文章编号:1001 - 9081(2009)06 - 1483 - 04

一种基于堆栈存储的 RFID 防冲突算法

陈炳才[1,2],徐东升[1],顾国昌[1],郭黎利[2]
(1. 哈尔滨工程大学 计算机科学与技术学院,哈尔滨 150001; 2. 哈尔滨工程大学 信息与通信工程学院,哈尔滨 150001)
(pinch_chen@126.com)

摘 要:针对现有几种基于二进制搜索法的射频识别(RFID)防冲突算法没有同时考虑识别次数和传输位数这两方面性能,通过改进读写器识别电子标签过程中的步骤和减少读写器发送指令的位数,提出了基于堆栈的 RFID 动态减位防冲突算法。该算法利用堆栈记忆存储功能避免每次从树型根部识别,从而减少识别次数;并采用适当协议进一步缩减必需的指令信息位。仿真结果表明该算法在识别次数和传输位数方面相比其他几种二进制搜索法都有很大降低,在次数效率和位数效率性能上亦有较大提高,故该算法在识别完成时间和能量消耗方面相应得到较大改善。

关键词:RFID 系统;防冲突;堆栈;二进制搜索法;动态减位
中图分类号: TP393.04 **文献标志码**:A

New anti-collision algorithm for RFID system based on stack storage

CHEN Bing-cai[1,2], XU Dong-sheng[1], GU Guo-chang[1], GUO Li-li[2]
(1. *College of Computer Science and Technology, Harbin Engineering University, Harbin Heilongjiang* 150001, *China*;
2. *College of Information and Communication Engineering, Harbin Engineering University, Harbin Heilongjiang* 150001, *China*)

Abstract: To solve the problem that both identifying times and the number of transmitting data could not be simultaneously run in some binary-search schemes, a new anti-collision algorithm for Radio Frequency Identification (RFID) system based on stack storage was proposed through modifying the process for reader to identify all tags and cutting short the number of bits sent by reader. In order to reduce identifying times, this algorithm utilized the memory function of stack to avoid starting identifying step from the root every time. In addition, it adopted suitable protocols to further cut down necessary command bits. Simulation results indicate that this algorithm can not only decrease the identifying times but also reduce the number of command bits. Moreover, it can obtain a better performance with regard to times-efficiency and bits-efficiency. Thus, this algorithm achieves an improvement in the aspects of operative duration and energy consumed.

Key words: Radio Frequency Identification (RFID) system; anti-collision; stack; binary-search scheme; dynamic reducing bits

0 引言

一般来讲,在同一个射频识别(Radio Frequency Identification, RFID)系统中所有的电子标签都工作在相同的频段。因此,如果读写器的作用范围内存在多个电子标签,且在同一时刻有多个电子标签发送信息到达读写器,就会出现信息的互相干扰,使得读写器不能正确识别电子标签的信息,于是就产生 RFID 电子标签冲突。

防冲突算法的研究是要解决如何快速和准确地从多张电子标签中选出一张标签与读写器进行数据交互,而其他未被选中的电子标签则在后续防冲突循环中陆续被选出与读写器进行信息交互的问题[1-2]。现在国内外研究的热点防冲突算法主要有 ALOHA 法和二进制搜索法,这两种方法都属于时分多路法(TDMA)[2-3]。基于二进制搜索法产生了动态二进制搜索法,后退式索引的二进制树形搜索算法,修剪枝二进制树形搜索算法等改进算法[4-6]。

本文研究基于二进制搜索的防冲突算法,并在动态二进制搜索法和后退式索引的二进制树形搜索法的基础上提出了基于堆栈的 RFID 动态减位防冲突算法。

1 现有二进制搜索法的特点和不足

基于二进制搜索的防冲突算法,其大体思想是通过读写器发送一个命令给所有标签,符合要求的标签(例如 ID 号小于读写器发送值的标签)都返回给读写器一定的信息,读写器根据所获得的信息判断冲突与否,然后决定下一步应该做什么工作,是识别某个确定的标签还是继续发送识别指令。

