

Round robin 调度算法在 FPGA 中的实现¹

孙华锦 高德远 张盛兵

(西北工业大学航空微电子中心 西安 710072)

摘 要 Round robin 调度算法是一个在许多方面有着广泛应用的经典调度算法。该文在考虑了 FPGA 的结构特点和实际系统需求后,利用桶式移位器和分段式优先级编码器,在 FPGA 中实现了 Round robin 调度算法,并对实现方法的面积和性能进行了讨论。系统测试结果表明该算法实现是高效的,满足了系统的需求,在实际系统中运行状况良好。

关键词 Round robin, 调度算法, FPGA

中图分类号 TN791

1 引 言

Round robin 调度算法作为一个经典的调度算法在计算机系统,网络应用方面有着广泛的应用。由于其实现相对简单,调度公平性比较好,在许多方面尤其是网络交换的数据包调度等场合现在仍有着大量的应用。

现场可编程门阵列(Field Programmable Gate Array, FPGA)是高密度现场可编程逻辑芯片,能够将大量的逻辑功能集成于一个单片集成电路中。FPGA 以其操作灵活、使用方便、开发迅速、投资风险小等突出优点,在数字系统设计领域得到越来越广泛的应用。当然,用 FPGA 实现数字系统也有其自身的局限性,其一是器件规模的限制,其二是单元延迟比较大。所以,这要求设计者对所选器件的结构、性能有深入的了解,在进行系统结构设计时,必须考虑到器件本身的结构及性能,尽可能使设计的结构,满足器件本身的要求,这样才能使设计在速度和资源利用两方面都有比较好的结果。

用 FPGA 实现 Round robin 调度算法是网络交换系统的一个实际需求,本文根据 FPGA 的结构特点,综合考虑了系统速度和资源利用两个方面的因素,提出了一个利用桶式移位器(Barrel Shifter, BS)和分段式优先级编码器实现 Round robin 的方法,满足了系统速度和资源的要求。

本文首先简要介绍 Round robin 调度算法,并指出了实现 Round robin 调度算法的关键所在。然后,着重讨论 Round robin 调度算法的实现,包括桶式移位器、分段式优先级编码器等重要部件的设计。最后,简要介绍 Verilog 描述、仿真、综合和 FPGA 实现的情况和结果。

2 Round robin 调度算法简介

Round robin 调度算法,又名时间片轮循法,是一个经典的调度算法。在分时系统中,算法将 CPU 的处理时间分为时间片(Time slice),一个时间片从几毫秒到几百毫秒不等。系统给每一个进程分配若干个时间片,被调度选中的进程进行完了系统分配的时间片后,系统就会发生调度,如果该进程还未完成要求的任务,则调度程序暂时停止该进程的执行,将它排到就绪队列的末尾,等待下一次调度。同时,调度程序会调度当前就绪队列中的下一个进程。这样就可以保证就绪队列中的所有进程在一个给定的时间均能得到执行。

Round robin 调度算法也广泛应用于网络交换处理系统中。在网络交换处理系统中,时间片比较固定,一般为系统传输一个信元的时间。而各个信元所在的不同的通道相当于分时系统中的进程。调度过程也类似,其示意图如图 1 所示。

在图 1 中,假设系统有 8 个需调度的通道,调度的起点在 A 通道,通道中的信元数目各不相同。首先,由于 A 通道有信元可发送,故第 1 轮调度出 A 通道,然后在第 2 轮调度中,先检查 B 通道,由于 B 通道没有信元可发送,所以跳过 B 通道,检查 C 通道,发现 C 通道有信元,因而 C 被选出。第 3 轮调度,先从 D 通道开始检查,从而调度出 D 通道,以此类推。其中第 6 轮调度完成了对所有通道的检查,重新从 A 通道开始检查;第 8 轮调度由于 D 通道在

