

DOI: 10.13382/j.jemi.2017.08.017

改进的混合查寻树 RFID 防碰撞算法*

史露强¹ 何怡刚¹ 罗旗舞¹ 李兵¹ 邓芳明^{1,2}

(1. 合肥工业大学电气与自动化工程学院 合肥 230009; 2. 华东交通大学电气与自动化工程学院 南昌 330013)

摘要:无线射频识别(RFID)系统采用的防碰撞算法直接决定了系统的工作效率。在查寻树(QT)算法的基础上提出了一种改进的混合查询树(IHQT)标签防碰撞算法。标签产生碰撞时,碰撞节点会裂分为多个子节点。为了避免空闲时隙的产生,IHQT算法在阅读器查询碰撞标签之前增加一个分支预测阶段,提出的分支预测方法可以准确地预测查询树中空闲时隙的位置,阅读器产生新的查询前缀时,不产生访问空闲时隙的查询前缀,从而完全避免了空闲时隙的产生。算法的性能分析和仿真结果表明,IHQT算法在阅读器开销少量增加(分支预测位 $n \leq 2$)甚至明显减少(分支预测位 $n \geq 3$)的前提下,时隙数、吞吐率有了显著的改进,优于已有的查寻树防碰撞算法。

关键词:无线射频识别;防碰撞算法;查寻树;分支预测

中图分类号: TP312 **文献标识码:** A **国家标准学科分类代码:** 520.1040

Improved hybrid query tree anti-collision algorithm of RFID

Shi Luqiang¹ He Yigang¹ Luo Qiwu¹ Li Bing¹ Deng Fangming^{1,2}

(1. School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China;
2. School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China)

Abstract: The anti-collision algorithm determines the operating efficiency of RFID systems. An improved hybrid query tree (IHQT) anti-collision algorithm is presented in this paper, which is based on the query tree algorithm. When collision occurs, the collided node will split into some child nodes. IHQT adds a branch prediction stage before the reader query collided tags to avoid the idle slots. The branch prediction method presented in this paper can predict the location of idle slots accurately. When the reader generates new query prefixes, those query prefixes which visit the idle slots will be avoided successfully. The performance analysis of algorithm and simulation results show that under the premise of a small increase even significant decrease in the query times of reader, the number of timeslots and throughput of IHQT algorithm are significantly better than the existing query tree anti-collision algorithms.

Keywords: RFID; anti-collision algorithm; query tree; branch prediction

0 引言

随着射频识别(radio frequency identification, RFID)标签被大量应用,标签碰撞问题成为限制RFID系统工作效率的瓶颈。标签防碰撞问题本质上就是无线通信系统中的多路存取问题,基于时分多址(TDMA)的RFID防

碰撞算法,不需要硬件上的改变,在算法层面就可以实现,原理简单,被广泛应用^[1]。常用的RFID防碰撞算法分为两类,1)基于ALOHA算法^[2-3];2)基于树的算法^[4-6]。基于ALOHA的算法又被称为概率性算法,主要有两个缺点:1)标签选择响应时隙是随机的,空闲时隙和碰撞时隙的数量无法确定,在标签数量未知的情况下,不能保证所有的标签被识别^[7],可能产生标签饥饿现象

收稿日期:2017-02 Received Date: 2017-02

* 基金项目:国家自然科学基金(51577046)、国家自然科学基金重点项目(51637004)、国家重点研发计划“重大科学仪器设备开发”项目(2016YFF0102200)、安徽省科技计划重点项目(1301022036)、江西省科技支撑项目(20161BBE50076)、合肥工业大学博士专项科研基金(JZ2016HGBZ1030)资助项目

(个别标签长期读不到);2) ALOHA 算法的吞吐率低,存在 36.8% 的理论限制^[8]。基于树的算法被称为确定性算法,能够保证每一个标签都被准确识别。

经典的树算法有二进制搜索树(binary search, BS)算法^[9]、查询树(query tree, QT)算法^[10]。目前有很多算法对经典的树算法进行改进,主要思路是增加额外查询、采用碰撞位的特征值或采用分组识别的方式来减少空闲时隙和碰撞时隙。文献[11]采用增加额外查询的方式,提出一种改进的树形结构 RFID 防碰撞算法,它把整个识别过程分为前缀确定和分支查询两个阶段,通过前缀确定阶段可以将查询树分为若干分支,在每一个分支内进行标签识别,以减少标签碰撞和空闲时隙的发生。增加额外查询可以在一定程度上减少空闲时隙和碰撞时隙,但是会增加阅读器的额外开销。为了改善上述问题,文献[12-13]根据碰撞位的特征值来减少空闲时隙和碰撞时隙。其中文献[12]可以根据两个碰撞位的特征,自调整搜索树的分叉数为 2 或 4,从而避免一些碰撞时隙和空闲时隙。文献[13]提出的自适应四叉修剪查询树(A4PQT)算法根据碰撞位的特点,决定是否剪除多余的分支,进而减少不必要的空闲时隙。采用碰撞位的特征值,可以在不增加额外查询的条件下,减少空闲时隙,在一定程度上提高算法性能。文献[14]采用分组识别的方式,提出基于分组机制的位仲裁查询树(GBAQT)算法,标签根据 ID 特征,通过 3 位仲裁的方式将未识别标签分为 2 组,阅读器根据各组标签特点和接收数据的碰撞位信息得到传输数据,达到减少空闲时隙的效果。文献[15]提出的算法通过将标签分为多个组,进行分别识别,以达到减少碰撞时隙的效果。上述改进算法都只能达到减少空闲时隙的效果,并不能完全避免空闲时隙的产生。空闲时隙没有标签响应,阅读器对空闲时隙的查询无法读到任何标签信息,只会增加阅读器的额外开销,能否进一步减少甚至消除空闲时隙成为提高防碰撞算法性能的关键。

