

RFID 标签双模式阻塞防碰撞算法

赵 宇, 沈海斌*

(浙江大学 超大规模集成电路设计研究所, 浙江 杭州 310027)

摘要:为解决“驻留标签之间碰撞、驻留标签与迁入标签之间碰撞、当驻留率较低时传统阻塞算法产生大量空闲时隙”等问题,提出了一种双模式的阻塞防碰撞算法。首先开展了对自适应二叉树分割算法(ABS)和传统阻塞算法的研究和性能分析。然后在保留上述算法优势的基础上,结合对迁入标签数目的估计算法,解决前面所提到的问题,并从标签数目、驻留率和迁入率等方面进行了算法的性能比较。理论分析和仿真实验都证明了该算法的有效性和可靠性。

关键词:射频识别;自适应二叉树分割原法;阻塞;防碰撞

中图分类号:TP301

文献标志码:A

文章编号:1001-4551(2011)07-0867-05

Double-mode blocking anti-collision algorithm in RFID tags

ZHAO Yu, SHEN Hai-bin

(Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China)

Abstract: In order to solve the problems of the collisions between staying tags, those between arriving tags and staying tags, and the large amount of idle time slots brought by traditional blocking algorithms when staying ratio is low, a new ‘double-mode’ blocking algorithm was proposed. Research and performance analysis about adaptive binary splitting(ABS) algorithm and traditional blocking algorithms were carried out. Considering the advantages of the algorithms above and the number of arriving tags, the problems mentioned above were solved in this algorithm. Performance analyses were done from the aspects of the number of total tags, the arriving ratio and the staying ratio. Theory analysis and experiments both prove the efficiency and reliability of this algorithm.

Key words: radio frequency identification(RFID); adaptive binary splitting(ABS); blocking; anti-collision

0 引 言

射频识别(RFID)技术在自动识别系统中应用越来越广泛^[1]。通常,RFID系统由一个读写器和一些标签组成,每个标签携带特有的识别序列号(ID)。当标签进入读写器的识别范围时,读写器可对其进行识别。然而,由于信道共享,当多于一个的标签同时传输其ID时,便产生碰撞^[2]。

标签防碰撞算法则用以解决上述问题。目前大多通信系统采用时分复用的方法,使某一个标签在某个时隙单独占用信道与读写器进行通信^[3]。具体来说可归纳为两类:一类是基于Aloha的算法,如动态时隙

Aloha(DSA),分群时隙Aloha(GSA)等,由于具有一定的随机性,存在“tag starvation”的问题,又称为概率性算法;另一类是基于树的算法,如二叉树算法(BT),查询树算法(QT)等,由于不存在上述问题,又称为确定性算法^[4]。

在很多RFID系统中,读写器可能会对某些一直存在于其识别范围内的标签进行重复识别,这种标签称为驻留标签。驻留标签数占总识别标签数的比率称为驻留率 p_s 。Myung^[5-7]提出的自适应二叉树分割算法(ABS)和自适应二叉树查询算法(AQS)即要解决驻留标签之间发生的碰撞问题。但这些算法没有解决驻留标签和迁入标签之间的碰撞。Yuan-Cheng Lai 和

收稿日期:2010-12-17

作者简介:赵 宇(1986-),女,山东临沂人,主要从事信息安全相关的集成电路设计方面的研究. E-mail: zhaoyu@vlsi.zju.edu.cn

通信联系人:沈海斌,男,博士,副教授. E-mail: shb@vlsi.zju.edu.cn

Chih-Chung Lin^[8]提出单解决通道阻塞算法(SRB)是在 ABS 算法的基础上,解决驻留标签和迁入标签的碰撞问题。当 p_s 较低时,SRB 会产生大量空闲时隙,性能较 ABS 低平。

本研究提出一种双模式的阻塞算法,结合对于迁入标签数目的估计算法,能够同时保留 ABS 与 SRB 算法的优势,并且具有较高的性能。当 p_s 较高时,采用阻塞算法,解决驻留标签和新迁入的标签之间的碰撞;在 p_s 较低时,采用类似 ABS 的算法,解决空闲时隙问题。驻留标签将根据写入的状态位决定是否进入 sleep 状态。

