



(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。 Int. Cl. G06K 17/00 (2006.01)	(45) 공고일자 (11) 등록번호 (24) 등록일자	2007년07월11일 10-0738485 2007년07월05일
---	-------------------------------------	--

(21) 출원번호 (22) 출원일자 심사청구일자	10-2006-0057251 2006년06월26일 2006년06월26일	(65) 공개번호 (43) 공개일자
----------------------------------	---	------------------------

(73) 특허권자 제주대학교 산학협력단
 제주특별자치도 제주시 아라1동 1번지 제주대학교

(72) 발명자 김지윤
 제주 제주시 삼양3동 2608번지

 강봉수
 제주 제주시 연동 314-77 연주빌라 301호

 임정현
 제주 제주시 삼도1동 525-22번지

 양두영
 제주 제주시 일도2동 대유대림아파트 308-508

 김성백
 제주 제주시 노형동 731-1 현대아파트 101-1504

 김철민
 제주 제주시 일도2동 113-6번지 우성2차아파트 202-101

 송왕철
 제주 제주시 일도2동 대유대림아파트 306동 302호

 이광만
 제주 제주시 일도2동 113-6번지 우성아파트 201동 403호

(74) 대리인 고길수
 남진우

(56) 선행기술조사문헌 KR1020070013646 A	KR100615704 B1
------------------------------------	----------------

심사관 : 김창주

전체 청구항 수 : 총 3 항

(54) 전파식별 시스템에서의 충돌방지 방법

(57) 요약

본 발명은 RFID 시스템 환경에서 리더가 빠르고 정확하게 물품에 부착된 태그들의 정보를 분별할 수 있게 하는 충돌방지 방법에 관한 것이다. 본 발명은, 스택(Stack)의 개념을 도입하여 충돌이 일어나지 않은 빈(Bin)을 우선 인식하고, 충돌이 일어난 빈(Bin)에 대하여는 충돌이 발생하기 전까지의 비트열을 스택에 저장하고, 하나의 태그를 완전히 인식한 후 스택을 검사하여 저장된 데이터가 있으면 그 값을 [VALUE]와 [LENGTH]에 저장한 후 PingID 명령어를 전송하고 스택에 저장된 데이터가 없으면 'VALUE=0, LENGTH=0'으로 하여 PingID 명령어를 전송함으로써, 태그의 인식 시간을 효과적으로 감소시킬 수 있다.

대표도

도 2

특허청구의 범위

청구항 1.

- (a) 비교할 비트열과 비교할 비트열의 길이에 대한 초기값을 설정하는 단계;
- (b) 리더가 태그의 인식을 위해 태그의 신호를 요청하는 제1 명령어를 공간상의 임의의 태그에 전송하는 단계;
- (c) 상기 리더는 빈 모듈레이션(Bin Modulation) 작업을 하고, 비트 비교열의 시작 위치 이후의 비트열이 빈(Bin)값과 일치하면 태그에서 응답하는 단계;
- (d) 상기 리더는 각 빈(Bin)에서의 태그의 응답 형태를 검사하여 임시 메모리에 저장하는 단계;
- (e) 상기 리더는 빈(Bin)을 검사한 후 빈(Bin)에서의 충돌 여부를 판단하여, 충돌이 발생한 빈(Bin)에 대해서는 각 태그의 응답을 비교하여 충돌이 발생하기 전까지의 비트열을 스택에 저장하고, 충돌이 일어나지 않으면 빈(Bin)값을 비교할 비트열에 포함하고 태그의 식별코드를 요청하는 제2 명령어를 태그에 전송하는 단계;
- (f) 비교할 비트열에 맞는 태그는 상기 제2 명령어에 대한 응답으로 자신의 식별코드를 상기 리더에 전송하는 단계;
- (g) 상기 리더는 CRC 에러를 체크하여 오류가 있으면 상기 제2 명령어를 태그에 다시 전송하고, CRC 에러를 체크하여 오류 없이 전송되었다면 태그의 식별 코드를 저장하는 단계;
- (h) 빈 검사가 완료되었는지를 판단하여 빈 검사가 완료되지 않았다면 (e) 단계로 되돌리고, 빈 검사가 완료되었다면 스택의 메모리를 검사하는 단계; 및
- (i) 스택의 메모리를 검사하여 저장된 데이터가 없으면 (a) 단계로 되돌리고, 저장된 데이터가 있으면 스택에 저장된 데이터를 비교할 비트열과 비교할 비트열의 길이에 저장하고 (b) 단계로 되돌리는 단계를 포함하는 전파식별 시스템에서의 충돌방지 방법.

청구항 2.

제1항에 있어서, 상기 제1 명령어에 대하여 태그는 8비트의 응답을 하고, 상기 리더는 3비트 단위로 빈 모듈레이션 작업을 수행하는 것을 특징으로 하는 전파식별 시스템에서의 충돌방지 방법.

청구항 3.

