

基于 EPC Gen2 防碰撞机制的研究与优化

吴 森^{1,2}, 刘德盟^{1,2}, 张钊锋¹

(1 中国科学院上海高等研究院, 上海 201210; 2 中国科学院研究生院, 北京 100190)

摘 要: RFID 系统必须解决多个标签同时接入阅读器发生碰撞的问题. 文中针对 EPC Gen2 算法中 Q 值调整算法和多标签读取机制的不足, 在不改变原有命令格式和不增加命令的条件下提出优化方案, 通过对多种方案的分析比较提出改进的多标签响应机制.

关键词: RFID; 防碰撞; EPC Gen2; Q 值

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2013)05-0100-04

Study and Optimization of the Anti-collision Mechanism Based on EPC Gen2 Protocol

WU Miao^{1,2}, LIU De-meng^{1,2}, ZHANG Zhao-feng¹

(1 Shanghai Advanced Research Institute, Chinese Academy of Science, Shanghai 201210, China;

2 Graduate University of the Chinese Academy of Science (GUCAS), Beijing 100190, China)

Abstract: The problem of the collision when multiple tags access to a reader at the same must be solved in RFID system. This paper introduces the anti-collision algorithm in the EPC Gen2 protocol, and optimize the standard Q value adjustment algorithm and multi-tag reading mechanisms under the conditions that do not change the original order form and not to increase the command, through the analysis of a variety of programs to propose an optimized multi-label response mechanism.

Key words: RFID; anti-collision; EPC Gen2; Q value

1 引言

RFID 系统的射频通信部分包括阅读器(Reader/Interrogator)和标签(Tag/Label). 阅读器发送的消息被其电磁能量覆盖内的多个标签接收到, 标签通过对电磁波背射调制(backscatter)的方式对阅读器做出应答. 当多个标签将信息同时返回一个阅读器时将发生标签数据混叠, 不能被阅读器准确读取. 为了解决此类问题标签碰撞问题必须采用标签防碰撞算法.

EPC Gen2^[1] 规定了基于动态帧时隙随机算法(Dynamic Framed Slotted ALOHA, DFSA)的 Q 算法(时隙-计数算法: slot-count algorithm)用于解决标签碰撞问题, 该标签防碰撞算法具有一定的自适应性并表现出较好吞吐率性能. 在以往的一些

文章中针对 EPC Gen2 标准防碰撞算法的改进方案都需要对现有的命令帧格式进行更改, 或者需要增加额外的命令条目^[2], 需要在标签端增加相应的设计复杂性和硬件开销. 这无疑给现已广泛生产使用 EPC Gen2 标准标签生产带来难题. 本文对 EPC Gen2 标准中的防碰撞算法进行优化, 在不改变原有 EPC Gen2 标准协议中的指令格式和不增加指令的前提下提出一种新的标签防碰撞机制, 可以有效降低标签识别延时, 减少标签端一定的硬件开销, 能更快更容易地应用到现有的 RFID 标签产品生产.

2 EPC Gen2 标签防碰撞算法的不足及改进方案

在 EPC Gen2 的标签防碰撞算法中由读写器规定一段时间长度(包含若干个时隙), 当标签接收到

读写器命令后随机选择一个时隙进行接入. 读写器通过 Query, QueryRep, QueryAdjust 等指令的组合对标签中时隙计数器数值进行调整, 当计数器值为 0 时标签进行响应. 将发送 Query 或 QueryAdjust 命令的间隔作为一帧的长度. 在一个盘存 (Inventory) 周期中可以在任意时刻更改时隙数, 使未识别标签重新选择响应时隙, 进入下一帧的响应周期中.