实际上 RFID 中防冲突算法的实施就是通过读写器与电子标签之间的通信,而衡量一个防冲突算法的好坏往往也是从这个通信过程入手。首先是读写器的发送指令的次数,即识别所有标签所需要读写器进行传输数据的次数,识别次数的减少意味着电能的消耗减少,可以增加读写器和标签的使用寿命,同时发送信息次数减少使相应的识别时间也缩短即一次识别过程加快了;其次是传输的位数,即读写器发送至电子标签的总的数据量,总的位数影响传输的耗电量及效率;还有识别时间,即识别出所有标签所用的时间,它是识别工作的关键要求,识别的速度越快越好,通过减少发送次数和发送位

收稿日期:2009 - 01 - 04;**修回日期**:2009 - 03 - 01。
基金项目:黑龙江省博士后基金资助项目(LBH-Z07218);哈尔滨工程大学基础研究基金资助项目(HEUFT07025)。
作者简介:陈炳才(1976 -),男,福建漳平人,讲师,博士,主要研究方向:无线通信与网络; 徐东升(1983 -),男,黑龙江哈尔滨人,硕士研究生,主要研究方向:RFID 系统; 顾国昌(1945 -),男,上海人,教授,博士,CCF 会员,主要研究方向:嵌入式系统; 郭黎利(1955 -),男,黑龙江哈尔滨人,教授,博士,主要研究方向:数字通信、扩频通信。

数就可以减少识别时间,提高识别速度。

动态二进制算法相对于基本二进制算法的优点在于其读写器每次发送的位数要比后者少很多,每次的传输量大概能减少到基本二进制算法的50%左右,进而减少了传输时间和电能的消耗。后退式二进制算法的优点在于它对传输次数上做的改进,它通过保留以往几次的传输命令,可使读写器在识别出一个标签后不需再从初始的最大标签ID号开始发送,而只需读取存储的命令来进行发送,这样使得读写次数将大大降低,由基本二进制算法的$N+\mathrm{lb}(N!)$,降低为$2\times N$(N为标签数)。

通过分析以上几种二进制搜索法,看到它们都存在着一些不足,例如动态二进制算法的传输次数过多,而且在传输位数上也值得改进;后退式二进制算法则存在传输位数过多的缺点,等等。正是在分析上述二进制搜索法优缺点的基础上,提出了基于堆栈的动态减位防冲突算法。

2 RDS算法思想

基于堆栈的动态减位防冲突算法(Reduced Dynamic binary algorithm based on Stack,RDS)的基本思想如下:

1)在动态二进制防冲突算法[4]的识别过程中,读写器进行防冲突识别通信过程的每一次指令发送只需要发送指令码的前段数据,ID号的前部分与该指令码相符的电子标签则对应发回ID号的后段数据。因为通信双方发送的数据只是电子标签ID号的一部分,而非全部,所以达到了数据传输量减少的作用,提高了发送效率。实际上在发送过程中,数据量的减少还可以进一步提高,因为根据动态二进制搜索法,在发生冲突的时候,检测到发生冲突的最高位,根据其冲突的原理知道,应该把此最高位置0,这样可以除去一半左右的数据免于发送;同时,在置0的时候,实际上此最低位是不必发送的,因为读写器与电子标签双方可以隐含地约定此位为0而免于发送。发送过程中发送次数是很多次,故每减少一位就能减少很多位,对于整体数据量的减少是相当大的。

2)对于后退式二进制算法[5],它的传输次数虽然减少了,但传输位数并没有减少,于是可以利用它的传输次数减少的思想,综合对动态二进制算法的改进,进而产生了基于堆栈的动态减位防冲突算法。此算法在传输次数以及传输位数上相对于现有的几种基于二进制搜索的防冲突算法有很大的改进,并且传输时间上大大地减少了,可以说是一种较优化的算法。