제1항에 있어서, 상기 리더는 상기 제1 명령어에 대한 태그의 응답 여부를 판단하여 태그의 응답이 없으면 태그에 대한 인식을 종료하는 단계를 더 포함하는 전파식별 시스템에서의 충돌방지 방법.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 전파식별 시스템에서의 충돌방지 방법에 관한 것으로, 더욱 상세하게는 RFID 시스템 환경에서 리더가 빠르고 정확하게 물품에 부착된 태그들의 정보를 분별하여 태그의 인식 시간을 효과적으로 줄일 수 있는 태그 데이터의 충돌방지 방법에 관한 것이다.

최근 많은 연구와 개발이 이루어지고 있는 유비쿼터스 컴퓨팅 분야는 기존 시스템 기능의 고도화를 유도함으로써 시장 구조, 교육 환경 및 경제 활동에 큰 변화를 가져오고 있다. 전파 식별(Radio Frequency Identification; 이하 'RFID'라 함)은 이러한 유비쿼터스 컴퓨팅 환경의 초석이 될 기술로 간주되고 있다.

RFID 시스템은 마이크로칩을 내장한 태그(Tag)에 저장된 데이터를 무선 주파수를 이용하여 리더(Reader)와 송수신하는 시스템이다. 태그는 리더의 요청에 의해 고유의 정보(데이터)를 RF 신호로 변환하여 리더로 전송하며, 리더는 수신된 RF 신호를 적절하게 처리하여 태그에 저장된 데이터를 식별한다. 리더에 의해 태그가 인식되면 데이터 처리 시스템을 통해 태그가 부착된 물체에 대한 데이터 처리가 가능하므로 현재의 바코드 시스템을 대신하여 다양한 분야에서 활용이 가능하게 된다.

이와 관련하여 정보통신부는 IT839 전략을 발표하여 정보기술(Information Technology; 이하 'IT'라 함) 산업 분야에서의 새로운 서비스의 도입과 신규 수요 창출에 역량을 집중하며 신규 서비스로서 RFID/USN(Ubiquitous Sensor Network) 도입을 강력하게 추진하고 있으며, RFID/USN 용 주파수 대역을 908.5~914MHz(5.5MHz)로 확정하였다.

RFID 시스템은 크게 리더(Reader)와 태그(Tag)로 구성된다. 수많은 태그들이 하나의 리더의 인식영역 내에 존재할 수 있다. RFID 시스템에서 리더는 인식영역 내의 태그에 요청 메시지를 전송하고, 요청 메시지를 수신한 태그는 자신의 정보를 리더에 전송한다. 그러나 인식영역 내에 다수의 태그가 존재하여 동시에 요청 메시지에 응답하게 되면 RF(Radio Frequency) 통신 채널 상에 충돌이 발생하게 되고, 결과적으로 리더는 태그의 정보를 정확하게 수신하지 못하게 된다.

이러한 문제를 해결하기 위해 충돌방지 기법이 사용되며, 다중접속 방식의 시스템 성능을 결정짓는 중요한 문제가 된다. RFID 시스템의 성능은 태그를 인식하는데 필요한 시간과 태그가 소모하는 전력으로 결정된다. 태그로부터 전송되는 데이터의 양이 작아지면 태그의 인식시간과 태그의 전력소모를 감소시킬 수 있다. 다중접속 방식을 갖는 RFID 시스템에서 태그 데이터 간의 충돌은 인식시간과 소모 전력의 증가를 가져오는 직접적인 원인이 되므로 효과적인 충돌방지 기법이 요구된다. 충돌방지 기법으로는 이진트리(Binary-tree), 알로하(ALOHA; Additive Links Online Hawaii Area) 방식 등이 있다. 충돌방지 알고리즘의 성능은 RFID 전체 시스템의 특성을 좌우하므로, 성능이 우수한 충돌방지 알고리즘이 요구된다.

발명이 이루고자 하는 기술적 과제

본 발명이 이루고자 하는 기술적 과제는 RFID 시스템 환경에서 리더가 빠르고 정확하게 물품에 부착된 태그들의 정보를 분별하여 태그의 인식 시간을 효과적으로 줄일 수 하는 전파식별 시스템에서의 충돌방지 방법을 제공함에 있다.