1 ABS 算法分析

ABS 算法是由 Jihoon Myung, Wonjun Lee 等提出的,该算法由 BT 算法改进而来,能够保留标签在上一帧获得的识别顺序,从而避免驻留标签在当前帧的碰撞。每个标签具有两个计数器:PSC 和 ASC。PSC 记录当前帧已识别的标签数目,ASC 记录标签当前帧的识别次序。读写器也有两个计数器:PSC 和 TSC。PSC 的功能与标签的相同,TSC 追踪最大的 ASC 值,并据以判断结束当前帧。

帧开始时,所有的 PSC 值初始化为零,标签和读写器的 PSC 值始终保持一致。当标签检测到其 ASC 与 PSC 值相同时,向读写器传输 ID。读写器根据在上一时隙的识别状况,给出 3 种反馈信息,标签根据反馈信息进行相应操作:

(1) 空闲:即没有标签需要识别。若 $ASC > PSC$,则表明尚未进行识别,则 $ASC = ASC - 1$,若 $ASC = PSC$,传输 ID,若 $ASC < PSC$,退出当前帧;

(2) 可读:即只有一个标签需要识别。 $ASC = PSC$ 的标签传输 ID,若 $ASC < PSC$,退出当前帧。 $PSC = PSC + 1$;

(3) 碰撞:即多于一个标签需要识别。若 $ASC > PSC$,则 $ASC = ASC + 1$,若 $ASC = PSC$,即碰撞标签,则 ASC 加上一个随机产生的二进制数,若 $ASC < PSC$,退出当前帧。

TSC 与 PSC 的关系是读写器判断帧结束的依据。当反馈为空闲时, $TSC = TSC - 1$;碰撞时, $TSC = TSC + 1$ 。直到 $TSC < PSC$ 时,结束当前帧。

现假设有 A、B、C 共 3 个标签需要识别。则开始时,所有的计数器值都为 0。其识别过程如表 1 所示,其中带下划线的标签表示其 $ASC = PSC$,并且将要识别。

表 1 第 i 帧识别过程

时隙数	反馈信息	TSC	PSC	标签:ASC		
				A	B	C
0	开始	0	0	<u>0</u>	<u>0</u>	<u>0</u>
1	碰撞	1	0	<u>0</u>	1	1
2	可读	1	1	0	<u>1</u>	<u>1</u>
3	碰撞	2	1	0	2	2
4	空闲	1	1	0	<u>1</u>	<u>1</u>
5	碰撞	2	1	0	<u>1</u>	2
6	可读	2	2	0	1	<u>2</u>
7	可读	2	3	0	1	2

当下一帧开始时,标签将保留上一帧得到的 ASC 值,读写器保留 TSC 的值。PSC 初始化为 0。由于每个标签的 ASC 值都是惟一的,不存在驻留标签之间的碰撞问题。对于新迁入的标签,其 ASC 值在 0 ~ TSC 之间随机产生。

假设当前帧识别完毕后,A、B、C 分别具有 0,1,2 的 ASC 值。若在下一帧开始时,标签 A 离开,B、C 驻留,同时有新的标签 D、E 迁入,且随机产生的 ASC 分别为 1 和 2,则其识别过程如表 2 所示。

表 2 第 $i+1$ 帧识别过程

时隙数	反馈信息	TSC	PSC	标签:ASC			
				A	B	C	D
0	开始	3	0	1	2	1	2
1	空闲	2	0	<u>0</u>	1	<u>0</u>	1
2	碰撞	3	0	<u>0</u>	2	1	2
3	可读	3	1	0	2	<u>1</u>	2
4	可读	3	2	0	<u>2</u>	1	<u>2</u>
5	碰撞	4	2	0	<u>2</u>	1	3
6	可读	4	3	0	2	1	<u>3</u>
7	可读	4	4	0	2	1	3
8	空闲	3	4	0	2	1	3

在上述识别过程之后,B、C、D、E 分别获得 0、2、1、3 的识别顺序。

2 基于自适应二叉树分割的阻塞防碰撞算法

通过上述分析可知,ABS 算法能够解决驻留标签之间的碰撞。然而,由于新迁入标签的 ASC 值是在 0 ~ TSC 之间随机产生,则可能会与驻留标签产生碰撞。另外,由于具体的应用需求,不是所有的驻留标签在下一帧都需要再次进行识别。