3 RDS算法指令

RDS所使用的指令集和其他二进制算法类似,不过它们的参数和功能则不尽相同,且在此算法中增加了PUSH和POP这两个应用于堆栈的指令,它的指令集如下。

1)REQUEST:请求命令(或缩写为REQ)。

命令形式:REQUEST(ID),REQUEST(Part ID)。它的参数可以是一个位数和ID长度相同的二进制数,也可以短于ID长度。根据长度不同电子标签返回值的含义也就不同。读写器通过该命令将ID号或部分ID号作为参数发送给电子标签。根据事先约定好的协议,符合该指令要求的标签要么发回整个ID号给读写器,要么发回部分ID号。不符合要求的标签则对此命令不作响应。通过该指令可以缩小标签的范围。

2)SELECT:选择命令(或缩写为SEL)。

命令形式:SELECT(ID)。它的参数是电子标签的ID号。它通过传输某个需要识别的标签ID来识别该标签。ID与该值相同的标签做出响应,以此作为执行读取或写入的开关。对其他标签不做响应。

3)READ:读取命令。

命令形式:READ(ID)。它的参数是电子标签的ID号。需要读取数据的电子标签首先需要通过SELECT命令加以选择,当再次收到READ命令时,该标签将自身存储的数据等信息发送给读写器。实际上在存储数据比较少的时候,例如只存储标签ID的时候,SELECT命令和READ命令可以合为一个命令来读取标签ID。

4)UNSELECT:去选择命令(或缩写为UN)。

命令形式:UNSELECT(ID)。它的参数是电子标签的ID号。该命令的发送是在READ命令读取数据之后,读写器发送给电子标签,已经发完数据的标签在收到此命令后,进入"休眠"状态。这种状态代表此标签已经识别完毕,无需再识别。在这种状态下,电子标签是非激活的,对下一轮发送的REQUEST命令不作响应。如果还要激活此标签,则需要将它移出读写器的作用范围,然后再移入,进行新一轮的防冲突读取操作。

5)PUSH:存储参数指令。

命令形式:PUSH(ID),PUSH(PartID)。此命令是用于当完成一次REQUEST之后,产生了冲突,这时应该把产生冲突的命令参数存储起来,以备下次REQUEST时使用。

6)POP:读取参数指令。

命令形式:POP()。此命令是用于在完成一个标签的识别之后,下次REQUEST需要发送的参数要通过POP指令传输过来,约定当堆栈为空时POP()命令返回的是栈底数据。

4 RDS算法流程

RDS算法的参数名称定义如下。

preID:代表读写器发送命令REQUEST时的参数,通常代表待识别标签ID号的前段数值。

partID:代表电子标签返回给读写器的数据,通常代表待识别标签ID号的后段数值。

tempID:用于临时存储新产生的preID值,待旧preID值压栈存储之后,将tempID赋给preID。

下面通过一个具体的实例来说明RDS算法操作的具体过程,为简单起见定义标签的ID位数为4位,共4个标签,分别是:1110,1100,1010,1000。其识别过程如图1所示。

首先读写器发送命令REQUEST(1111),ID小于1111的电子标签返回其ID号,读写器接收到的结果是1xx0,则下一次发送的参数为1,保存此次发送的参数1111。

发送命令REQUEST(1),ID前两位是10的电子标签返回其后半部分ID为10、00。读写器接收到的结果为x0,则下一次发送的参数为10,保存此次发送的参数1。

发送命令REQUEST(10),ID前三位是100的电子标签返回其后半部分ID为0,读写器接收到的结果为0,没有产生冲突,则读写器对ID为1000的电子标签进行数据读取和去选择,之后通过POP命令产生下一次的REQUEST参数1。

发送命令REQUEST(1),ID前两位是10的电子标签返回其后半部分ID为10,读写器接收到的结果为10,没有产生冲突,则读写器对ID为1010的电子标签进行数据读取和去选择,之后通过POP命令产生下一次的REQUEST参数1111。

发送命令 REQUEST(1111),ID 小于 1111 的电子标签返回其 ID 号为 1110、1100,读写器接收到的结果是 11x0,则下一次发送的参数为 11,保存此次发送的参数 1111。