针对以上文献存在的问题,本文提出一种改进的混合查询树(improved hybrid query tree, IHQT)算法。该算法在阅读器查询碰撞标签前增加一个分支预测阶段,根

据预测结果可知道哪些节点为空闲节点,进而把空闲节点全部剪除。虽然 IHQT 算法增加分支预测查询,一定程度上增加了阅读器的开销,但是因为空闲时隙被全部消除,阅读器查询空闲时隙产生的额外开销也被消除。理论分析和仿真表明,本文提出的 IHQT 算法不会产生空闲时隙,能有效提高 RFID 系统的性能。

1 IHQT 防碰撞算法

1.1 改进算法的分支预测方法

本文提出一种分支预测的方法,旨在准确预测空闲节点的位置。算法开始时,清空堆栈。阅读器初始查询时,因为堆栈为空,所有标签都会响应。若阅读器读取范围内只有一个标签,该标签会被成功识别,查询结束;若有多个标签同时响应,根节点产生碰撞。若查询前缀 $q_1q_2\cdots q_k$ 对应树节点查询结束,并产生碰撞。阅读器在查询前缀 $q_1q_2\cdots q_k$ 后添加 n 位二进制 1 产生新的二进制序列 $q_1q_2\cdots q_k111\cdots(n$ 个 1)。待识别标签接收到分支预测指令后把前缀 $r_1r_2\cdots r_k$ 与二进制序列 $q_1q_2\cdots q_k111\cdots(n$ 个 1) 进行比较,如果前缀 $r_1r_2\cdots r_k$ 和 $q_1q_2\cdots q_k$ 匹配,标签会把 ID 前缀 $r_1r_2\cdots r_k$ 后面的 n 位二进制数字换算为对应的十进制数字 e ,并向阅读器回复一个长度为 2^n ($0 \sim 2^n - 1$) 的二进制响应序列(分支预测响应序列),序列中只有第 e 位为 1(从第 0 位开始计数),其他 $2^n - 1$ 位都为 0^[11]。阅读器接收到所有的标签响应序列后进行判断,序列中记为 0 的位置对应的子节点是空闲节点。阅读器产生新查询前缀时,会自动避开空闲节点。碰撞树节点就会根据分支预测的结果产生 $1 \sim 2^n$ 个子节点,这些子节点都会有标签响应,避免了空闲时隙的产生。

1.2 算法分析

假设在阅读器的查询范围内有 4 个标签,标签 ID 分别为 A(0010)、B(0100)、C(1001)和 D(0101)。 $n=1$ 的 IHQT 算法的识别过程如表 1 所示,采用二叉树(2-ary QT)算法和 $n=1$ 的 IHQT 算法的比较如图 1 所示。从图 1 可知, IHQT 算法的识别过程没有空闲时隙产生,碰撞节点的子节点有 1 到 2 个,没有空闲时隙。

表 1 $n=1$ 的 IHQT 算法识别标签过程

Table 1 Identification process of IHQT when $n=1$

查询轮次	1	2	3	4	5	6	7	8
分支预测 R to T	ϵ	1		01		011	0101	
分支预测 T to R	ϵ	10/01		10/01		10	10/01	
查询前缀 R to T	ϵ	0	1	00	01	010	0100	0101
标签响应 T to R	碰撞	碰撞	C 成功	A 成功	碰撞	碰撞	B 成功	D 成功
标签 A	A	A		A				
标签 B	B	B			B	B	B	
标签 C	C		C					
标签 D	D	D			D	D		D

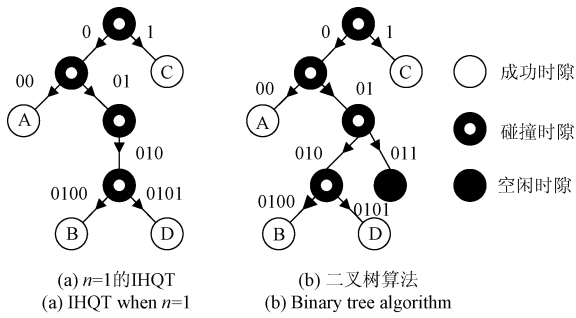


图 1 不同算法的标签响应过程比较

Fig. 1 Identification process of IHQT and binary tree algorithm

$n = 2$ 的 IHQT 算法的识别过程如表 2 所示, 二叉树 (4-ary QT) 算法和 $n = 2$ 的 IHQT 算法的比较如图 2 所示。由图 2 所示, 当 $n = 2$ 时 IHQT 算法的标签识别过程不会产生空闲时隙。图 2(a) 与图 1(a) 进行比较可知, 当待识别标签相同时, $n = 2$ 的 IHQT 算法比 $n = 1$ 的 IHQT 算法碰撞时隙更少, 性能更好。