¹ 2002-03-04 收到, 2002-07-29 改回

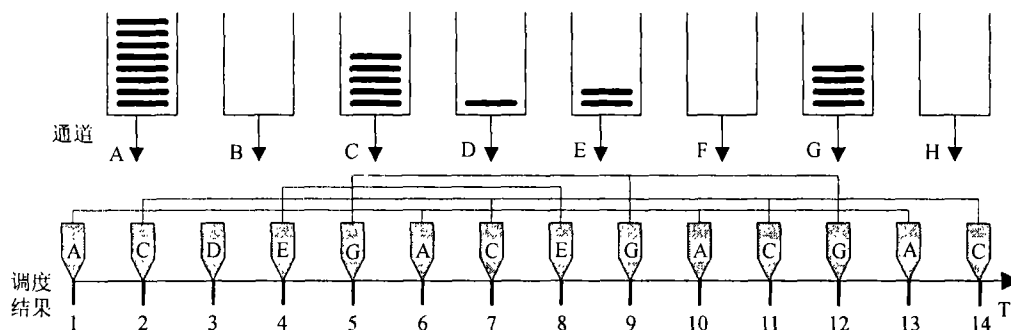


图1 Round robin 调度算法示意图

第3轮调度中把一个信元发送出去后处于无信元可发状态,从而调度出了E通道。从图中可以看出要实现 Round robin 算法,关键是从起始的通道开始,如何找到第一个“准备好”(有信元可发送)的通道。

3 Round robin 调度算法在 FPGA 中的实现

正如文中第二节所述, Round robin 调度算法实现的关键是对于给定的开始通道号,如何快速,高效的找到下一个已经准备好的通道。因此如何在 FPGA 中有效地解决这个问题成为我们设计的重点。

文献 [1] 中提出了几种实现 Round robin 调度算法的方法: (1) 用链表结构实现 Round robin, 其结构规整, 可扩展性比较好, 但实现起来也比优先级编码复杂, 且要占用 RAM 资源; (2) 用树型结构来实现 Round robin, 这是一个比较新型的结构思路, 可扩展性也比较好, 但是它的调度周期比较长且不固定, 而且同样需要占用 RAM 资源; (3) 基于令牌的优先级编码, 这是最简单的实现方法, 但当需调度的通道数比较大时, 其延迟时间太长。

Round robin 算法应用的实际系统主要功能是完成网络信元格式的转换, 包括了流控和 CPU 接口。作为整个系统的一部分, 算法实现考虑当前系统的实际情况: 信元缓存和系统状态表占用了大量的 RAM 资源, RAM 资源比较紧张; 系统所用通道数目前不会超过 256 个; 一次调度的周期不能过长, 系统速度要求比较高, FPGA 内部时钟要达到 100MHz; 系统功能比较复杂, 资源占用比较多, 要求算法的实现尽量减少资源的占用。因而文献 [1] 中用链表结构和用树型结构来实现 Round robin 都由于需要占用 RAM 资源而不适合在本系统中使用, 基于令牌的优先级编码由于系统的通道数目会达到 256 个, 故其实现延迟会很大, 无法满足系统速度的要求, 也不能在本系统中直接使用。在综合权衡资源和速度需求, 结合本系统的实际情况后, 本文设计了一种利用桶式移位器和分段式优先级编码器来实现 Round robin 调度算法的方法。此方法逻辑实现上比较简单, 节省了 FPGA 内部资源, 同时又考虑了电路的延迟, 在关键路径上做了优化措施, 满足了系统整体速度的要求。在资源和系统速度允许的情况下, 对于需调度通道数的扩展也是比较方便的。

3.1 逻辑结构的设计与实现 实现 Round robin 调度算法的逻辑结构框图如图 2 所示。通道状态寄存器用来存放需调度通道是否准备好的状态, 它是 Round robin 调度的基础, 准备好的通道才可以被调度上, 没准备好的则不可能被调度出来。以 128 个通道为例, 通道状态寄存器就是 128 位的寄存器, 哪一位为 ‘1’ 表示哪一个通道已经准备好, 其状态位的设定由系统的其他逻辑实现。

