

一种多核处理器中断控制器的设计

张海金, 张洵颖, 肖建青

(西安微电子技术研究所, 陕西 西安 710065)

摘 要: 为适应多核处理器对中断处理的需求, 基于 Open PIC 协议设计实现了一种多核处理器的中断控制器, 并使用 VHDL 语言对其进行了硬件描述. 该中断控制器作为 APB 从机, 能够根据中断的目标、优先级的配置情况以及处理器核的中断处理情况实现中断在多个处理器核间的自由分配. 本文将中断仲裁、选择和分配进行流水化处理, 从而实现中断的快速准确分配.

关键词: 中断仲裁; 中断选择; 中断分配; 流水

中图分类号: TP302.2

文献标识码: A

文章编号: 1000-7180(2016)07-0069-05

DOI:10.19304/j.cnki.issn1000-7180.2016.07.015

Design of An Interrupt Controller for Multi-core Processor

ZHANG Hai-jin, ZHANG Xun-ying, XIAO Jian-qing

(Xi'an Microelectronics Technology Institute, Xi'an 710065, China)

Abstract: To meet the needs of multi-core processor for interrupt handling, this paper designs and implements an interrupt controller for multi-core processor based on the Open PIC protocol. The structure of the interrupt controller is described by VHDL. As a slave of Advanced Peripheral Bus (APB), this interrupt controller can dispatch interrupts based on the interrupts' destination and priority, the interrupt handling of the processor cores is also considered. With the pipeline of interrupt arbitration, selection and dispatch, this interrupt controller can dispatch interrupts quickly and accurately.

Key words: interrupt arbitration; interrupt selection; interrupt dispatch; pipeline

1 引言

由于处理器核与外部设备之间主要通过中断进行通信, 所以中断管理策略成为了处理器研究中不可忽视的问题. 针对多核环境下的中断管理, 很多厂商提出了各自的中断控制器体系结构. 例如, Intel 在多核处理器协议中运用大量篇幅描述了 APIC 的中断管理方案^[1], ARM 提出了 GIC 协议^[2], IBM 基于 Open PIC 协议提出了 MPIC 协议^[3]. 此外, 很多研究人员也提出了自己的多核中断管理方案. 文献^[4]设计实现了一种用于异构多核处理器核间通信的多核中断控制器, 但未实现中断在多个处理器内核间的自由分配功能; 文献^[5]设计的多核中断控制器实现了中断的广播传送, 但其广播传送目标为所有处理器核, 不可配置. Open PIC 协议^[6]提出的可

编程中断控制器体系结构不仅实现了中断在多个处理器核间的自由分配, 也实现了目标可配置的定时器中断、核间中断的广播传送. 因此, 本文基于 Open PIC 协议设计实现了一种作为 APB 从机的多核处理器中断控制器, 使其能够应用于以 AMBA 为片上总线多核处理器系统中. 该中断控制器支持协议规定的两种中断传送方式, 即直接式传送和分布式传送, 并在协议基础上实现了中断仲裁、选择和分配的流水化操作. 该中断控制器能够根据中断目标情况和处理器核中断处理情况实现中断在多个处理器核间的自由分配.

2 多核中断控制器的结构设计

该中断控制器主要包括 Timer 中断产生模块、核间中断产生模块、全局配置模块、中断捕获模块、

收稿日期: 2015-09-15; 修回日期: 2015-10-16

中断仲裁模块、中断选择模块以及中断分配模块,其结构如图 1 所示。

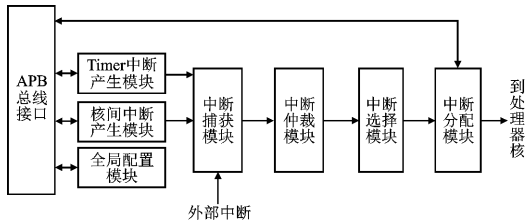


图 1 中断控制器结构框图

用户可以通过 APB 总线接口访问全局配置模块对中断源的中断目标、中断向量、中断优先级以及中断屏蔽与否进行设置。Timer 中断产生模块为多个内核提供了基于中断的任务同步机制,用户可以通过 APB 总线接口配置 Timer 的计数基数并使能 Timer 减数计时,当计数到 0 时,Timer 将产生中断并开始新一轮的计时。核间中断产生模块提供了多核之间相互中断甚至是自中断的机制,用户可以通过 APB 总线接口配置相应寄存器来产生目标为一个或者多个处理器核的核间中断,甚至是自中断。