발명의 구성

본 발명은, (a) 비교할 비트열과 비교할 비트열의 길이에 대한 초기값을 설정하는 단계와, (b) 리더가 태그의 인식을 위해 태그의 신호를 요청하는 제1 명령어를 공간상의 임의의 태그에 전송하는 단계와, (c) 상기 리더는 빈 모듈레이션(Bin Modulation) 작업을 하고, 비트 비교열의 시작 위치 이후의 비트열이 빈(Bin)값과 일치하면 태그에서 응답하는 단계와, (d) 상기 리더는 각 빈(Bin)에서의 태그의 응답 형태를 검사하여 임시 메모리에 저장하는 단계와, (e) 상기 리더는 빈(Bin)

을 검사한 후 빈(Bin)에서의 충돌 여부를 판단하여, 충돌이 발생한 빈(Bin)에 대해서는 각 태그의 응답을 비교하여 충돌이 발생하기 전까지의 비트열을 스택에 저장하고, 충돌이 일어나지 않으면 빈(Bin)값을 비교할 비트열에 포함하고 태그의 식별코드를 요청하는 제2 명령어를 태그에 전송하는 단계와, (f) 비교할 비트열에 맞는 태그는 상기 제2 명령어에 대한 응답으로 자신의 식별코드를 상기 리더에 전송하는 단계와, (g) 상기 리더는 CRC 에러를 체크하여 오류가 있으면 상기 제2 명령어를 태그에 다시 전송하고, CRC 에러를 체크하여 오류 없이 전송되었다면 태그의 식별 코드를 저장하는 단계와, (h) 빈 검사가 완료되었는지를 판단하여 빈 검사가 완료되지 않았다면 (e) 단계로 되돌리고, 빈 검사가 완료되었다면 스택의 메모리를 검사하는 단계 및 (i) 스택의 메모리를 검사하여 저장된 데이터가 없으면 (a) 단계로 되돌리고, 저장된 데이터가 있으면 스택에 저장된 데이터를 비교할 비트열과 비교할 비트열의 길이에 저장하고 (b) 단계로 되돌리는 단계를 포함하는 전파 식별 시스템에서의 충돌방지 방법을 제공한다.

상기 제1 명령어에 대하여 태그는 8비트의 응답을 하고, 상기 리더는 3비트 단위로 빈 모듈레이션 작업을 수행한다.

상기 리더는 상기 제1 명령어에 대한 태그의 응답 여부를 판단하여 태그의 응답이 없으면 태그에 대한 인식을 종료한다.

이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 실시예를 상세하게 설명하기로 한다. 그러나, 이하의 실시예는 이 기술분야에서 통상적인 지식을 가진 자에게 본 발명이 충분히 이해되도록 제공되는 것으로서 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 다음에 기술되는 실시예에 한정되는 것은 아니다. 도면상에서 동일 부호는 동일한 요소를 지칭한다.

RFID 시스템은 크게 리더(Reader)와 태그(Tag)로 구성된다. 복수의 태그들이 하나의 리더 인식영역 내에 존재하는 경우, 리더는 충돌없이 각 태그를 동시에 인식해야 한다. RFID 시스템에서 리더는 인식영역 내의 태그에 요청 메시지를 전송하고, 요청 메시지를 수신한 태그는 자신의 정보를 리더에 전송한다. 그러나 인식영역 내에 다수의 태그가 존재하여 동시에 요청 메시지에 응답하게 되면 RF(Radio Frequency) 통신 채널 상에 충돌이 발생하게 되고, 결과적으로 리더는 태그의 정보를 정확하게 수신하지 못하게 된다. 이와 같이 각 태그의 출력 신호가 서로 간섭하여 충돌이 발생하는 경우, 리더는 태그에서 보낸 신호를 인식하지 못하게 된다. 본 발명의 바람직한 실시예에서는 각 태그 데이터 간의 충돌을 효과적으로 방지할 수 있는 방법을 제시한다.

리더가 태그로 전송하는 명령어는 8개의 필드와 5개의 패리티비트(Parity Bit) 필드로 구성되고, 그 형식은 다음과 같다.

[PREAMBLE][CLKSYNC][SOF][CMD][P1][PTR][P2][LENGTH][P3][VALUE][P4][P5][EOF]

리더가 태그로 전송하는 명령어 중에서, [PREAMBLE]는 태그에 에너지를 공급하는 부분이다. [CLKSYNC]는 모두 '0'으로 된 20비트로 이루어져 있고, 태그에 클럭 동기를 제공한다. [CMD]는 리더 명령어의 코맨드 코드(Command code)가 들어가는 부분이다. [PTR]은 태그의 비트 비교열의 시작 위치이고, [LENGTH]는 밸류(VALUE)의 길이이다. [VALUE]는 비교할 비트열이다. [SOF]와 [EOF]는 시작과 끝을 알리는 것으로 각각 1비트씩 구성되고, [PTR]과 [LENGTH]는 각각 8비트씩 구성되며, [VALUE]는 상황에 따라 가변적이다. [P1], [P2], [P3], [P4] 및 [P5]는 정보의 전달 과정에서 오류가 생겼는지를 검사하기 위해 원래의 정보에 덧붙이는 패리티 비트 필드이다.

도 1은 리더와 태그 사이에 전송되는 신호의 단계를 설명하기 위하여 도시한 도면이다.

도 1을 참조하면, 리더에서 태그로 전송되는 신호는 후술하는 바와 같이 5개의 단계로 이루어진다.