发送命令 REQUEST(11),ID 前三位是 110 的电子标签返回其后半部分 ID 为 0,读写器接收到的结果为 0,没有产生冲突,则读写器对 ID 为 1100 的电子标签进行数据读取和去选择,之后通过 POP 命令产生下一次的 REQUEST 参数 1111。

发送命令 REQUEST(1111),ID 小于 1111 的电子标签返回其 ID 号为 1110,读写器接收到的结果是 1110,没有产生冲突,则读写器对 ID 为 1110 的电子标签进行数据读取和去选择,之后通过 POP 命令产生下一次的 REQUEST 参数 1111。

发送命令 REQUEST(1111),ID 小于 1111 的电子标签返回其 ID 号。因为已经没有需要识别的电子标签,所以没有数据进行返回,则一次完整的识别过程结束。

读写器 --->	REQ(1111)	REQ(1)	REQ(10)	SEL(1000) UN(1000)	REQ(1)	SEL(1010) UN(1010)	REQ(1111)	REQ(11)	SEL(1100) UN(1100)	REQ(1111)	SEL(1110) UN(1110)	REQ(1111)
标签 <---	1xx0	x0	0		10		11x0	0		1110		NULL
Tag(1110)	1110						1110			1110		
Tag(1100)	1100						1100	0				
Tag(1010)	1010	10			10							
Tag(1000)	1000	00	0									
堆栈操作	PUSH(1111)	PUSH(1)		POP()		POP()	PUSH(1111)		POP()		POP()	结束

图 1　RDS 算法识别过程

5　仿真分析

5.1　识别次数及效率分析

首先对传输的总次数作分析。令 T 代表识别的总次数,T_B 代表基本二进制算法的识别总次数,T_{DB} 代表动态二进制算法的识别总次数,T_{BA} 代表后退式二进制算法的识别总次数,T_{RDS} 代表 RDS 算法的识别总次数,N 表示电子标签的总数量。

下面将分别对传输总次数和次数传输效率进行仿真分析。本仿真利用作者基于VC++开发的仿真平台来实现读写器与电子标签的收发过程,通过大量的数据来仿真读写过程。分别对 $K=8$,$K=16$,$K=32$ 作标签个数与识别总次数的对应图,结果如图 2 所示。通过图 2 可以发现基本二进制和动态二进制的识别总次数几乎重合,后退式二进制与 RDS 算法的识别总次数基本一致,而且无论标签的位数是多少,识别的总次数相差无几,说明总次数只和标签的个数有关。并且从图中可以看出 RDS 算法的识别总次数要远远低于基本二进制算法和动态二进制算法,为了定量分析 RDS 算法的识别次数的优点,对 $K=24$ 的数据做定量分析,如表 1 所示。

由表 1 可以看出,RDS 算法次数比基本二进制和动态二进制算法少很多,经过计算 RDS 算法的搜索总次数仅是基本二进制和动态二进制的 42.21%,这是很大的提高。

次数传输效率定义为:

次数传输效率 = 有效的传输次数/总的传输次数

其中有效次数定义为每个电子标签的识别只需一次情况下识别所有标签所需的传输次数。次数传输效率的计算结果,基本二进制算法和动态二进制算法为 21.1%,而 RDS 算法为 50%,提高了近 2.36 倍。

5.2　传输数据量及效率分析

传输总位数(S)是衡量防冲突算法好坏的另一个指标,它与传输总次数与每次传输的位数有关系,通过此参数的比较可以综合比较传输的耗时以及耗电量。令 S 代表识别的总位数,S_B 代表基本二进制算法的识别总位数,S_{DB} 代表动态二进制算法的识别总位数,S_{BA} 代表后退式二进制算法的识别总位数,S_{RDS} 代表 RDS 算法的识别总位数,K 表示电子标签的位数,N 表示电子标签的总数量,令 R 代表每次传输的位数,则各种算法的每次的传输位数分别为 R_B、R_{DB}、R_{BA}、R_{RDS}。