当 $n = L$ (L 为标签 ID 的长度) 时, IHQT 算法的识别过程如表 3 和图 3 所示。由表 3 和图 3 可知, 当分支预测位数等于标签 ID 的长度 L 时, 标签的识别效率最高, 只有一个碰撞时隙, 其余时隙全是成功时隙。

表 2 $n = 2$ 的 IHQT 算法识别标签过程

Table 2 Identification process of IHQT when $n = 2$

查询轮次	1	2	3	4	5	6
分支预测 (R to T)	ε	11			0111	
分支预测 (T to R)	ε	1000/0100/0010			1000/0100	
查询前缀 (R to T)	ε	00	01	10	0100	0101
标签响应 (T to R)	碰撞	A 成功	碰撞	C 成功	B 成功	D 成功
标签 A	A	A				
标签 B	B		B		B	
标签 C	C			C		
标签 D	D		D			D

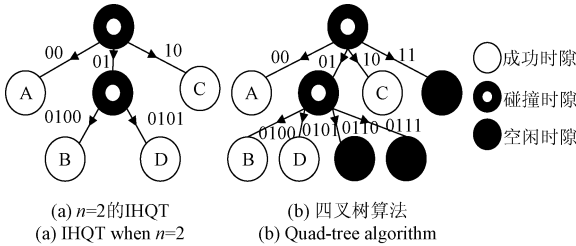


图 2 不同算法的标签响应过程比较

Fig. 2 Identification process of IHQT and Binary tree algorithm

1.3 算法流程

IHQT 算法流程分为阅读器查询流程和标签响应流程, 如图 4 和 5 所示。算法指令及其作用如下:

INIT: 初始化命令, 清空堆栈。

FORECAST-BRANCH($q_1 q_2 \dots q_k 111 \dots, n$ 个 1): 分支预测指令, 碰撞产生后, 阅读器会向所有待识别标签发送分支预测指令。

REQUEST($q_1 q_2 \dots q_k$): 查询指令, 其中 $q_1 q_2 \dots q_k$ 是查询前缀。标签接收到查询指令后把 ID 前缀与查询前

表 3 $n = L$ 的 IHQT 算法识别标签过程

Table 3 Identification process of IHQT when $n = L$

查询轮次	1	2	3	4	5
分支预测 (R to T)	ε	1111			
分支预测 (T to R)	ε	16 位序列 2, 4, 5, 9 位为 1			
查询前缀 (R to T)	ε	0010	0100	0101	1001
标签响应 (T to R)	碰撞	A 成功	B 成功	D 成功	C 成功
标签 A	A	A			
标签 B	B		B		
标签 C	C				C
标签 D	D			D	

缀进行比较, 若匹配, 则标签响应并向阅读器发送剩余的 ID 信息。

该指令, 标签信息被成功读取。

PUSH: 产生新的查询前缀时 $q_1 q_2 \dots q_k XXX \dots$ 时, 执行该指令, 把新查询前缀压入堆栈。

READ-DATA: 读取指令, 只有一个标签响应时, 执行

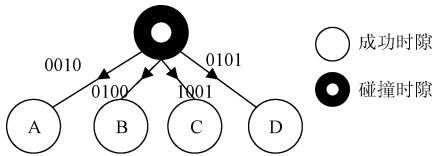


图3 n=L时 IHQT 算法识别过程

Fig.3 Identification process of IHQT when n=L

标签状态分为如下4种:

等待状态 (waiting status): 标签等待接收阅读器的分支预测指令 (FORECAST-BRANCH) 和查询指令 (REQUEST)。

激活状态 (activated status): 标签接收阅读器发送的分支预测指令 (FORECAST-BRANCH) 和查询指令 (REQUEST), 并把查询字符串与自身的前缀进行比较, 若匹配, 就会进入激活状态, 并根据不同的指令做出不同的响应。

成功状态 (success status): 若一个时隙有且只有一个标签响应, 阅读器执行 READ-DATA 命令后, 标签进入成功状态, 进入成功状态的标签不能被重复读取。

碰撞状态 (collision status): 一个时隙有超过一个标签同时响应, 碰撞产生, 标签状态是碰撞状态, 没有被成功识别。

1.3.1 阅读器查询流程

1) 初始化, 执行 INIT 指令。

2) 阅读器广播查询指令 REQUEST(q₁q₂...q_k), 是否有标签响应? 若没有, 转到步骤6); 若有, 首次查询前缀为空, 阅读器识别范围内的标签全部响应; 若不是首次查询, 标签把 ID 前缀与查询前缀 q₁q₂...q_k 进行比较, 若匹配, 标签响应并把剩余 ID 信息发送给阅读器。

3) 判断树节点是否产生碰撞? 若没有, 执行指令 READ-DATA, 并转到步骤6); 若碰撞, 转到步骤4)。

4) 阅读器执行指令 FORECAST-BRANCH(q₁q₂...q_k111...n个1)。不生成空闲子节点对应的查询前缀, 生成非空闲子节点对应的查询前缀, 对生成的查询前缀执行 PUSH 指令。