제1 단계는 '트랜잭션 갭 RF 오프(Transaction Gap RF Off)'에서 시작하고, 트랜잭션 갭 RF 오프(Transaction Gap RF Off) 후에 64 μ s의 지속파(Continuous Wave; 이하 'CW'라 함) 신호(변조되지 않은 신호)가 태그에 전송되는 단계이다.

제2 단계는 태그에 에너지가 공급되는 단계이다. 이는 리더 명령어 부분 중 '[PREAMBLE]' 부분으로서, [PREAMBLE]에 의해 태그에 에너지가 공급되게 된다.

제3 단계는 리더 명령어 패킷(packet)이 태그로 전송되는 단계이다. 이는 '데이터 모듈레이션 윈도우(Data Modulation window)' 부분에 해당된다.

제4 단계는 태그에게 리더 명령어를 해석할 시간을 제공하는 단계이다. 제4 단계는 짧은 '태그 셋업 윈도우(Tag Setup window)'로 태그에게 리더 명령어를 해석할 시간을 제공한다.

제5 단계는 태그 응답이 수신되는 단계이다.

리더 명령어가 전송되는 부분인 '데이터 모듈레이션 윈도우(Data Modulation window)'에서 리더 명령어 패킷의 한 부분인 [CLKSYNC]는 태그에 동기를 제공하고, 클럭 주기 T_0 는 태그의 전송률을 결정한다. 태그는 다음 '트랜잭션 갭(Transaction Gap)'이 시작되기 전까지 리더에서 검출되어야 하고, 프로토콜(Protocol)에서는 '데이터 모듈레이션(Data modulation)'의 [EOF] 다음부터 '트랜잭션 갭(Transaction Gap)' 사이의 시간을 '코스트 인터벌(Coast Interval)'이라 하여 이 시간을 최대 20ms로 규정하고 있다.

리더가 태그로 전송하는 명령어는, 공급 라인에서 태그 인식과정에 사용하는 규정된 명령어(required command)와 공급 라인 이전에 태그에 식별자를 프로그램할 때 사용하는 프로그래밍 명령어(programming command)로 나뉜다. 규정된 명령어에는 ScrollAllID 명령어, ScrollID 명령어, PingID 명령어, Quiet 명령어, Talk 명령어, Kill 명령어가 있다. 프로그래밍 명령어에는 ProgramID 명령어, VerifyID 명령어, LockID 명령어, EraseID 명령어가 있다. 리더 명령어 중에서 태그 인식과정에서 주로 쓰이는 명령어는 ScrollID 명령어와 PingID 명령어이다.

ScrollID 명령어는 태그의 전체 ID를 요청하는 명령어이다. ScrollAllID 명령어는 [VALUE]에 상관없이 태그의 전체 ID를 요청하는 명령어이다. PingID 명령어는 태그 인식과정에서 리더가 태그의 신호를 요구하는 명령어이다. Quiet 명령어는 식별된 태그의 신호는 다른 태그를 인식하는 동안 방해가 되기 때문에 식별된 태그에 잠시 응답하지 않게 휴지(Asleep) 상태가 되도록 요청하는 명령어이다. Talk 명령어는 식별되지 않은 태그에 응답하도록 요청하는 명령어이다. Kill 명령어는 완전히 인식된 태그에 데드(Dead) 상태가 되도록 요청하는 명령어이다. VerifyID 명령어는 모든 메모리 데이터 비트가 정확하게 프로그램된 것을 확인하기 위해서 사용한다.

태그는 리더 명령어를 수신하면 응답을 하거나, 자신의 상태를 변화시킨다. 태그가 리더의 동작 영역으로 들어가게 되어 적절한 파워의 리더 신호를 수신하면 태그는 모든 상태로 변화 가능한 어웨이크(Awake) 상태가 된다. 태그는 ScrollID 명령어, ScrollAllID 명령어, VerifyID 명령어 또는 PingID 명령어를 수신하면 응답(Reply) 상태가 되어 리더에게 응답 신호를 전송한다. 완전히 식별된 태그의 신호는 다른 태그를 인식하는 동안 방해가 되기 때문에 리더는 식별된 태그에 Quiet 명령어를 전송하고 수신한 태그는 잠시 응답하지 않도록 휴지(Asleep) 상태가 되며, 그 반대 과정은 Talk 명령어를 사용한다. 즉, 리더는 식별이 되지 않은 태그에는 Talk 명령어를 전송한다. 완전히 인식된 태그 중에 프라이버시 등의 문제로 제거되어야 할 때, 리더는 그 태그에게 Kill 명령어를 보내고, 태그는 암호화 및 비교하여 일치하면 데드(Dead) 상태가 된다.

태그는 명령어에 따라 수행하는 응답이나 상태가 다르다. 태그가 응답 상태로 되는 리더 명령어는 총 4가지(ScrollID 명령어, ScrollAllID 명령어, VerifyID 명령어, PingID 명령어)로, 그 중에서 태그 인식 과정에서 주로 쓰이는 PingID 명령어에 대한 응답과 ScrollID 명령어에 대한 응답은 다음과 같다.