文献[8]中提出的 SRB 算法针对第一个问题提出了解决方案。根据文献[9]与文献[10]所得的结论,基于二叉树的算法最优帧大小为 0.88 n,由此产生一个新的值 TSCEXT,即在 TSC 的基础上,增加一个对于迁入标签数的估计值。与 ABS 相对应的,迁入标签其 ASC 计数器会在 TSC + 1 ~ TSCEXT 之间随机产生,则

无论驻留标签数目多少,都将占用 TSC 个时隙,而迁入标签则服从 BT 算法。

但由此会产生一个新的问题。若标签数目很大,驻留标签较少,将会产生大量空闲时隙,但是 ABS 算法却没有这个问题。

由此,本研究改进性地提出双模式阻塞算法(DMB),根据驻留率 p_s 的大小适时调整策略,能够保证解决驻留标签的碰撞和阻塞算法空闲时隙的问题。

该算法读写器和标签具体操作流程的伪代码如图1、图2所示。其中变量 $count$ 为在该帧内新迁入的标签计数器, $stay$ 为在该帧内驻留标签的计数器。 z 为上一帧估算的标签数和上一帧新迁入的标签数之间的一个权重,并且该值对于数目估计影响不大^[8]。 $needstore$ 为标签是否需要同一个读写器进行重复识别的状态位, rID 为读写器发送的 ID 号, tID 为标签记录的读写器 ID 号。

读写器操作:

```

1  传输帧开始命令
2   $PSC = 0$  //PSC 为已识别标签数目的计数器
3  if  $TSC = NULL$  or  $TSC < 0$  then
4     $TSC = 0$  //最终最大 ASC 值
5     $count = 0$  //新迁入标签数目计数器
6     $stay = 0$  //驻留标签数目计数器
7  endif
8   $Est = z \times Est + (1 - z) \times count$  //迁入标签数目估计(在上一帧迁入数目与历史估计数目间均衡)
9   $TSCEXT = TSC + \lfloor 0.88 \times Est \rfloor$ 
10  $count = 0$    $stay = 0$ 
11  $TSC = TSCEXT$ 
12 传输  $TSC, TSCEXT, rID, MODE$  等信号到标签
13 while  $PSC \leq TSC$  do
14   接收标签响应信号并检测做出判断
15   if 碰撞 then
16      $TSC = TSC + 1$  //与最大 ASC 值同步
17      $response = collision$  //给出响应类型
18   else if 可读 then
19     if  $Newtag()$  then
20        $count = count + 1$ 
21     else if  $Staytag()$  then
22        $stay = stay + 1$ 
23     end if
24     存储标签 ID
25      $PSC = PSC + 1$  //在该时隙识别一个标签
26      $response = readable$ 
27   else if 空闲 then
28      $TSC = TSC - 1$  //与最大 ASC 值同步
29      $response = idle$ 
30   end if
31 end while
32  $p_s = w \times p_s + (1 - w) \times stay / (stay + count)$ 
   //计算驻留率,并在历史驻留率与本帧驻留率间均衡
33 if  $p_s < Threshold$  then //与阈值进行比较进行模式选择
34    $MODE = lowstayratio$ 

```

```

35 else then
36    $MODE = highstayratio$ 
37 end if
38 传输响应信号  $response$ 
38 结束帧并传输帧结束命令

