatomy of a forward-looking open standard [RapidIO][J]. Computer，2002，35(1)：140-141.

[3]　张娟娟．RapidIO高速串行总线的研究与实现[D]．长沙：湖南大学，2011.

作者简介：
崔超，工程师，硕士，主要研究方向为计算机软件；张明权，在读研究生，主要研究方向为计算机软件；于洪，助理研究员，硕士，主要研究方向为计算机软件，网络安全；肖峰，学士，主要研究方向为计算机软件。