在该中断控制器中,Timer 中断和核间中断都是上升沿敏感的,而外部中断的触发方式则可以通过 APB 总线接口进行设置。中断捕获模块根据用户设置的外部中断触发方式以及 Timer 中断和核间中断的产生情况对所有中断进行捕获,将捕获结果传递给中断仲裁模块。中断仲裁模块根据中断的优先级、中断目标进行仲裁,选择出以某一处理器核为目标的具有最高优先级的一组中断传送给中断选择模块。中断选择模块基于平等原则从中断仲裁模块输出的一组中断中选择出某个中断,计算其索引值,将其传递给中断分配模块。中断分配模块根据中断索引值、中断目标以及处理器核的中断处理情况将该中断分配给某一个处理器核。处理器核接收到中断分配模块分配的中断后,通过 APB 总线读中断应答寄存器(IAR)对中断进行应答,中断控制器将处理器核应答的中断置为 activity 态,此时该中断的相关寄存器将不能被改写。当中断处理结束时,处理器核需要写中断结束寄存器(EOD)通知中断控制器中断处理结束,中断控制器取消中断的 activity 态。

该中断控制器的中断仲裁模块、中断选择模块和中断分配模块实现中断传递的流水化处理,在经过开始的流水线排空后,每个时钟周期都有一个中断被分配给处理器核,提高了中断分配效率。其中,中断仲裁模块在一个时钟周期内完成,中断选择模

块的第一个时钟周期根据前次中断传送的结果将中断输入进行移位操作,第二个周期进行有效中断的查找并给出中断索引值,中断分配模块的第一个时钟周期根据中断索引值确定中断分配目标、生成相关寄存器的操作信号,第二个周期根据第一个周期生成的寄存器操作信号改写相关寄存器的值并生成中断信号。

2.1 中断仲裁模块

该模块首先根据中断的优先级、中断目标进行目标优先级译码,将中断捕获模块捕获的所有中断进行分组,以处理器核 X 为目标优先级为 Y 的中断被分配到 cpuX_prioY 组内,在这里优先级为 0 的中断将被忽略,因为这样的中断永远不可能被某个处理器核处理。随后,根据 cpuX_prio1 、 cpuX_prio2 、……、 cpuX_prio15 的情况进行仲裁,选择出以处理器核 X 为目标的最高优先级组 $\text{cpuX_highest_prio_group}$ 。同时,判断每一个处理器核的中断情况,给出所有处理器核有无中断的有效信号 cpu_valid 。最后,根据 cpu_valid 运行处理器核选择状态机选择某个处理器核,将该处理器核的编码以及相应的最高优先级中断组传送给中断选择模块,其结构如图 2 所示。

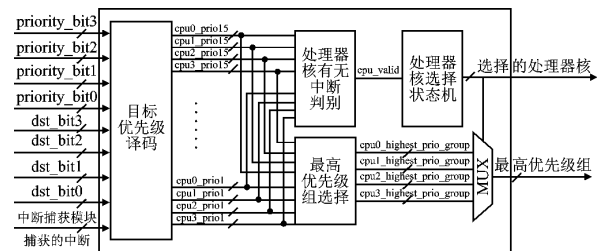


图 2 中断仲裁模块结构图

在图 2 中,处理器核选择状态机根据当前状态以及 cpu_valid 的值选择某一处理器核,中断仲裁模块根据选择的处理器核选择最高优先级组传递给中断选择模块。假设当前状态为 0100,则上次选择的处理器核为 2# 处理器核。状态机首先判断 3# 处理器核有无中断 ($\text{cpu_valid}(3) = 1$) 与否,若 $\text{cpu_valid}(3) = 1$,则选择 3# 处理器核,若 $\text{cpu_valid}(3) = 0$,则判断 0# 处理器核的中断有无情况,有则选择 0# 处理器核,无则判断 1# 处理器核,以此类推。处理器核选择状态机的状态转换图如图 3 所示。