5) 堆栈弹出查询前缀, 转到步骤2)。

6) 判断堆栈是否为空? 若为空, 则查询结束, 转到步骤7); 若不为空, 则转到步骤5)。

7) 结束。

1.3.2 标签响应流程

1) 标签为等待状态。

2) 接收到阅读器发送的广播命令 REQUEST(q₁q₂...q_k), 标签进入激活状态。

3) 若标签是首次响应, 查询前缀为 ε, 所有标签响应; 若不是首次响应, 标签将 ID 前缀 r₁r₂...r_k 与查询前缀

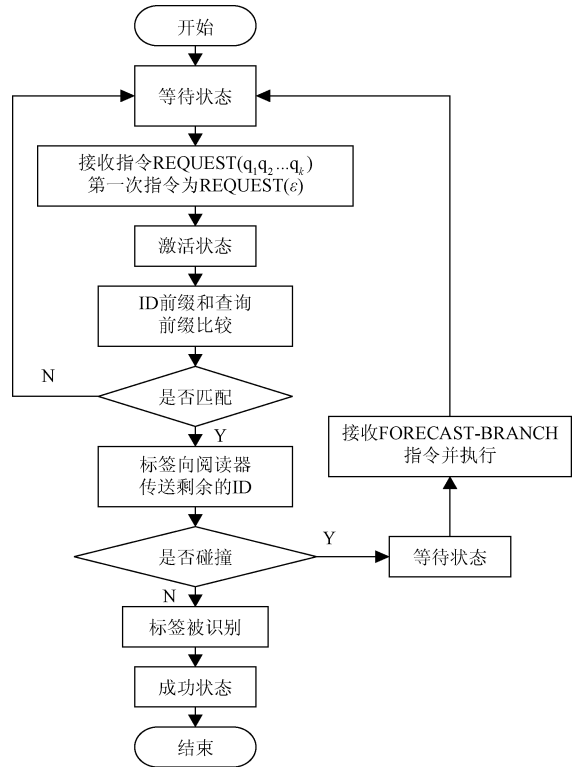


图4 IHQT 算法阅读器查询流程

Fig.4 Flow chart of reader query in IHQT

q₁q₂...q_k 比较, 若匹配, 转到步骤4); 若不匹配转到步骤1)。

4) 发送剩余的 ID 信息。

5) 判断标签是否碰撞? 若碰撞, 转到步骤6); 若不碰撞, 转到步骤8)。

6) 进入等待状态。

7) 接收分支预测指令 FORECAST-BRANCH(q₁q₂...q_k111...n个1), 标签进入激活状态, 并根据指令向阅读器发送分支预测响应序列。转到步骤1)。

8) 成功识别, 进入成功状态。

2 算法性能分析

对于一个 RFID 系统, 衡量防碰撞算法的指标主要有3个: 时隙数、吞吐率和阅读器查询次数(阅读器开销)。

2.1 时隙数

树形 RFID 防碰撞算法的总时隙数等于树的节点数。当分支预测位数 n 小于标签 ID 长度 L 或等于标签 ID 长度 L 时, IHQT 算法本质上是把 2ⁿ 叉树和 2^L 叉树的空闲节点全部剪除, 只留下碰撞节点和成功节点。

首先分析二叉树算法, 假设标签数为 m, 碰撞时隙数 m₁, 空闲时隙数 m₂, 成功时隙数 m₃。

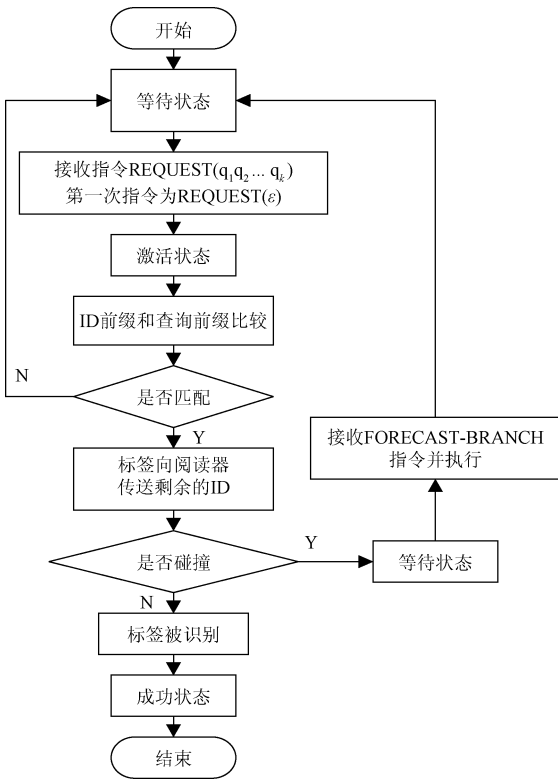


图 5 IHQT 算法标签响应流程

Fig. 5 Flow chart of tags response in IHQT

$$m_3 = m \tag{1}$$

二叉树总时隙数:

$$M = m_1 + m_2 + m_3 \tag{2}$$

每一个碰撞节点有两个子节点,空闲节点和成功节点没有子节点,根节点不是任何节点的子节点。则二叉树总时隙数可以表示为:

$$M = 2 \times m_1 + (m_2 + m_3) \times 0 + 1 = 2 \times m_1 + 1 \tag{3}$$

二叉树中每一个碰撞节点最多产生 1 个,最少产生 0 个空闲子节点。若每一个碰撞节点都产生 1 个空闲子节点,则 $m_2 = m_1$;若每个碰撞节点产生 0 个空闲子节点,则 $m_2 = 0$ ($n = 1$ 时的 IHQT 算法)。