桶式移位器根据调整逻辑送过来的移位次数在一个或数个时钟周期内对通道状态寄存器完成移位操作。具体分几个时钟周期来完成移位操作主要看通道状态寄存器的位数, 寄存器位数比较多时, 若在一个时钟周期内完成移位操作, 电路的延迟太大, 满足不了系统对速度的要求,

那么就需要分几个时钟周期来完成移位操作,以减小电路延迟。

桶式移位器完成移位操作后,将所得结果送到分段式优先级编码器进行优先级编码。同样考虑到系统速度的要求,借鉴文献 [1] 中树型结构实现的思路,设计了一个分段式优先级编码器来减小电路延迟,以满足系统速度的要求。

调整逻辑主要是一个累加器,它把分段式优先级编码器的编码结果加 1 后进行累加,这就相当于把每次 Round robin 调度的开始点给保存在累加器中。累加器的位数由调度通道数决定,128 个通道需要 7 位的累加器,其电路延迟可以满足系统要求,不需要进行特别的优化。

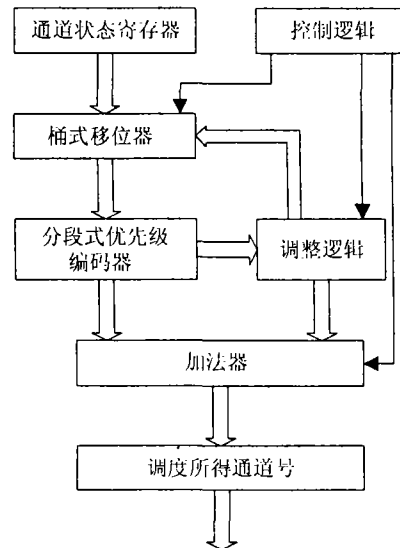


图 2 算法实现逻辑结构图

最后把分段式编码器的编码结果和调整逻辑保存的调度开始点相加就可以得到本次调度所得的通道号,完成本次调度。

此外,算法实现的控制逻辑由有限状态机 (FSM) 来实现,它控制整个调度过程的实现,包括每次调度的开始,调整逻辑的累加,调度结果的取得等操作。

从逻辑结构框图上可以看出,通过调整逻辑保存的调度开始点对调度状态寄存器进行移位操作,这就把每次调度的开始点移到了调度状态寄存器的首位,然后在对移位后的调度状态寄存器进行优先级编码,将编码结果与调度开始点相加就得到了本次调度的通道号,从而实现了 Round robin 调度算法,其中桶式移位器和分段式优先级编码器是算法实现的关键路径,其电路延迟的大小决定了算法实现对系统速度的影响,需要加以优化处理。

3.2 桶式移位器 桶式移位器是现代高速 RISC 微处理器芯片中普遍采用的结构,它能使各种位数的移位操作都能在单周期内完成。这对于算法的实现是很关键的,因为若是采用普通的移位器,128 位的调度状态寄存器极限情况须移 127 位,则需要 127 个时钟周期才能完成,如此长的调度周期对算法的实现来说是不可接受的。

一般说来,桶式移位器的实现方案有全译码方式、全编码(不译码)方式和部分译码方式等多种。全译码方式实现简单,资源占用比较少,可以在一个时钟周期内实现移位,但移位位数较多时,电路时延较大。全编码方式的电路时延小,但其资源占用较多,实现移位需多个时钟周期(由具体移位位数决定)。部分译码方式介于前两种方式之间,在电路时延和资源占用两方面做了个折衷。表 1 是用 3 种方案实现 128 位移位的结果比较。