EPC Gen2 标准可以在识别数量较大的标签时仍然保持在一定的系统吞吐率和较小的识别延时; 标签产生的 16 位伪随机数的分布不会对标签群产生影响. 但是该算法仍然存在不足之处, 最大帧长 2^{15} 是一个很大的数字; 但在实际应用中标签的数目一般不超过几千个, 这无疑造成了寄存器资源的浪费. EPC Gen2 标准中 Q 值的调整依赖于参数 C 的取值. Q 值可能出现反复性的变化, 不断的变化帧长, 需要反复发送 QueryAdjust 命令, 容易造成时隙资源浪费, 增加标签识别所需功耗, 同时也使系统增加了不稳定因素. RFID 标签往往追求低成本, 低功耗, 设计中应尽可能的减少硬件资源开销和标签识别功耗.

针对上述的不足之处, 本文尝试在不改变原有 EPC Gen2 标准协议存在的指令格式和不增加指令的前提下优化原有的防碰撞机制, 旨在提高系统稳定性, 吞吐率和降低标签识别延时.

2.1 随机时隙分组方案

Query 命令中的 Q 值为 4bit, 由 Q 值决定随机数的取值范围 $[0, 2^Q - 1]$, 则标签端的需要 15 位的时隙计数器. 且这 15 位的标签寄存器需要在每个时隙进行改变, 不能与其他的寄存器进行复用. 考虑借鉴动态帧时隙 (DFSA) 和分组 DFSA 算法的思想, 根据 Q 值将标签分成若干组进行分组识别. 假如设定每组的最大时隙数为 256 个, 即标签端需要 8 位的时隙计算器, 分组数 G 由 Q 值决定, 当 Q 值大于 8 时, G 取值为 $Q - 8$, 当 Q 值小于等于 8 时, G 取值为 0; 且在一组识别完成后的初始 Q 值为 8, 在识别 256 个标签时处于最佳 Q 值状态. 在标签端的逻辑可如下表示:

$G_reg[2:0] \leq ! Q_reg[3] || \sim (| Q_reg[2:0]) ? 0 : Q_reg[2:0]$

标签在时隙结束时接收到 QueryRep 命令, 时隙计数器进行减 1 操作; 时隙计数器为 0 和分组数为 0 的标签对阅读器进行响应. 改进后的随机时隙分组机制的一帧最大长度为 256, 可识别的标签数量为 $8 \times 256 = 2048$ 个, 减少了寄存器开销和 Q 值

调整次数, 且满足当下的 RFID 系统应用. 同理亦可以将每组设置为 128 个标签, 可以根据实际需要识别的标签数情况进行调整分组数和每组中标签数量, 减少 Q 值的调整次数.

2.2 快速 Q 值调整方案

EPC Gen2 标准中的 Q 值根据每个时隙是否发生碰撞的情况和 C 的取值进行改变, 需要实时检测 $Round(Q_{fp})$ 的值, 一旦 $Round(Q_{fp})$ 的值发生改变即发送 QueryAdjust 命令开启新的一帧. Q 值可能发生反复的变化, 这增加了读写器的计算复杂度、影响系统稳定性也增加标签端识别功耗. 标签端处理较长的指令需要更多的时间亦消耗更大的功耗. 这就要求在识别数量较大的标签时需要快速定位于较准确的 Q 值以减少碰撞发生次数.

对 EPC Gen2 防碰撞算法进行分析, 当接入信道连续的发生碰撞, Q 值平均每 $1/C$ 次进行加 1, 当接入信道连续空闲时, Q 值平均每 $1/C$ 次进行减 1 操作^[3]. 若采用基于经验值的 $C = 0.8/Q$ 进行调整 Q_{fp} 的值^[4], 则忽略了考虑碰撞和空闲发生的概率. 算得到的发生碰撞和空闲的概率比^[5], 如 (1) 式所示. 在进行一次盘存过程中为了使 Q 可以快速调整为理想值, 根据发生碰撞和空闲的情况分别定义不同的 C 值^[6]分别为 C_c 和 C_i , 分别将 C_c 和 C_i 按比值定义为一定值: $C_c = 0.21183$ 和 $C_i = 0.15$.