태그 인식과정에서 태그가 충돌없이 인식되었다면, 리더는 그 태그에 ScrollID 명령어를 전송하고 명령어를 수신한 태그는 그 응답으로 자신의 전체 ID를 리더에 전송하고, 리더는 태그 응답을 받아 CRC(Cyclic Redundancy Checking) 에러(Error)를 체크하여 오류가 없다면 태그의 ID를 저장한다.

PingID 명령어는 태그 인식과정에서 리더가 태그의 신호를 요구하는 명령어이다. 리더는 PingID 명령어를 공간상의 임의의 태그에 전송한 후 3비트 단위의 빈 모듈레이션(Bin modulation) 작업을 한다. PingID 명령어 패킷에서 [VALUE]와 [PTR] 이후 자신의 비트열이 일치하는 태그는 다음의 8비트를 리더로 응답한다. 만약 하나의 빈(Bin)에서 여러 개의 태그 신호가 응답하게 되면 충돌이 발생한 것이고, 하나의 태그 신호만 응답되면 충돌이 일어나지 않은 것이다. 만약, 충돌이 일어나지 않았다면 그 빈(Bin)값을 밸류(VALUE)에 추가하여 ScrollID 명령어를 전송하고, 충돌이 일어났다면 그 빈(Bin)값을 밸류(VALUE)에 추가하여 다시 PingID 명령어를 전송한다. ScrollID 명령어를 수신한 태그는 그 응답으로 자신의 전체 ID를 리더에 전송하고, 리더는 태그 응답을 받아 CRC 에러를 체크하여 오류가 없다면 태그의 ID를 저장하게 된다.

본 발명의 바람직한 실시예에 따른 전파식별 시스템에서의 충돌방지 방법은 EPC(Electric Product Code) 클래스 1 프로토콜(Class 1 Protocol)에서 정의된 규정을 만족하면서 이를 더 개선시킨 충돌방지 방법이다.

클래스 1 프로토콜의 알고리즘은 빈 트리(Bin tree)에 기반을 두고, 이는 3비트 단위로 확장하는 형태이다. 클래스 1 프로토콜의 리더에서 태그를 인식하는 과정은, 리더가 'VALUE=0, LENGTH=0'을 포함하는 PingID 명령어를 공간상의 임의

의 태그에 전송함으로써 시작된다. PingID 명령어를 전송한 후 리더는 빈 모듈레이션(Bin Modulation) 작업을 하고, 태그 자신의 [PTR] 이후 비트열이 빈(Bin)값과 일치하면 빈(Bin)에서 8비트의 응답을 한다. 여기서, [PTR]은 태그의 비트 비교열의 시작 위치이다. 빈(Bin)을 검사한 후 각 빈(Bin)에서 충돌 여부를 판단한다.

빈(Bin)에서 충돌이 일어나지 않으면, 리더는 빈(Bin)값을 [VALUE]에 포함한 후 ScrollID 명령어를 태그에 보내게 된다. [VALUE]에 맞는 태그는 자신의 전체 ID를 ScrollID 명령어에 대한 응답을 리더에 보내게 되고, 리더는 CRC 에러를 체크하여 오류 없이 전송되었다면 태그 식별 코드를 저장하고 과정을 처음으로 되돌린다.

만약, 빈(Bin)에서 충돌이 일어나면 리더는 빈(Bin)값을 [VALUE]에 저장한 후 다시 PingID 명령어를 공간에 전송한다. 여기서, 빈(Bin) 검사는 빈(Bin)의 처음 시작 위치로 돌아가 원위치에서부터 다시 시작한다.

본 발명의 바람직한 실시예에 따른 전파식별 시스템에서의 충돌방지 방법은 클래스 1 프로토콜의 알고리즘에서 스택(Stack)의 개념을 도입하여 충돌이 일어나지 않은 빈(Bin)을 우선 인식하고, 충돌이 일어난 빈(Bin)은 스택(Stack)에 저장하는 방법으로 구현한다. 이는 PingID 명령어에 대한 태그의 응답이 8비트인 점을 고려한 것이다. PingID 명령어에 대한 태그의 응답이 8비트인 점에 착안하여 충돌이 발생한 Bin에서 입력되는 여러 태그들의 8비트열을 리더에서 검사하여 비교한 후 일치하는 비트열까지 스택에 저장한다. 하나의 태그를 완전하게 인식한 후, 다시 'VALUE=0, LENGTH=0'으로 하여 PingID 명령어를 보내는 것이 아니라 우선 스택을 검사하여 저장된 데이터가 있으면 그 값을 [VALUE]와 [LENGTH]에 저장한 후 PingID 명령어를 보내고, 스택에 저장된 데이터가 없으면 다시 'VALUE=0, LENGTH=0'으로 하여 PingID 명령어를 보내는 형태이다.

이하에서, 본 발명의 바람직한 실시예에 따른 전파식별 시스템에서의 충돌방지 방법을 더욱 상세히 설명한다.