表 1 128 位桶式移位器实现性能对比

实现方法	时钟频率 (MHz)	Slice 使用数 (个)	寄存器使用数 (个)	Look Up Table (LUT) 使用数 (个)	实现周期数
全译码方式	73.6	284	128	568	1
全编码方式	130.11	448	896	131	7
部分译码方式	102.1	303	256	606	2

注:表中编码是在 Xilinx 公司 Virtex-E 系列器件 xcv600efg676-6^[2] 上实现的

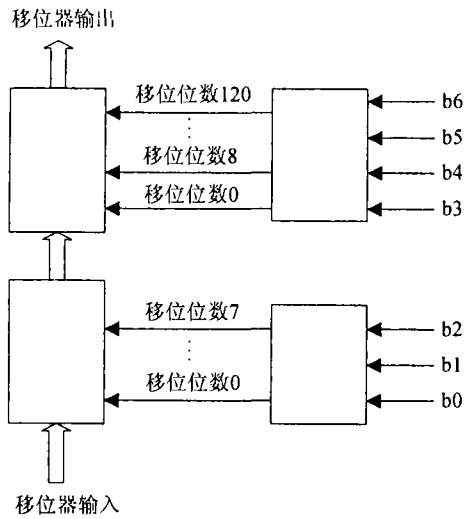


图 3 桶式移位器结构图

从表 1 中可以看出，全译码方式的时延太大，不适合本系统使用；全编码方式实现周期数太长，且资源占用较多，也不适合在本系统中使用；只有部分译码方式在性能和资源占用方面都可以接受，符合本系统的要求，是一种较优的设计。对于 128 位的桶式移位器来说，移位分成两部分，第 1 部分（移位控制的低 3 位）的译码控制移 0, 1, 2, 3, 4, 5, 6, 7 位，第 2 部分（移位控制的高 4 位）的译码控制移 0, 8, 16, ..., 112, 120。其结构如图 3 所示。这样完成各种移位次数 (0-127) 的移位操作需 2 个时钟周期。其电路延迟经两级分解后也达到了系统速度的要求。

3.3 分段式优先级编码器

分段式优先级编码器的设计借鉴了文献 [1] 中用树型结构

来实现 Round robin 的思路。将需要进行优先级编码的状态位进行分段，先对每一段的状态位进行优先级编码，得到编码结果和一个状态位，然后再对每个段的状态位进行优先级编码，最后将两次编码的结果合起来就得到最后的优先级编码结果。以 32 个状态位为例，图 4 给出了分段式优先级编码器的结构示意图。