$$P_r = \frac{P_c}{P_i} = e - 2 \quad (1)$$

$$T_R = \frac{T_{coll}}{T_{idle}} = 1.98 \quad (2)$$

$$C_R = \frac{C_c}{C_i} = \frac{P_c T_{coll}}{P_i T_{idle}} = 1.4122 \quad (3)$$

本方案分别考虑根据发生碰撞碰撞和空闲的概率和在 EPC Gen2 协议下发生碰撞和空闲时需要处理的时间, 进行动态调整 C 值. 发生碰撞和空闲情况的处理时间需要考虑发送指令所需要的时间, 发送 RN16 的时间, 等待时间等. 假定发送的指令为 QueryAdjust; 可以分别得到处理碰撞和空闲所需的时间^[6] $T_{coll} = 516.625 \mu s$, $T_{idle} = 260.625 \mu s$, 时间比为 T_R 如式 (2) 所示. 据此考虑的 C_c 和 C_i 之比 C_R 为 1.4122. C_c 和 C_i 将被设置为动态值, C_i 值为 0.100 ~ 0.354, C_c 值为 0.14122 ~ 0.49992. 经过仿真验证可以在不降低吞吐率的情况下, 进一步减少碰撞发生次数, 从而进一步降低识别所耗时延而提高识别效率.

2.3 规定步长的方案

为保持一定的吞吐率和避免 EPC Gen2 标准下

Q 值出现反复的跳变,需要反复的发送 QueryAdjust 命令,产生没必要的时延和消耗额外的功耗.考虑加入步长的思想,即在规定的步长 T 内才调整 Q 值,亦可以考虑在规定的步长内对发生碰撞和空闲的次数进行计数,再根据关系动态调整 T 值.规定步长 Q 值调整算法的流程如图 1 所示.同时在 Query/QuerAdjust 指令后在阅读器上载入的时隙计数器 $RS=2^Q$,每个时隙后 RS 进行减 1 操作,当 $RS=0$ 或 Q 值改变时开始新的一帧.经过仿真后可知 Q 值比 EPC GEN2 标准趋于稳定,即无需多次的反复发送 QueryAdjust 指令,取而代之的是 QueryRep 指令,可以进一步减少时延,降低功耗.

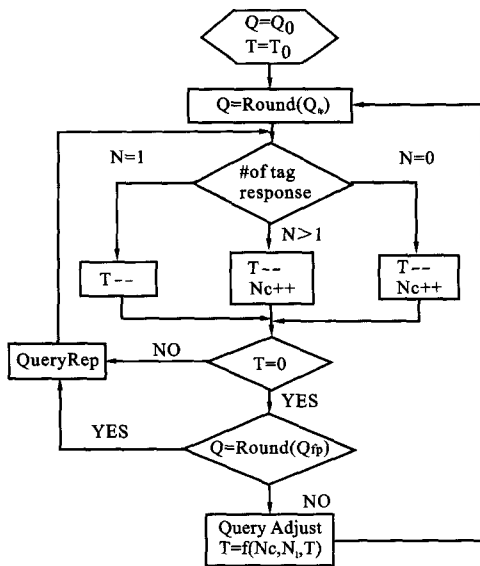


图 1 步长 Q 值调整方案流程图

2.4 内插增加不同帧长的方案

帧长的调整依赖于 Q 值,当帧长一定时(2^Q),在某一个时隙内标签接入成功、发生碰撞及空闲的概率^[7]如式(4)所示.通过仿真得知标签的数目跟帧长相当时,接入的成功概率最高.但是在 EPC Gen2 标准的协议中帧长以指数方式增长,例如当 $Q=10$ 和 11 时,帧长为 1024 和 2048,当标签的数量在它们之间时会降低接入成功概率,发生碰撞和空闲的时隙明显增加,并且根据 Gen2 中的 Q 值算法 Q 值会在 11 和 12 之间反复跳变这样不利于系统的稳定性,且增加了系统功耗和识别延后.

$$\begin{cases} P_{\text{Idle}} = \left(1 - \frac{1}{2^Q}\right)^{N_{\text{Tag}}} \\ P_{\text{Coll}} = 1 - P_{\text{Idle}} - P_{\text{Succ}} \\ P_{\text{Succ}} = \frac{N_{\text{Tag}}}{2^Q} \left(1 - \frac{1}{2^Q}\right)^{N_{\text{Tag}}-1} \end{cases} \quad (4)$$

本方案根据实际应用标签数量的情况,将内插

不同帧长以调整相应的标签数,尝试做出调整如表 1 所示.