도 2를 참조하면, 태그 인식과정이 시작되면, 비교할 비트열([VALUE])과 비교할 비트열의 길이([LENGTH])에 대한 초기값(VALUE=0, LENGTH=0)을 설정하고(S100), 리더는 공간상의 임의의 태그에 'VALUE=0, LENGTH=0'을 포함하는 PingID 명령어를 전송하여 모든 태그들의 신호를 요청한다(S102). 리더는 PingID 명령어를 전송한 후 빈 모듈레이션 작업을 하고, 태그는 [PTR](비트 비교열의 시작 위치) 이후 비트열이 빈(Bin)값과 일치하면 빈(Bin)에서 8비트의 응답을 한다(S104).

PingID 명령어에 대한 태그의 응답 여부를 판단하여(S106) 응답이 없으면 종료하고, 태그에서 응답이 있으면 리더는 각 빈(Bin)에서의 태그의 응답 형태를 검사하여 임시 메모리에 저장한다(S108).

리더는 빈(Bin) 검사를 하여(S112) 충돌이 있는지 여부를 판단하고(S114), 충돌이 발생한 빈(Bin)에 대해서는 각 태그의 8비트 응답을 비교하여 일치하는 비트열까지 즉, 충돌이 발생하기 전까지의 비트열을 스택에 저장한다(S116). 하나의 빈(Bin)에서 여러 개의 태그 신호가 응답하게 되면 충돌이 발생한 것이고, 하나의 태그 신호만 응답하게 되면 충돌이 일어나지 않은 것이다. 충돌이 있는지 여부를 판단하여(S114), 충돌이 없는 경우 충돌이 없는 빈(Bin)에 대하여 그 빈(Bin)값을 [VALUE]에 포함하고(S118), ScrollID 명령어를 태그에 보내 태그를 인식한다(S120). [VALUE]에 맞는 태그는 자신의 전체 식별코드(ID)를 ScrollID 명령어에 대한 응답으로 리더에 보내게 되고(S122), 리더는 CRC 에러를 체크하여(S124) 오류(Error)가 있는지 여부를 판단한다(S126).

CRC 에러를 체크하여 오류가 있는 경우에는 S120 단계로 되돌려서, 다시 ScrollID 명령어를 전송하고(S120), ScrollID 명령어에 대한 응답으로 태그의 전체 ID를 전송받고(S122) CRC 에러를 체크하여(S124) 오류가 있는지 여부를 판단한다(S126). 오류 없이 전송되었다면 태그의 식별 코드(ID)를 저장한다(S128).

태그의 식별 코드(ID)를 저장하는 단계(S126) 후에 또는 빈 검사를 하여 충돌이 발생한 경우로서 각 태그의 8비트 응답을 비교하여 충돌이 발생하기 전까지의 비트열을 스택에 저장하는 단계(S116) 후에는, 8개의 빈(Bin) 검사를 완료하여 하나의 태그가 완전히 인식되었는지를 판단하고(S130), 8개의 빈(Bin) 검사가 완료되지 않았다면 빈(Bin) 검사 단계(S112)로 되돌린다.

하나의 태그가 완전히 인식되면, 리더는 'VALUE=0, LENGTH=0'로 하여 PingID 명령어를 전송하지 않고, 스택의 메모리를 검사하여(S132) 스택에 저장된 데이터가 있는지 여부를 판단한다(S134).

스택에 저장된 데이터가 있으면, 리더는 스택에 저장된 데이터를 [VALUE](비교할 비트열)와 [LENGTH](비교할 비트열의 길이)에 저장하고(S136), PingID 명령어를 전송하는 단계(S102)로 되돌려서 상기 S102 단계부터 이후의 과정을 반복한다. 스택에 저장된 데이터가 없으면, 'VALUE=0, LENGTH=0'으로 초기값을 설정하는 S100 단계로 되돌려서 S100 단계부터 이후의 과정을 반복한다.

도 3은 EPC 클래스 1 프로토콜에 따른 알고리즘과 본 발명의 바람직한 실시예에 따른 전파식별 시스템에서의 충돌방지 방법에 대한 태그 인식 시간 특성을 비교한 그래프이다. 도 3에서 (a)는 본 발명의 바람직한 실시예에 따른 전파식별 시스템에서의 충돌방지 방법에 대한 그래프이고, (b)는 EPC 클래스 1 프로토콜에 따른 알고리즘에 대한 그래프이다. 도 3에서 총 태그 인식 시간은 도 1을 참고하였고, T_0 는 북아메리카에서 규정된 $14.25\mu s$ 로 계산하였다.