```

图1 读写器操作伪代码

标签操作:

```

1 接收命令开始帧并接收  $TSC, TSCEXT, rID, MODE$  等信号
2  $PSC = 0$  //PSC 为已识别标签数目的计数器,与读写器同步
3 if  $tID \neq rID$  or  $tID = NULL$  then
4    $tID = rID$  //根据读写器 ID 判断是否驻留标签
5   if  $MODE = lowstayratio$  then //判断模式
6     ASC 为随机生成  $0 \sim TSCEXT$  之间的整数 //与 ABS 类似的算法
7   else then
8     ASC 为随机生成  $TSC + 1 \sim TSCEXT$  之间的整数 //阻塞算法
9   end if
10 else if !  $needstore$  then
11   标签进入 sleep 状态 //标签自主退出识别
12 end if
13 while  $PSC \leq ASC$  do
14   if  $PSC = ASC$  then
15     传输标签 ID
16     接收读写器返回  $response$ 
17     if  $response = collision$  then
18       ASC 加上一个随机生成的二进制数  $i$  //  $i$  为随机的 1 或 0
19     else
20        $PSC = PSC + 1$  //成功识别一个标签
21     end if
22   else if  $PSC < ASC$  then
23     接收读写器返回  $response$ 
24     if  $response = collision$  then
25        $ASC = ASC + 1$  //类似二叉树算法思路,使标签所在的二叉树深度加 1
26     else if  $response = readable$  then
27        $PSC = PSC + 1$ 
28     else if  $response = idle$  then
29        $ASC = ASC - 1$  //类似二叉树算法思路,使标签所在的二叉树深度减 1
27   end if
28 end while

```

图2 标签操作伪代码

3 性能分析与仿真

本研究假设在读写器识别区域内有 n 个标签待识别,识别延时用 T_i 表示, i 代表第 i 帧, $|T_i| = n$ 即在第 i 帧, T_i 时间内识别标签数为 n 。

不论是 ABS, AQS, SRB, 还是本研究提出的 DMB 算法,都源于二叉树 BT 算法。由文献[5]可知,BT 算法识别 n 个标签所需的时隙数为:

$$D_{BT}(T_i) = 1 + \sum_{k=0}^{\infty} 2^{k+1} \left\{ 1 - \left(1 - \frac{1}{2^k}\right)^n - n \frac{1}{2^k} \left(1 - \frac{1}{2^k}\right)^{n-1} \right\} \quad (1)$$

当第 i 帧识别 n 个标签后,在第 $i+1$ 帧,假设有 a 个标签迁入, l 个标签迁出,则有 $n-l$ 个标签驻留。由文献[8]可知,第 $i+1$ 帧的所需时隙数为:

$$D_{ABS}(T_{i+1} | T_i) = (n-l) \sum_{x=0}^a C_a^x \left(\frac{1}{n}\right)^x \left(1 - \frac{1}{n}\right)^{a-x} D_{BT}(1+x) + l \sum_{x=0}^a C_a^x \left(\frac{1}{n}\right)^x \left(1 - \frac{1}{n}\right)^{a-x} D_{BT}(x) \quad (2)$$

当 p_s 较高时,由于 DMB 是阻塞算法,不论是退出识别的标签还是迁出标签,均对应第 $i+1$ 帧空闲时隙,而驻留标签则占用可读时隙,因此仍需要 n 个时隙。该算法假设文献[10]给出迁入标签的最优估计 $|0.88 \hat{a}|$ 在理论上是正确的,因此得到该算法在该模式下的所需时隙数位:

$$D_{DMB}(T_{i+1} | T_i) = n + |0.88 \hat{a}| \sum_{x=0}^a C_a^x \left(\frac{1}{|0.88 \hat{a}|}\right)^x \left(1 - \frac{1}{|0.88 \hat{a}|}\right)^{a-x} D_{BT}(x) \quad (3)$$

当 p_s 较低时,驻留标签仍保留上一帧识别序列,因此不存在驻留标签之间的碰撞问题。本研究估计迁入标签数为 $\hat{a}$,因此若迁入标签 ASC 值分布在 $0 \sim TSCEXT$ 之间,则可能发生驻留标签与迁入标签的碰撞。对于某一个驻留标签而言,若 a 个标签中有 x 个选择该 ASC 值,根据二项分布,则发生碰撞的标签概率值为:

$$C_a^x \left(\frac{1}{n+\hat{a}}\right)^x \left(1 - \frac{1}{n+\hat{a}}\right)^{a-x} \quad (4)$$

则其总识别延时为:

$$\sum_{x=0}^a C_a^x \left(\frac{1}{n+\hat{a}}\right)^x \left(1 - \frac{1}{n+\hat{a}}\right)^{a-x} D_{BT}(1+x) \quad (5)$$