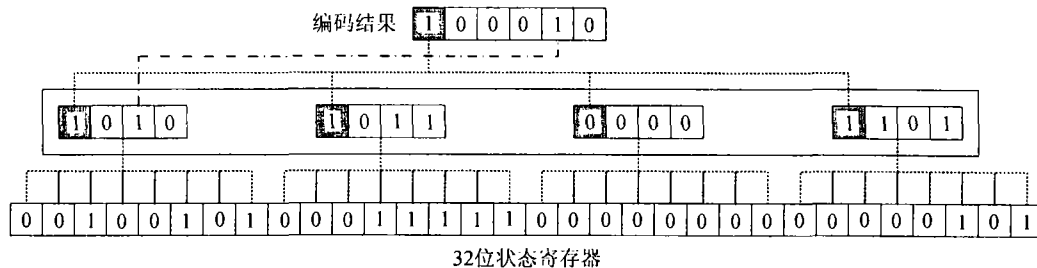


图 4 分段式优先级编码器结构示意图

图 4 中把 32 个状态位分成 4 段，每段 8 位。先对每段作一个 8 位的优先级编码，得到编码结果和状态位，然后对 4 个状态位进行一个 4 位的优先级编码，最后将 2 个编码的结果合到一起就得到最后的编码结果。这样就把一个 32 位优先级编码分解成 4 个 8 位优先级编码和一个 4 位优先级编码，电路时延大大减少。表 2 是 128 位优先级编码各种实现方法的对比，其中二级分段式优先级编码是把一个 128 位的优先级编码分解成了 8 个 16 位优先级编码和一个 8 位优先级编码；三级分段式优先级编码是把一个 128 位的优先级编码分解成了 16 个 8 位优先级编码和 5 个 4 位优先级编码。从表中数据可以很清楚的看出，无论是电路时延，还是资源占用分段式优先级编码都有了明显的改善。这证明分段式优先级编码器在 FPGA 实现上是行之有效的，符合了 FPGA 的结构特点，取得了比较明显的效果。同时也可以看出三级分段式优先级编码比二级优先级编码在速度上占优势，但资源占用比较多。在满足系统速度的情况下，应尽量减少资源占用，所以本次算法实现选用了二级分段式优先级编码。

表 2 128 位优先级编码实现性能对比

实现方法	延迟时间 (ns)	时钟频率 (MHz)	寄存器使用数 (个)	LUT 使用数 (个)
直接编码	23.972	41.7	0	282
分段式 (二级)	7.865	127.1	40	147
分段式 (三级)	7.565	132.2	88	180

注: 表中编码是在 Xilinx 公司 Virtex-E 系列器件 xcv600efg676-6^[2] 上实现的

4 结 论

我们用 Verilog 硬件描述语言^[3] 对算法的实现进行了描述, 用 Synplify 进行综合, 用 ModelSim 进行仿真, 仿真模拟的结果符合预先设计的时序要求。最后用 Xilinx 公司的 Foundation 进行了布局布线, 选用的器件为 Virtex-E 系列的 xcv600efg676-6^[2]。其结果如表 3 所示。

表 3 算法实现布线结果表

时钟周期 (ns)	时钟频率 (MHz)	Slice 使用数 (个)	LUT 使用数 (个)	等效门数 (个)
9.831	101.719	583 (8%)	1083 (7%)	9734

从表 3 中数据可以看出, 算法实现的布线结果满足系统速度的要求, 其资源占用率也可以接受。系统测试结果表明算法实现实际系统中运行良好。此外, 在其他应用 Round robin 调度算法的场合, 该算法实现也有很大的参考价值。

参 考 文 献

- [1] M. G. H. Katevenis, Fast switching and fair control of congested flow in broadband networks, IEEE on Selected Areas Comm., 1987, SAC-5(8), 1315-1326.
- [2] The Programmable Logic Data Book 2000, Xilinx Incorporation, 2000, Section 3.
- [3] 夏宇闻, 复杂数字电路与系统的 Verilog HDL 设计技术, 北京, 北京航空航天大学出版社, 1998, 第二章, 第四章.

THE FPGA IMPLEMENTATION OF THE ROUND ROBIN SCHEDULING ALGORITHMS

Sun Huajin Gao Deyuan Zhang Shengbing

(Aviation Microelectronics Center, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract Round robin scheduling algorithm is a classic scheduling algorithm with many applications. An FPGA implementation by using barrel shifter and pipelined priority encoder is presented in this paper with considering the FPGA structure characteristic and system's requirement. And the performance and resource consumption of the implementation are discussed. The test result shows that the implementation of the algorithm is effective and fit for the FPGA structure. The system with the implementation of algorithm runs very well.

Key words Round robin, Scheduling algorithm, FPGA

- 孙华锦: 男, 1977 年生, 博士生, 主要研究方向: 计算机体系结构、数字系统设计、微处理器设计及 ASIC 系统设计等。
- 高德远: 男, 1946 年生, 教授, 博士生导师, 主要研究方向: 计算机体系结构、VLSI 系统设计、ASIC 系统设计、电子设计自动化技术及计算机网络等。
- 张盛兵: 男, 1967 年生, 博士, 副教授。主要研究方向: 计算机体系结构、微处理器设计和测试技术、ASIC 设计与测试技术等。