分支预测位 $n = 1$ (且标签 ID 长度 $L > 1$) 时, IHQT 算法总时隙数为:

$$M = 2m - 1 \tag{4}$$

再分析四叉树算法,每一个碰撞节点有 4 个子节点,空闲节点和成功节点没有子节点,根节点不是任何节点的子节点。则四叉树总时隙数为:

$$M = 4 \times m_1 + 1 \tag{5}$$

当 $n = 2$ (且标签 ID 长度 $L > 2$) 时, IHQT 算法的混合树没有空闲节点。所以 $m_2 = 0$ 。 $n = 2$ 时 IHQT 算法的总时隙数为:

$$M = (4m - 1)/3 \tag{6}$$

当 $n = 3$ (且标签 ID 长度 $L > 3$) 时, IHQT 算法的总时隙数为:

$$M = (8m - 1)/7 \tag{7}$$

当 $n = 4$ (且标签 ID 长度 $L > 4$) 时, IHQT 算法的总时隙数为:

$$M = (16m - 1)/15 \tag{8}$$

分支预测位数 n 等于标签 ID 的长度 L 时,所有标签都会在根节点的子节点上被识别,所以 IHQT 算法的总时隙数为:

$$M = \begin{cases} 1, & m = 1 \\ m + 1, & m > 1 \end{cases} \tag{9}$$

2.2 吞吐量

吞吐量 S 又被称为算法效率,是成功时隙数与总时隙数的比值。本文提出的分支预测位 $n \leq L$ 时, IHQT 算法的吞吐量分别为:

$$S_n = \frac{m}{(2^n \times m - 1)/(2^n - 1)} \tag{10}$$

$$S_L = \begin{cases} 1, & m = 1 \\ \frac{m}{m + 1}, & m > 1 \end{cases} \tag{11}$$

2.3 阅读器查询次数

经典多叉树算法的阅读器查询次数等于成功时隙数、碰撞时隙数和空闲时隙数之和 ($m_1 + m_2 + m_3$)。本文提出的 IHQT 算法增加了分支预测查询,从这个角度是增加了阅读器的查询次数,即增加了阅读器的开销。但是因为分支预测,可以消除所有的空闲时隙,避免了阅读器对空闲时隙的查询,在一定程度上又减少了阅读器的开销。

在 IHQT 算法中,阅读器需要对碰撞时隙查询两次,第一次查询得知该时隙为碰撞时隙,第二次查询预测该碰撞时隙的子节点中哪一个节点为空闲时隙。阅读器对成功时隙只需要一次查询就可以获取标签的信息。所以阅读器的查询次数 Y :

$$Y = 2 \times m_1 + m_3 \tag{12}$$

当分支预测位数 n 小于标签 ID 长度 L 时:

$$m_1 = (m - 1)/(2^n - 1) \tag{13}$$

所以当 $n < L$ 时:

$$Y_n = \frac{2(m - 1)}{2^n - 1} + m \tag{14}$$

当 $n = L$ 时,只有根节点为碰撞节点,根节点的子节点都是成功时隙。若在识别范围内只有一个标签 ($m = 1$),则阅读器一次查询就能识别,若在识别范围内标签数 $m > 1$ 时,阅读器的查询次数为 $m + 1$ 。

$$Y_L = \begin{cases} 1, & m = 1 \\ m + 1, & m > 1 \end{cases} \tag{15}$$

3 仿真与分析

本文采用 MATLAB 软件对算法进行仿真。图 6 和 7 所示分别是当标签数量从 0 ~ 500 时,不同分支预测位的 IHQT 算法的总时隙数和碰撞时隙数比较结果。图 6 中可以看出随着标签数量增多时,IHQT 算法的总时隙数会不断增多。但是当标签数量相等时,分支预测位越大,算法的总时隙数越小。图 7 中可以看出当分支预测位越大时,碰撞时隙数越少。由于 IHQT 算法中没有空闲时隙,成功时隙数等于标签数量,所以从图 7 得出的结论可以验证图 6 中得出的结论。

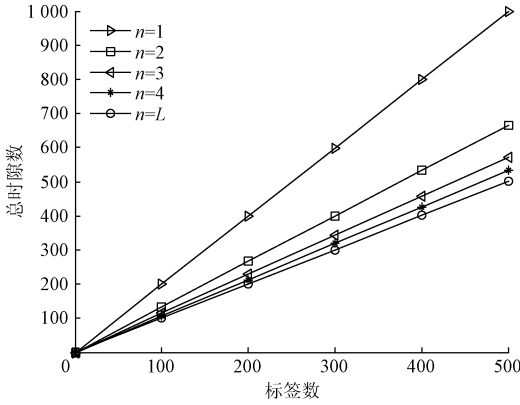


图 6 IHQT 算法不同分支预测位总时隙比较
Fig. 6 The total slot comparison of IHQT with different branch prediction bit