由此可以得到,在该模式下,DMB 算法所需时隙数为:

$$D_{DBM} = (n-l) \sum_{x=0}^a C_a^x \left(\frac{1}{n+\hat{a}}\right)^x \left(1 - \frac{1}{n+\hat{a}}\right)^{a-x} D_{BT}(1+x) + (l+\hat{a}) \sum_{x=0}^a C_a^x \left(\frac{1}{n+\hat{a}}\right)^x \left(1 - \frac{1}{n+\hat{a}}\right)^{a-x} D_{BT}(x) \quad (6)$$

现假设共有 $N=100$ 个标签, $|T_i|=50$, $p_a, p_s=0.5$ 分别为迁入标签和驻留标签的概率。本研究先考虑 N 对各算法的影响。

当 N 增大时,驻留标签数增大,迁入和迁出标签数目也随之增加,因此其总识别所需时隙数呈线性增加如图 3 所示,由图 3 可以看出, N 对 DMB 算法的影

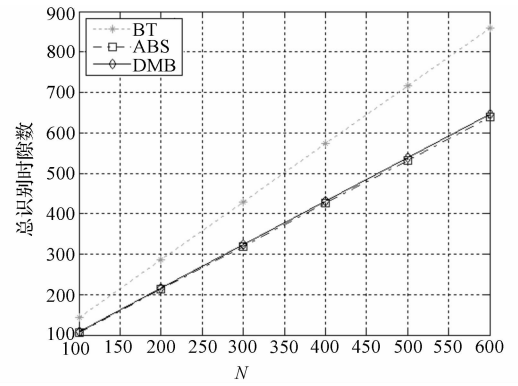


图 3 标签数 N 对算法的影响

响与 ABS 算法相似。

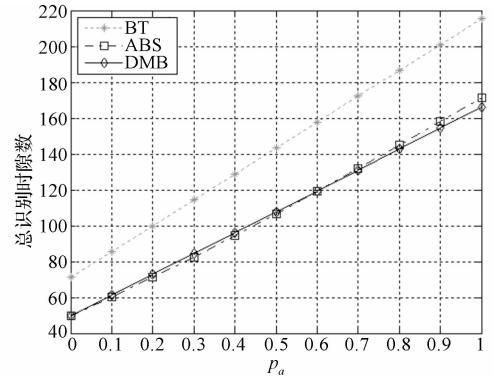


图 4 迁入率 p_a 对算法的影响

当驻留率 p_s 确定时,随着迁入标签数目的增加,DMB 算法比 ABS 算法的空闲时隙数要多,碰撞时隙数少,但总时隙数相差不大,在 $p_s=0.5$ 的基础上得到的实验结果如图 4 所示。

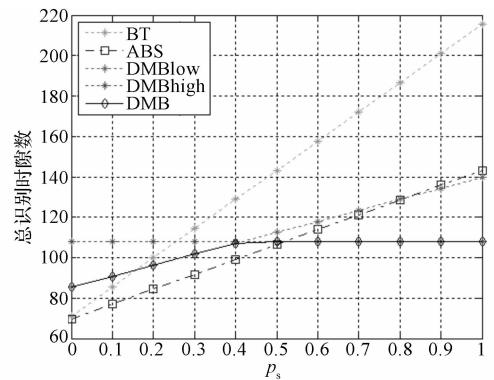


图 5 驻留率 p_s 对算法的影响

DMB 算法最重要的任务就是解决 p_s 对延时的影响,实际结果如图 5 所示,由图 5 可以看出,在 $p_s < threshold$ 时,曲线为 DMBlow 的部分, $p_s > threshold$ 时,曲线为 DMBhigh 的部分, $threshold$ 大约在 0.41 左右,但需要进一步的理论证明。其中 DMBlow 与 DMBhigh 分别为上述两种模式对 p_s 的变化曲线。

4 结束语

本研究提出一种双模式的阻塞防碰撞算法 (DMB),在 p_s 较高时,采用阻塞思想,能够有效避免驻留标签之间的碰撞,也能解决驻留标签与迁入标签之间的碰撞问题;在 p_s 较低时,解决了传统阻塞算法大量空闲时隙的问题。因此该算法相对于 ABS 和传统阻塞算法来说,性能显著得到了提高,这可以从上面的数学分析和仿真图得到。对于使用的文献[10]给出的标签估计,本研究假设该理论是正确的,对于估计错误所带来的影响,主要在于 $\hat{a}$ 的估计值与实际迁入标签数目 a 之间的偏差,但是也不会太大。对于实际偏差的大小和对于 *threshold* 最优值估计,目前也还没有理论的证明,这都是未来进一步研究的内容。

参考文献 (References):

[1] 叶吉波,黄言态. RFID 技术在图书管理中的应用及实现[J]. 机电工程,2009,26(4):104-106.