도 1 및 도 3을 참조하면, 리더에서 태그로 전송하는 신호는 첫 번째 단계인 트랜잭션 갭(transaction gap)이 $1.25T_0$, 두 번째 단계인 CW RF신호는 $64\mu s$, 세 번째 단계인 데이터 모듈레이션(data modulation) 부분, 네 번째 단계는 태그 셋업(Tag Setup) 단계로 $8T_0$, 마지막 단계인 태그 응답 전송부분은 $64T_0$, 다음 트랜잭션(Transaction gap)과의 인터벌(interval)을 $2.5T_0$ 로 하여 계산하였다. 세 번째 단계인 데이터 모듈레이션(data modulation) 부분에서 [VALUE]를 제외한 나머지는 총 51비트로 구성된다. 51비트에 아래의 표 1에서 시뮬레이션 결과로 구한 VLP와 VLS를 포함하여 데이터 모듈레이션 부분의 시간계산을 산출하였다. VLP(Value Length per PingID command)는 PingID 명령어 당 [VALUE]의 길이이고, VLS(Value Length per ScrollID command)는 ScrollID 명령어 당 [VALUE]의 길이이다.

[표 1]

태그의 수	EPC 클래스 1 프로토콜 알고리즘			본 발명의 충돌방지 방법		
	총 명령어 수	VLP	VLS	총 명령어 수	VLP	VLS
10	38.2	2.18	5.46	24.8	3.63	6.64
50	224.6	2.68	7.48	121	5.79	8.93
100	485.4	3.11	8.56	242.4	6.94	10.03
200	1034.6	3.59	9.52	487.6	8.14	11.18
500	2810	4.22	10.86	1222	9.54	12.58
1000	5975	4.75	11.93	2435.2	10.45	13.51

리더에서 태그로 전송된 명령어 수 중에서 PingID 명령어 수는 빈 모듈레이션(Bin Modulation) 수와 동일하고, ScrollID 명령어와 Quiet 명령어 수는 인식한 태그 수와 같다. 하나의 PingID 명령어에 대한 명령 패킷의 전송과 그 응답 시간을 $T_{p1cycle}$, 하나의 ScrollID 명령어에 대한 명령 패킷의 전송과 그 응답 시간을 $T_{s1cycle}$, 하나의 Quiet 명령어에 대한 명령 패킷의 전송과 그 응답 시간을 $T_{q1cycle}$ 이라 하고, 전체 태그 응답 시간을 T_{total} 이라 하면, 이를 구하는 식은 다음과 같다.

수학식 1

$$T_{p1cycle} = 1.25T_0 + 64\mu s + 51T_0 + [VLP]T_0 + 8T_0 + 64T_0 + 2.5T_0$$

수학식 2

$$T_{s1cycle} = 1.25T_0 + 64\mu s + 51T_0 + [VLS]T_0 + 8T_0 + 64T_0 + 2.5T_0$$

수학식 3

$$T_{q1cycle} = 1.25T_0 + 64\mu s + 51T_0 + [VLS]T_0 + 8T_0 + 2.5T_0$$

수학식 4

$$T_{total} = (\text{PingID 명령어 수} \times T_{p1cycle}) + (\text{ScrollID 명령어 수} \times T_{s1cycle}) + (\text{Quiet 명령어 수} \times T_{q1cycle})$$

총 태그 인식 시간을 계산해 본 결과, 본 발명의 바람직한 실시예에 따른 충돌방지 방법은 태그 수가 100개일 때는 EPC(Electric Product Code) 클래스 1 프로토콜의 알고리즘보다 54.5%의 감소율을 보이고, 태그 수가 1000개일 때는 63.4%의 인식 시간이 감소됨을 알 수 있다. 태그 수가 증가할수록 더욱 더 클래스 1 표준보다 인식 시간이 감소됨을 알 수 있다. 이와 같이 본 발명의 바람직한 실시예에 따른 충돌방지 방법은, RFID 시스템 환경에서 리더가 빠르고 정확하게 물품에 부착된 태그들의 정보를 분별할 수 있게 한다.

발명의 효과

상기한 바와 같이 본 발명은 RFID 시스템에서 하나의 리더 영역에 다중 태그의 접속으로 인한 충돌 문제를 효과적으로 대체할 수 있고, 다중 태그의 인식시간을 효과적으로 감소시킬 수 있다. 또한, EPC(Electric Product Code) 프로토콜을 만족하게 구현을 하였으므로, 현재 판매되는 태그나 리더에 적용하여 사용할 수 있다. 실제 환경에서 적용 가능하도록 EPC 프로토콜의 규정을 따르면서 이진트리(binary-tree)와 슬롯 알로하(Slotted ALOHA) 프로시저의 특성을 결합한 빈-슬롯 하이브리드 검색(Bin-slotted hybrid search) 기법으로 다중 태그 인식시간을 효과적으로 감소시킬 수 있다.

이상, 본 발명의 바람직한 실시예를 들어 상세하게 설명하였으나, 본 발명은 상기 실시예에 한정되는 것은 아니며, 본 발명의 기술적 사상의 범위내에서 당 분야에서 통상의 지식을 가진 자에 의하여 여러 가지 변형이 가능하다.