2.2 中断选择模块

中断仲裁模块输出的以某一处理器核为目标的最高优先级中断组中的所有中断具有相同的优先级,中断选择模块在这组中断中选择出其中一个中

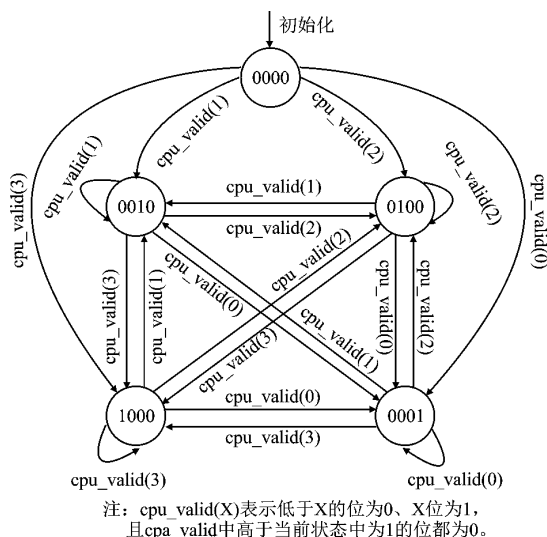


图3 处理器核选择状态转换图

断,将其中断索引值传送给中断分配模块进行分配。如果该模块接收到的中断为 $0000 \cdots 0100 \cdots 0000$ (136位,每位对应一个中断,为1则表示该中断有效),则从中断0开始查询,查询到第一个1后返回该中断的索引值。但是,如果每次都从中断0开始查询则会出现部分中断被阻塞的情况。例如,若上次接收到的中断为 $1000 \cdots 0010$ (136位),本次接收到的中断为 $1000 \cdots 0001$ (136位),两次得到的中断索引值分别为1和0,可见中断135将被阻塞,在极端情况下该中断有可能永远不会被传送,这就使得具有相同优先级的中断被传送的机会不均等。为防止上述情况发生,本设计在该模块中为每一个处理器的每一个优先级设置一个已传送中断索引值加1寄存器,每次查找有效中断之前先将上次传送的中断置于查找队列的最后,从上次传送的中断的下一个中断开始进行中断查询,查询到第一个有效中断后将得到其相对于上次传送中断的位置,据此可以确定当前查询到的中断的索引值,将其传送到中断分配模块的同时将该值加1寄存,以便下次查找时应用。其过程如图4所示,为方便说明,假定要查询的中断共有16个。

由图4可以看出,中断索引值确定的过程中最重要的是中断的移位操作和第一个有效中断的查找操作。其中,中断的移位操作可以通过桶形移位器实现;第一个有效中断的查找操作,如果出现当前所要查找的中断位于上次传送结果的前一个的情况,就需要对135个中断的情况进行检测,此时的查找过程将会变得缓慢。本设计采用分组分级并行查找的方式对第一个有效中断进行查找。首先,将136个中

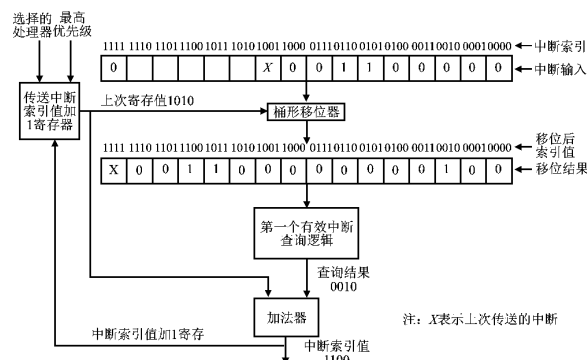


图4 中断索引值的确定

断分成每8个中断为一组的17组,各组查找组内第一个有效中断确定其在组内的编号,接着将低16组的查找结果有效位分成两组进行再次查询,得到查询结果后根据查找情况确定第一个有效中断在移位后的中断组中的索引值,根据该值和上次寄存的结果就能够确定第一个有效中断的索引值。每组内8个中断的查找模块 Detect 的输入输出关系如表1所示,输出最高位表示检测到有效中断与否,1有效。第一个有效中断的查找和索引值确定结构如图5所示。