[2] 丁治国. RFID 关键技术研究及实现[D]. 合肥:中国科学技术大学信息学院,2009:14-55.

[3] 向垂益. RFID 二进制树防碰撞算法的研究与实现[D]. 长沙:湖南大学电气与信息工程学院,2009:3-6.

[4] 李世煜. 射频识别(RFID)系统防碰撞算法研究与设计[D]. 成都:西南交通大学信息科学与技术学院,2008:13-16.

[5] MYUNG J, LEE W, SRIVASTAVA J. Adaptive binary splitting for efficient RFID tag anti-collision[J]. **IEEE Communications Letters**,2006(10):144-146.

[6] MYUNG J, LEE W, SRIVASTAVA J. Tag-Splitting: adaptive collision arbitration protocols for RFID tag identification[J]. **IEEE Transactions on Parallel and Distributed Systems**, 2007,18(6):763-775.

[7] MYUNG J, LEE W. Adaptive binary splitting: a RFID tag collision arbitration protocol for tag identification[J]. **Mobile Networks and Applications**,2006(11):711-722.

[8] LAI Y C, LIN C C. Two blocking algorithms on adaptive binary splitting: single and pair resolutions for RFID tag identification[J]. **IEEE/ACM Transactions on Networking**, 2009,17(3):962-975.

[9] PARK J, CHUNG M Y, LEE T. Identification of RFID tags in framed-slotted ALOHA with robust estimation and binary selection[J]. **IEEE Communications Letters**, 2007, 11(5):452-454.

[10] EOM J, LEE T J. Framed-slotted ALOHA with estimation by pilot frame and identification by binary selection for RFID anti-collision[C]//International Symposium on Communications and Information Technologies (ISCIT 2007). Sydney:[s. n.],2007:1027-1031.

[编辑:李 辉]

(上接第 862 页)

参考文献 (References):

[1] 张小飞. 嵌入式 Web 服务器在工业现场中的应用[J]. 中国高新技术企业,2010(25):86-87.

[2] 汤碧玉,曾楠,郑灵翔,等. 嵌入式系统中基于 Web 的远程监控设计与实现[J]. 厦门大学学报:自然科学版,2004,43(5):632-635.

[3] 杜 鵬. 基于嵌入式 Web 服务器的远程监控系统的研究[D]. 西安:西安电子科技大学技术物理学院,2008.

[4] RASKOVIC D, REVURI V, GIESSEL D, et al. Embedded Web Server for Wireless Sensor Networks, System Theory (SSSD) [C]//41st Southeastern Symposium on Tullahoma, TN, USA:[s. n.],2009:19-23.

[5] FERNANDEZ R O, PERES R. Remote Monitoring and Control of Electronic Nutrunners with Embedded Internet [C]//

IEEE 2002 28th Annual Conference of Industrial Electronics Society, 2002:2371-2375.

[6] 叶 欣,李世伦. 基于 Web 的单晶生长炉远程监控系统[J]. 机电工程,2010,27(3):49-52.

[7] 叶宏材,陈峙桐. Windows CE. NET 嵌入式工业用控制器及自动控制系统设计[M]. 北京:清华大学出版社,2005.

[8] LWA K L K, LEUNG R. A Design and implementation of active network socket programming[J]. **Microprocessors and Microsystems**,2003,27(5):277-284.

[9] 贾东耀,彭树林. 网络控制器 DM9000A 在嵌入式系统中的应用[J]. 电子产品世界,2008(8):94-96.

[10] 王田苗,魏洪兴. 嵌入式系统设计与实例开发[M]. 3 版. 北京:清华大学出版社,2008.

[编辑:李 辉]