表 1 调整后 Q 值与帧长的关系

Q	0~9	10	11	12	13	14	15
Length	2^Q	768	1024	1280	1536	1792	2048

一般来说,标签端时隙计数器上的 15 位随机数由模拟端的 16 位随机数发生器提供,它同时复用于产生 handle 值或 RN16.采用此方案后,标签端需要在原有基础上进行微小的改动.例如,当 $Q=13$ 时可以用如下方式在标签时隙计数器产生实现 0~1536 的随机数:

$$\text{slot_counter_reg} \leq 5'h00, \text{RN16_i}[9:0] \} + \{6'h00, \text{RN16_i}[8:0]\}$$

相比 EPC Gen2 标准标签端随机数产生机制改动不大.仿真当进行 2048 个标签的读取时,仿真结果显示在提高系统吞吐率的同时明显减少了 QueryAdjust 的发送次数,降低了标签识别所需功耗.

2.5 随机分组快速 Q 值定位方案

本方案综合随机分组和快速定位 Q 值的方法,即在标签硬件设计时采用优化的随机分组方案,读写器端的 Q 值调整采用快速 Q 值定位的方案.采用此方案可以有效减少发生碰撞次数,减少 Q 值的变化频率即减少了发送 QueryAdjust/Query 的发送次数,并且在标签需要分组时可以保持比 EPC Gen2 和上述几种方法都高的吞吐率.在确定标签的应用场合时(确定标签识别时大致数量)即可以采用随机分组快速 Q 值定位的思想进行标签端设计,对于减少标签识别时延和识别功耗可以有较大的提高.

3 仿真结果分析

针对 EPC Gen2 防碰撞算法存在的问题,采用上述的改进办法分别进行仿真验证.标签数量为 128~2048 个,由于算法属于随机 ALOHA 范畴,采取多次仿真将结果取平均值的方式可以更准确得到真实值.在性能的优劣上主要看对系统吞吐率是否有提高、碰撞发生次数是否减少、以及发送 QueryAdjust/Query 指令的数量是否有减少.

以上述的 3 个具体指标作为评价标准,仿真得到的平均值如图 2~图 4 所示.仿真规定步长算法中的步长为 4,快速 Q 值定位算法中将 C_c 和 C_i 设置为动态值, C_i 值为 0.100~0.354, C_c 值为 0.14122~0.49992,并且规定 Q 值越大则相应的 C 值越小.随

机分组快速 Q 值定位算法中 C_c 和 C_i 设置为动态值, C_i 值为 0.12185~0.3540, C_c 值为 0.3086~0.4999, 规定 Q 值越大则相应的 C 值越小。通过跟 EPC Gen2 的 Q 算法比较, 得知采用规定步长和快速随机分组的办法可以明显地减少发送 QueryAdjust/Query 指令的次数, 采用帧长内插和快速 Q 值定位的方式可以不同程度的减少发送 QueryAdjust/Query 指令的次数(图 2)。快速 Q 值定位的方式可以有效减少发生碰撞的次数(图 3), 并在一定程度上有效提高了系统的吞吐率(图 4)。采用随机分组快速 Q 值定位的方案融合了二种算法的优势, 可以在 QueryAdjust/Query 指令发送次数、碰撞发生次数和标签识别吞吐率 S 上均表现出良好的性能。