表1 Detect 模块的输入输出关系

| 输入 | 输出 | 输入 | 输出 |
|-----------|------|----------|------|
| XXXXXXXX1 | 1000 | XXX10000 | 1100 |
| XXXXXX10 | 1001 | XX100000 | 1101 |
| XXXXX100 | 1010 | X1000000 | 1110 |
| XXXX1000 | 1011 | 10000000 | 1111 |
| 其他 | 0000 | —— | —— |

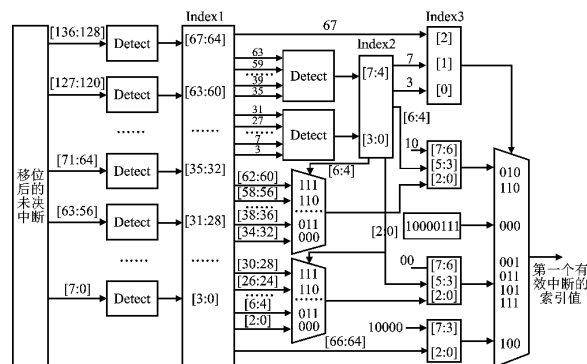


图5 第一个有效中断查找逻辑

2.3 中断分配模块

该模块首先根据中断仲裁模块输出的中断索引值查询该中断的目标从而确定该中断的分配方式,

接着根据其分配方式、中断优先级以及目标值确定可用的目标处理器,根据所有目标处理器的当前状态确定是否对该中断进行分配.在中断分配模块中,最重要的是中断可用目标处理器以及中断分配方式判断,其判断流程如图 6 所示.

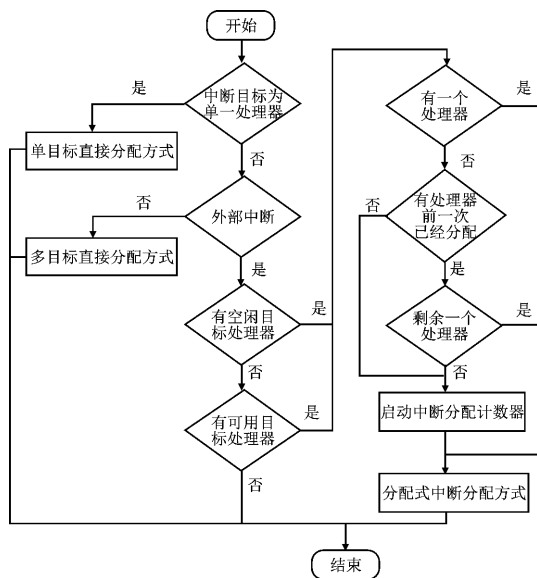


图 6 中断分配和目标处理器判断流程

在该模块中,每一个处理器都有一组中断处理寄存器与之对应,这些寄存器包括中断请求寄存器(IRR)、中断应答寄存器(IAR)、处理器当前优先级寄存器(CTPR)、正在处理中断寄存器(ISR)以及中断结束寄存器(EOI).其中,IRR 寄存中断的优先级和中断索引值;IAR 寄存中断的中断向量,处理器可以读该寄存器获得中断向量并对中断进行应答;CTPR 寄存处理器当前的任务优先级,优先级低于 CTPR 值的中断将被处理器屏蔽;ISR 有 15 位,每一位对应一个优先级,用来指示当前处理器正在处理中断的优先级,也用来记录处理器的中断嵌套情况;EOI 寄存器是只写寄存器,处理器通过写该寄存器来通知中断控制器中断处理结束,中断控制器将清除 ISR 中优先级最高的有效位.

3 仿真验证

为验证多核中断控制器的功能正确性,构建如图 7 所示的中断控制器验证环境.其中,多核处理器的运行程序存储于外部存储器,存储器控制器根据 AHB 解析的主机命令实现对外部存储器的访问,AHB 到 APB 桥将 AHB 总线时序转化为 APB 总线时序,多核中断控制器挂载在 APB 总线上.如果此时有 8 路外部中断源,其配置情况如表 2 所示,所

有处理器核的任务优先级配置为 2,定时器中断晚于外部中断产生.