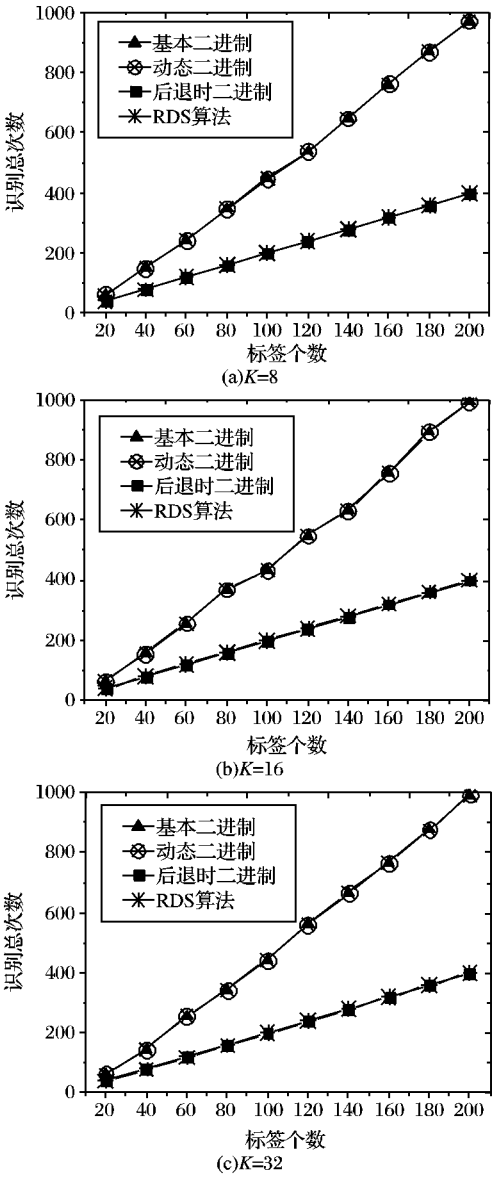

图 2　识别次数比较

下面对传输总位数和位数传输效率进行仿真分析。分别对 $K=8$,$K=16$,$K=32$ 作传输总位数与位数传输效率的对应图,结果如图3所示。通过图3可以发现在传输总位数方面RDS算法的传输位数是最少的(也就是最优的),然后依次是后退式二进制算法,动态二进制,基本二进制算法。这也证实了前面的理论分析结果。从图中可以看出RDS算法的识别总位数要远远低于其他算法,为了定量分析RDS算法的识别位数,对 $K=24$ 的数据做定量分析,如表2所示。