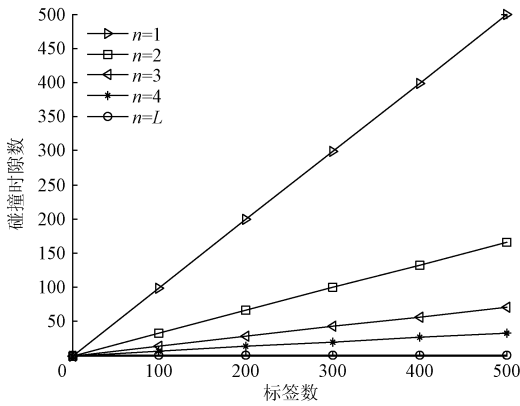


图 7 IHQT 算法不同分支预测位数碰撞时隙比较
Fig. 7 The collision slot comparison of IHQT with different branch prediction number

图 8 所示为分支预测位不同时,IHQT 算法的吞吐率比较。可以看出,当标签数量众多的情况下,分支预测位越大,IHQT 算法的吞吐率越大。分支预测位 $n = 1$ 时,IHQT 算法的吞吐率随标签数量增多逐步趋近于 0.5;分支预测位 $n = 2$ 时,IHQT 算法的吞吐率随标签数量增多

逐步趋近于 0.75;分支预测位 $n = 3$ 时,IHQT 算法的吞吐率随标签数量增多逐步趋近于 0.875;分支预测位 $n = 4$ 时,IHQT 算法的吞吐率随标签数量增多逐步趋近于 0.938;分支预测位 $n = L$ (L 为标签 ID 的长度)时,IHQT 算法的吞吐率随标签数量增多逐步趋近于 1。

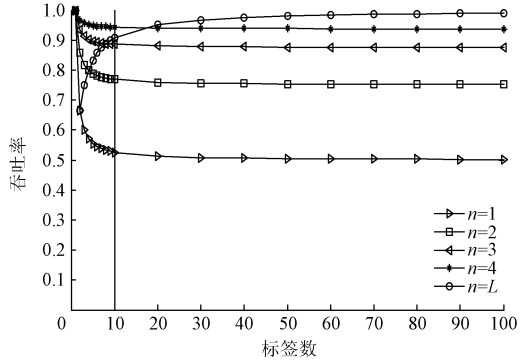


图 8 IHQT 算法不同分支预测位数吞吐率比较
Fig. 8 The throughput comparison of IHQT with different branch prediction number

图 9 所示为分支预测位等于 2 的 IHQT 算法、AHT 算法、GBAQT 算法和 A4PQT 算法总时隙数的比较图。可以看出在识别标签数量相同的情况下,分支预测位等于 2 的 IHQT 算法总时隙数小于 AHT 算法、GBAQT 算法和 A4PQT 算法。当标签数量为 500 时,分支预测位等于 2 的 IHQT 算法的总时隙数大约是 660,AHT 算法的总时隙数大约是 900,GBAQT 算法的总时隙数大约是 930,A4PQT 算法的总时隙数大约是 1 050。所以 IHQT 算法可以有效的减少搜索的总时隙数,性能相较于其他 3 种算法性能上有明显的优势。

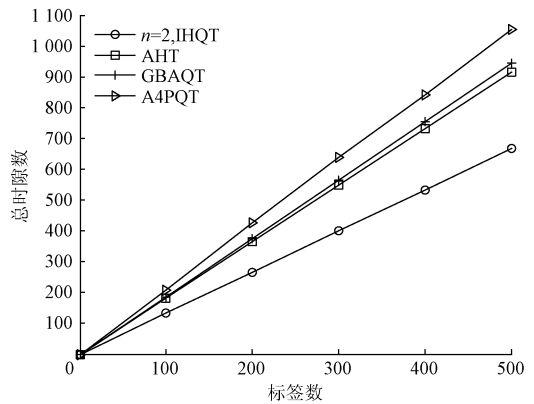


图 9 不同算法在识别相同标签情况下总时隙数比较
Fig. 9 The total time slot comparison of different algorithms

图 10 所示为对分支预测位等于 2 的 IHQT 算法、AHT 算法、GBAQT 算法和 A4PQT 算法的吞吐率进行了比较,可以看出分支预测位等于 2 的 IHQT 算法的系统效

率明显好于 AHT 算法、GBAQT 算法和 A4PQT 算法。当标签数量足够多时,分支预测位等于 2 的 IHQT 算法的吞吐率约在 0.75, AHT 算法的吞吐率大约为 0.55, GBAQT 算法的吞吐率保持在 0.54 左右。A4PQT 算法的吞吐率最低,约只有 0.49。

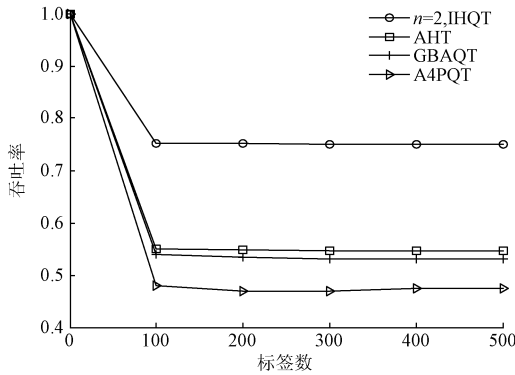


图 10 不同算法在识别相同标签情况下吞吐率比较
Fig. 10 The throughput comparison of different algorithms