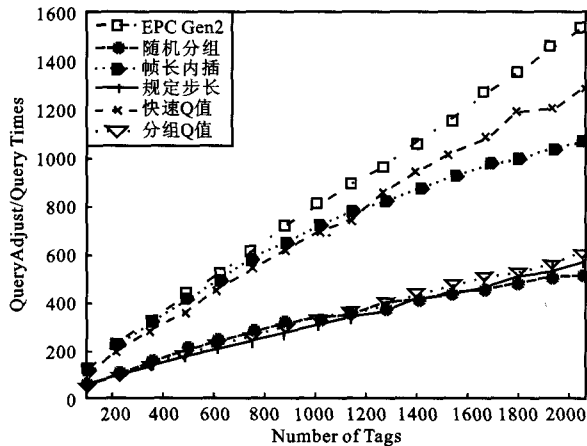


图2 QueryAdjust/Query 指令发送次数与标签数量关系

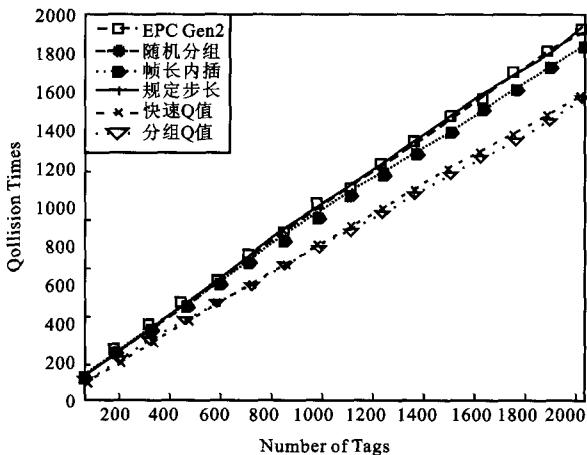


图3 碰撞发生次数与标签数量关系

4 结束语

本文在研究 EPC Gen2 标准防碰撞算法的基础上, 采用随机分组, 内插增加帧长, 规定步长, Q 值快速调整等方式分别进行仿真并分析。仿真结果表明在 128~2048 个的标签范围内, 结合随机分组和 Q

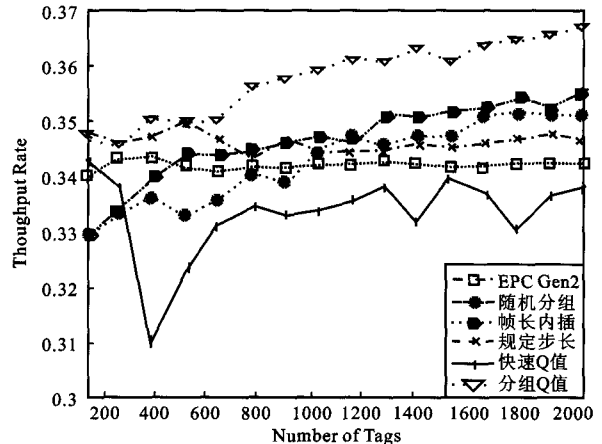


图4 吞吐率与标签数量关系

值快速调整的方式可以在碰撞发生次数, QueryAdjust/Query 指令发送次数以及标签读取吞吐率 S 上均表现出较好的性能。在实现上, 读写器端只需要在 Q 值调整方式做相应的改动, 不需要修改原有的指令和增加额外的指令。标签端只需更改指令解析中的 Q 值解析部分, 不需要增加额外的指令解析开销, 对原有标准的标签改动不大, 并且可以节省一定的寄存器开销。

参考文献:

- [1] EPCglobal. Radio-frequency identity protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz - 960 MHz version 1.2.0 [EB/OL] [2012-07-12]. URL: <http://www.epcglobalinc.org/standards>, 2008.
- [2] KIM S C, S K Kim. An enhanced anti-collision algorithm for EPC Gen2 RFID System[C] // IEEE Computer Society. Washington: IEEE, 2011: 293-296.
- [3] Daneshmand M, Sohraby K, Bo Li. Performance analysis of RFID generation-2 protocol[J]. IEEE Transactions on Wireless Communications, 2009, 8(5): 2592-2601.
- [4] Floerkemeier C, M Wille. Comparison of transmission schemes for framed ALOHA based RFID protocols [C] // International Symposium Applications and the Internet Workshops. Phoenix: IEEE, 2006: 94-97.
- [5] Lee D, K Kim, W Lee. Q+-Algorithm: An enhanced rfid tag collision arbitration algorithm[J]. Ubiquitous Intelligence and Computing, 2007(4611): 23-32.
- [6] Teng Jiafu, Xuan Xiuwei, Bai Yu. A fast Q algorithm based on EPC generation-2 RFID protocol[C] // International Conference on Wireless Communications Networking and Mobile Computing. Chengdu: IEEE, 2010: 1-4.

4 结束语

采用 $0.18\mu\text{m}$ 标准 CMOS 工艺,实现了中心频率为 600MHz、四级差分环形压控振荡器。仿真结果表明,在 1.8V 电源电压,振荡频率为 600MHz 时,该压控振荡器的功耗仅有 4.16mW。最坏的情况,该压控振荡器的频率调整范围为 0.21—1.18GHz;偏离中心频率 10MHz 情况下,压控振荡器的相位噪声为 -118.13dBc/Hz 。本文提出的带对称负载的差分环形振荡器具有低功耗、高增益和小面积的优势,可用于锁相环、频率综合器和时钟产生等电路中。

参考文献:

- [1] Floyd M Gardner, 著,姚剑清,译. 锁相环技术[M](3 版). 北京:人民邮电出版社,2007.
- [2] John G Maneatis, Mark A Horowitz. precise delay generation using coupled oscillators[J]. IEEE Journal Of Solid-State Circuits, 1993, 28(12):1273—1282.
- [3] 桑红石,方海涛. 环形振荡器对称负载特性分析[J]. 微电子学与计算机,2011,28(5): 197—200.
- [4] Behzad Razavi. Design of analog CMOS integrated circuits[M]. New York, USA: McGraw—Hill, 2011.
- [5] 伍翠萍,何波. 一种低电压低功耗的环形压控振荡器设

计[J]. 微电子学与计算机,2008,25(5): 69—72.

- [6] 蔡志明,陈莹梅. GPS 射频接收芯片中低功耗压控振荡器的设计[J]. 微电子学,2009,39(6): 790—792.
- [7] Sun Ling, Tang Lu, Jing Wei—ping, et al. CMOS ring VCO for UHF RFID readers[J]. Journal of China Universities of Posts and Telecommunications, 2010 (3):20—23.
- [8] J K Panigrahi, D P Acharya. Performance analysis and design of wideband CMOS voltage controlled ring oscillator[C]// IEEE 5th International Conference on Industrial and Information Systems Proceedings, ICIIS 2010, India; IEEE, 2010: 234—238.
- [9] Luciano Secerino de Paula, Sergio Bampi, Eric Fabris et al. A high swing low power CMOS differential voltage—controlled ring oscillator[C]// 14th IEEE International Conference on Electronics, Circuits and Systems Proceedings, ICECS 2007. Marrakech: IEEE, 2007, 498—501.

作者简介:

谢连波 男,(1983—),硕士研究生. 研究方向为模拟集成电路设计.

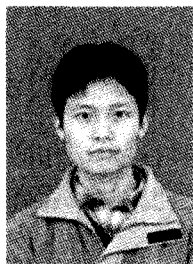
桑红石 女,(1970—),副教授. 研究方向为数字图像处理算法 VLSI 实现与并行处理器体系结构.

(上接第 103 页)

- [7] Cha J R, J H Kim. Novel anti-collision algorithms for fast object identification in RFID system[C] // Inter-

national Conference on Parallel and Distributed Systems. Fukuoka; IEEE, 2005: 63-67.

作者简介:



吴森 男,(1987—),硕士研究生. 研究方向为 UHF RFID 标签数字电路设计.

张钊锋 男,(1972—),博士生导师. 研究方向为物联网核心芯片的研究、柔性快速开发平台的研究、车联网关键技术.