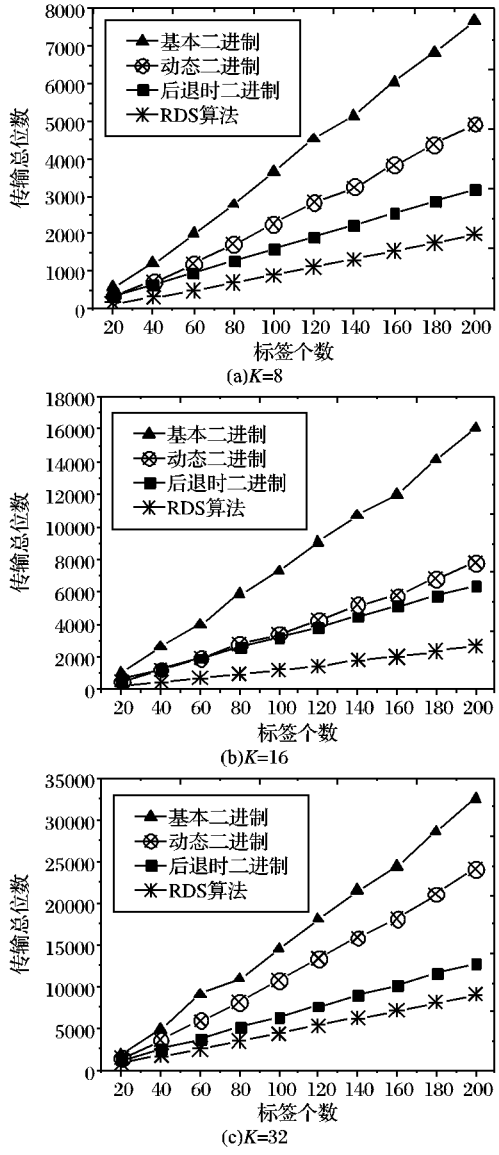

图3 传输数据量比较

由表2可以看出,RDS算法位数比其他几类算法少很多,经过计算RDS算法的识别总位数仅是基本二进制的25.5%,动态二进制的38.9%,后退式二进制的59.4%,有如此大的改进可以说RDS算法远远优于其他算法。

位数传输效率定义为:

位数传输效率=有效位数/总的传输位数

其中有效位数定义为在每个电子标签的识别时只需一次完整ID发送情况下识别所有标签所需的传输位数。位数传输效率的计算结果,RDS算法为84.1%,基本二进制算法为21.4%,动态二进制算法为32.7%,后退式二进制算法为50%。因此在位数传输效率方面RDS算法也优于其他算法。

6 结语

通过分析得知,与基本二进制搜索法、动态二进制搜索法及后退式二进制搜索法相比,当识别的标签数、标签ID位数相同时,本文提出的RDS算法在识别次数方面有了很大减少,在识别过程中所需的传输位数同样小于其他三种算法。另外,在次数传输效率方面,RDS算法比基本二进制搜索法和动态二进制搜索法提高2.36倍;在位数传输效率方面,RDS算法相对于其他三种算法至少提高1.68倍,最高达近4倍。可见,RDS算法有较优的性能。

当然,通过分析也知道,RDS算法对读写器的存储能力、电子标签的信息处理能力的要求也会相应提高,在实际的RFID系统应用中有可能会增加硬件成本。

表1 识别次数部分仿真数据

标签个数	基本二进制	动态二进制	后退二进制	RDS算法
20	62	62	40	40
40	148	148	80	80
60	251	250	120	120
80	369	369	160	160
100	462	462	200	200
120	602	602	240	240
140	655	655	280	280
160	746	746	320	320
180	886	886	360	360
200	1031	1031	400	400

表2 传输数据量部分仿真数据

标签个数	基本二进制	动态二进制	后退二进制	RDS算法
20	1728	1124	960	545
40	3480	2299	1920	1072
60	5736	3753	2880	1620
80	8520	5584	3840	2226
100	10704	6959	4800	2779
120	13992	9086	5760	3446
140	15936	10425	6720	4043
160	18456	12133	7680	4640
180	21120	13824	8640	5189
200	23208	15316	9600	5827

参考文献:

[1] 刘冬生,邹雪城,李泳生,等.射频识别系统中的防碰撞算法[J].华中科技大学学报,2006,34(9):84-86.

[2] EOM J B, LEE T J. An efficient framed-slotted ALOHA algorithm with pilot frame and binary selection for anti-collision of RFID tags [J]. IEEE Communications Letters, 2008, 12(11): 861-8863.

[3] 张颇,崔喆.RFID系统中一种改进的防冲撞算法[J].计算机应用,2008,28(8):1412-3412.

[4] 鞠伟成,俞承芳.一种基于动态二进制的RFID抗冲突算法[J].复旦大学学报,2005,44(1):46-50.

[5] 杜海涛,徐昆良,王威廉.基于返回式二进制树形搜索的反碰撞算法[J].云南大学学报:自然科学版,2006, 28(S1):133-136.

[6] 王亚奇,顾亦然,蒋国平.改进型的二进制搜索RFID系统反碰撞算法[J].计算机应用,2007,27(11):2877-2879.

[7] CHOI J H. Query tree-based reservation for efficient RFID tag anti-collision [J]. IEEE Communications Letters, 2007, 11(1): 85-87.

[8] 韩磊,张虹,马海波.散列树形搜索反碰撞算法的研究[J].计算机应用,2006,26(12):3019-3022.