图 11 展示的是二叉树、 $n=1$ 的 IHQT、4 叉树、 $n=2$ 的 IHQT、八叉树、 $n=3$ 的 IHQT 算法、十六叉树、 $n=4$ 的 IHQT 算法的阅读器查询次数随标签数量增加时的变化曲线。由图 11 可知,当标签数量相同时, $n=1$ 的 IHQT 算法和 $n=2$ 的 IHQT 算法比二叉树算法和四叉树算法的阅读器查询次数多, $n=3$ 的 IHQT 算法和 $n=4$ 的 IHQT 算法比八叉树算法和十六叉树算法的阅读器查询次数少。说明当分支预测位数 $n \leq 2$ 时, IHQT 算法比对应的多叉树算法阅读器开销大。当分支预测位数 n 增加时, IHQT 算法的碰撞时隙减少,又因为没有空闲时隙,阅读器开销比对应的多叉树算法小。本文提出的 IHQT 算法,虽然增加了额外的查询,但是因为空闲时隙全部被消除,且随着分支预测位数 n 增加,其碰撞时隙数减少,只要分支预测位数 $n \geq 3$, 其阅读器开销小于对应的多叉树

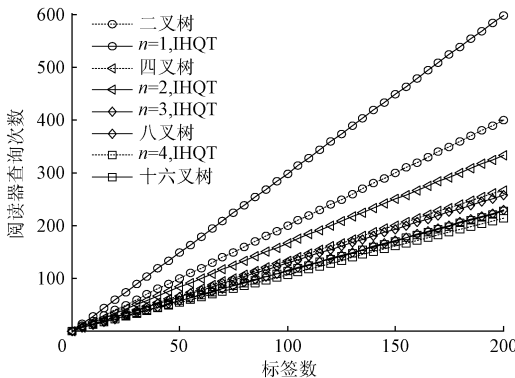


图 11 IHQT 与对应树算法阅读器开销比较
Fig. 11 The reader overhead comparison of IHQT and corresponding tree

算法。当分支预测位数 $n=L$ (标签 ID 长度) 时,由 2.3 节可知, IHQT 算法的阅读器查询次数为标签数 $m+1$ 。对应的 $2L$ 叉树算法的树节点有 $2L$ 个分支,阅读器查询次数为 $2L+1$,远远大于 $m+1$ 。

多叉树算法的改进思路主要是增加额外查询、采用碰撞位的特征值或采用分组识别的方式来减少空闲时隙和碰撞时隙。由表 4 所示,本文提出的 IHQT 通过额外查询的方式, IHQT 算法的混合查寻树中没有空闲时隙,当分支预测位数 $n \geq 2$ 时, IHQT 算法的吞吐率高于已有的改进树算法,且当 n 增大时,阅读器开销随之降低。

表 4 不同算法整体性能比较

Table 4 The overall performance comparison of different algorithms

算法	额外查询	分组识别	特征值识别	空闲时隙	阅读器开销 ($m=50$)	吞吐率
IHQT $n=1$	是	否	否	无	148	0.5
IHQT $n=2$	是	否	否	无	82	0.75
IHQT $n=3$	是	否	否	无	64	0.875
IHQT $n=4$	是	否	否	无	57	0.938
IHQT $n=L$	是	否	否	无	51	~1
[11]	是	是	否	有	70	0.50
[12]	否	否	是	有	76	0.55
[13]	否	否	是	有	85	0.49
[14]	否	是	否	有	--	0.54
[15]	是	否	否	有	92	0.61

4 结 论

分支预测可以准确预测所有空闲时隙,在产生新的查询前缀时避免产生访问空闲时隙的查询前缀,本文在已有改进的树算法基础上,提出了一种 IHQT 算法,该算法在阅读查询碰撞标签前增加了一个分支预测阶段,可以成功消除查询树中的所有空闲时隙。本文提出的混合查询树本质上是把对应的多叉查寻树中空闲时隙全部剪除后的树形结构,当所有的空闲时隙全部被剪除,则可排除空闲时隙对算法性能的影响,因此, IHQT 的性能明显优于其他基于树的防碰撞算法。分支预测位决定了混合查寻树对应的查询树的分叉数,很大程度上影响 IHQT 算法的性能,分支预测位为 n 时,混合查询树对应 $2n$ 叉查寻树,增加分支预测查询会一定程度上导致阅读器开销增大,同时,分支预测位数的增加会减小碰撞时隙。本文的分析与仿真结果表明:混合查询树的碰撞节点数量会随分支预测位的增大而减小, IHQT 算法的性能也随之提高,当分支预测位数 $n \geq 3$ 时, IHQT 算法的阅读器开销优于对应的多叉树算法。