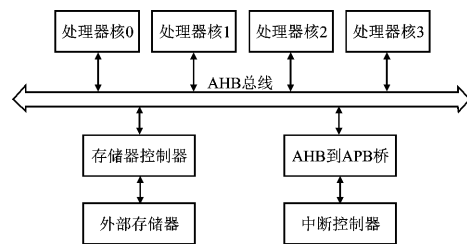


图 7 多核中断控制器仿真环境架构

表 2 中断源配置情况

| 中断源 | 中断优先级 | 中断目标 |
|--------|-------|--------------|
| 外部中断 0 | 3 | 处理器核 0 |
| 外部中断 1 | 2 | 处理器核 0、1 |
| 外部中断 2 | 4 | 处理器核 0、1 |
| 外部中断 3 | 5 | 处理器核 2、3 |
| 外部中断 4 | 9 | 处理器核 0、1、2 |
| 外部中断 5 | F | 处理器核 1、2、3 |
| 外部中断 6 | 7 | 处理器核 0、2、3 |
| 外部中断 7 | A | 处理器核 0、1、2、3 |
| 定时器中断 | E | 处理器核 1、2、3 |

当多路外部中断同时产生时,根据中断源的配置情况可知:中断控制器将优先级最高的外部中断 7 分配给处理器核 0,随后到来的以处理器核 0 为目标的所有中断均被该中断屏蔽,直到处理器核 0 响应了该中断,后续中断才进入 IRR 寄存器等待处理器核响应;以处理器核 1 为目标的中断中,外部中断 5 具有最高的优先级,所以该中断将优先分配给处理器核 1;由于处理器核 0 和 1 中都有高于外部中断 4 的优先级的中断正在处理,所以外部中断 4 将分配给处理器核 2;与外部中断 4 类似,外部中断 6 将分配给处理器核 3 进行处理.图 8 给出了一种中断处理情况下的中断分配结果,可以看出每个处理器核的第一个中断分配情况与上述分析一致,后续的中断分配情况与每个处理器核的中断处理情况有关.在图 8 所示的仿真结果中,由于外部中断 1 的优先级等于所有处理器核的任务优先级,该中断将被屏蔽;当定时器中断产生时,所有优先级低于该中断的外部中断将被屏蔽(未进入 IRR 等待的中断)或取代(已经进入 IRR 等待的中断),定时器进入该处理器核的 IRR 等待处理;当每个处理器核处理完定

时器中断后,优先级低于该中断的外部中断才能够被处理器核响应。图8标示出了中断产生、分配、应答、嵌套情况,这里不再赘述。仿真验证证明,本文设

计的多核处理器中断控制器功能正确,实现了中断的快速准确分配。



图8 仿真验证结果

4 结束语

本文通过中断仲裁、选择和分配的流水化操作,设计实现了一种适用于多核处理器的中断控制器。该中断控制器最多支持128路外部中断、4路定时器中断、4路核间中断,最多支持4个处理器核的中断分配。由仿真验证结果可知,该中断控制器能够根据中断目标、处理器核的中断处理情况对中断进行快速准确地分配,完全实现了Open PIC协议规定的中断控制器功能。

参考文献:

- [1] Embleton S, Sparks S, Zou C C. SMM rootkit: a new breed of OS independent malware[J]. Security and Communication Networks, 2013, 6(12): 1590-1605.
- [2] Dall C, Nieh J. KVM/ARM: the design and implementation of the linux ARM hypervisor[J]. ACM SIGARCH Computer Architecture News, 2014, 42(1): 333-348.
- [3] Multiprocessor interrupt controller data book, revision 1.2[R]. New York: International Business Machine Corporation, 2011.

- [4] 王龙杰, 胡越黎, 刘廷尧, 等. 多核 MV12 片上系统中中断控制器的设计与实现[J]. 半导体技术, 2012, 37(10): 755-759.
- [5] Tumeo A, Branca M, Camerini L, et al. An interrupt controller for FPGA-based multiprocessors[C]// Proceedings of 2007 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (IC-SAMOS 2007). USA, Samos, 2007: 82-87.
- [6] The open programmable interrupt controller (PIC) register interface specification, revision 1.2[R]. US: Advanced Micro Devices and Cyrix Corporation, 1995.

作者简介:

张海金 男, (1987-), 硕士研究生. 研究方向为微处理器设计. E-mail: xuansong004@126.com.

张洵颖 男, (1981-), 研究员, 硕士生导师. 研究方向为计算机系统结构.

肖建青 男, (1985-), 博士研究生. 研究方向为微处理器设计.