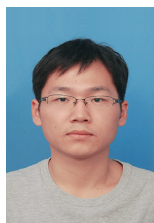
参考文献

- [1] ZHAI C, ZOU Z, CHEN Q, et al. Optimization on guard time and synchronization cycle for TDMA-based deterministic RFID system [C]. IEEE International Conference on RFID Technology and Applications, 2015: 71-75.
- [2] CHEN W T. An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA [J]. IEEE Transactions on Automation Science and Engineering, 2009, 6(1):9-15.
- [3] BARLETTA L, BORGONOVO F, AND FILIPPINI I. Asymptotic analysis of schoute's estimate for dynamic frame aloha [C]. 23rd International Conference on Software, Telecommunications and Computer Networks, 2015:113-118.
- [4] JIANG Y J, XU Y F, WANG Q. Cancellation strategy in dynamic framed slotted ALOHA for RFID system [C]. IEEE Wireless Communications and Networking Conference (WCNC), 2013: 854-859.
- [5] 黄刚. 电子标签的防碰撞应用研究 [J]. 电子测量与仪器学报, 2012, 26(4):372-377.
HUANG G. Research on application of RFID anti-collision of electronic tag [J]. Journal of Electronic Measurement and Instrument, 2012, 26(4):372-377.
- [6] 李志坚, 赖顺桥. 一种基于碰撞位指示的射频识别标签防碰撞算法 [J]. 电子与信息学报, 2014, 86(12): 2842-2847.
LI ZH J, LAI SH Q. An anti-collision algorithm based on collided bits indicator in radio frequency identification systems [J]. Journal of Electronics & Information Technology, 2014, 86 (12): 2842-2847.
- [7] YANG C N, HU L J, LAI J B. Query tree algorithm for RFID tag with binary-coded decimal EPC [J]. IEEE Communications Letters, 2012, 16(10): 1616-1619.
- [8] 王勇, 唐小虎, 张莉娟. RFID系统中停留标签的组策略防碰撞算法 [J]. 电子与信息学报, 2016, 88(8): 594-599.
WANG Y, TANG X H, ZHANG L J. Group strategy for remaining tags in anti-collision algorithm for RFID system [J]. Journal of Electronics & Information Technology, 2016, 88(8):594-599.
- [9] HE Y J, WANG X Y. An ALOHA-based improved anti-collision algorithm for RFID systems [J]. IEEE Wireless Communications, 2013, 20(5):152-158.
- [10] MYUNG J H, LEE W J, SRIVASTAVA J. Adaptive binary splitting for efficient RFID tag anti-collision [J]. IEEE Communications Letters, 2006, 10(3): 144-146.
- [11] 王雪, 钱志鸿, 刘晓慧, 等. 改进的树形结构 RFID 防碰撞算法 [J]. 通信学报, 2015, 36(7):129-137.
WANG X, QIAN ZH H, LIU X H, et al. Improved tree

structure anti-collision algorithm of RFID [J]. Journal on Communications, 2015, 36(7):129-137.

- [12] 宋建华, 郭亚军, 韩兰胜, 等. 自调整混合树 RFID 多标签防碰撞算法 [J]. 电子学报, 2014, 42(4): 685-689.
SONG J H, GUO Y J, HAN L SH, et al. An adjustable hybrid tree anti-collision algorithm for RFID multi-tag identification [J]. Acta Electronica Sinica, 2014, 42(4):685-689.
- [13] ZHANG W, GUO Y J, TANG X M, et al. An efficient adaptive anticollision algorithm based on 4-ary pruning query tree [J]. International Journal of Distributed Sensor Networks, 2013, 14(2): 135-143.
- [14] 付钰, 钱志鸿, 程超, 等. 基于分组机制的位仲裁查询树防碰撞算法 [J]. 通信学报, 2016, 37(1):123-129.
FU Y, QIAN ZH H, CHENG CH, et al. Bit arbitration query tree anti-collision algorithm based on grouping mechanism [J]. Journal on Communications, 2016, 37(1):123-129.
- [15] LAI Y C, HSIAO L Y, CHEN H J, et al. A novel query tree protocol with bit tracking in RFID tag identification [J]. IEEE Transactions on Mobile Computing, 2013, 12(10):2063-2075.

作者简介



史露强, 2015 年于华侨大学获得学士学位, 现为合肥工业大学博士研究生, 主要从事 RFID 技术及智能电网技术的研究。

E-mail: 2015110409@mail.hfut.edu.cn

Shi Luqiang received B. Sc. from Huaqiao University in 2015. He is currently a Ph. D. candidate in Hefei University of

Technology. His research interest includes RFID technology and smart grid technology.



何怡刚(通讯作者), 1992 年于湖南大学获得硕士学位, 1996 年于西安交通大学获得博士学位, 现为合肥工业大学教授、博士生导师, 合肥工业大学电气与电子工程学院院长, 国家杰出青年科学基金获得者, 主要研究方向为模拟和混合集成电路设计、测试与故障诊断、智能电网技术、射频识别技术、虚拟仪器和智能信号处理。

E-mail: 18655136887@163.com

He Yigang (Corresponding author) received M. Sc. from Hunan University in 1992 and Ph. D. from Xi'an Jiaotong University in 1996, respectively. He is currently a professor and the dean of School of Electrical Engineering and Automation, Hefei University of Technology. He is one of winners of National Distinguished Young Scientists Foundation. His research interests are in the areas of circuit theory and its applications, testing and fault diagnosis of analog and mixed-signal circuits, smart grid, radio frequency identification technology, and intelligent signal processing.