도면의 간단한 설명

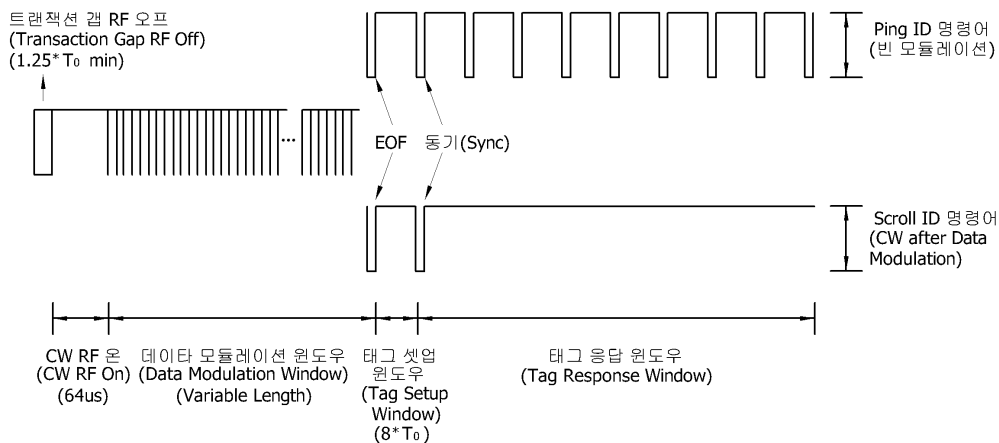
도 1은 리더와 태그 사이에 전송되는 신호의 단계를 설명하기 위하여 도시한 도면이다.

도 2는 본 발명의 바람직한 실시예에 따른 전파식별 시스템에서의 충돌방지 방법을 설명하기 위하여 도시한 순서도이다.

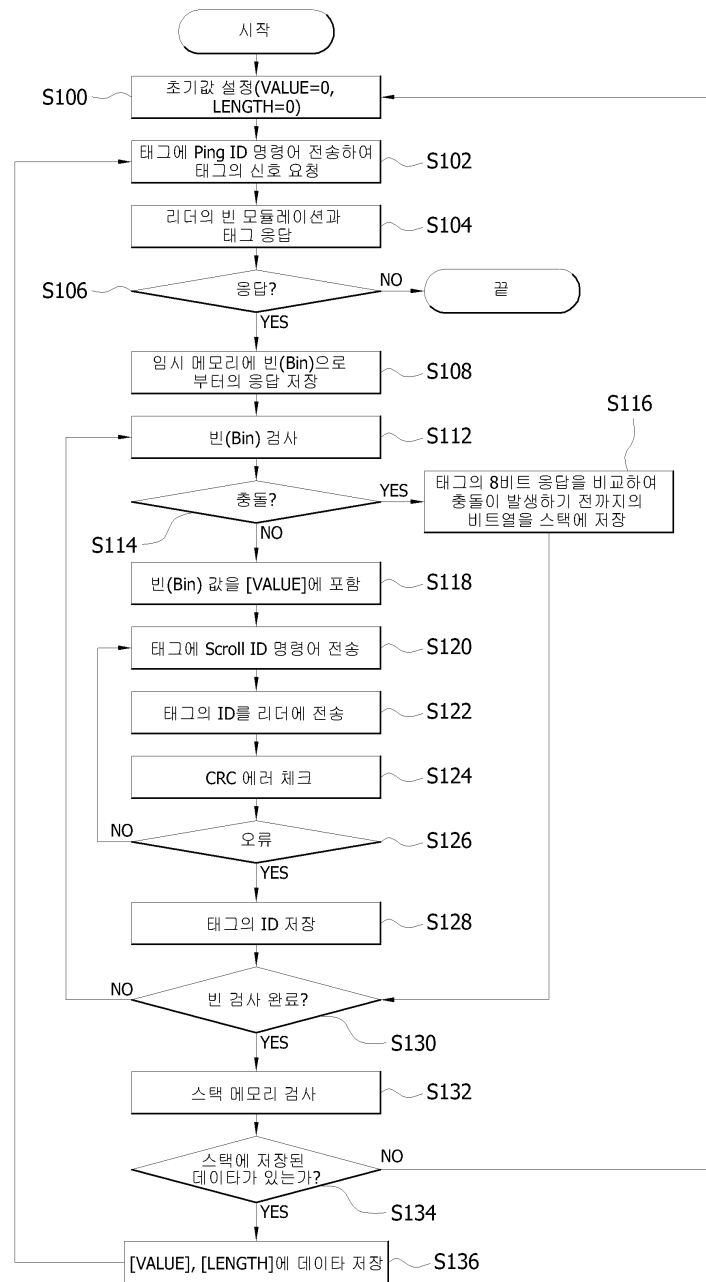
도 3은 EPC 클래스 1 프로토콜에 따른 알고리즘과 본 발명의 바람직한 실시예에 따른 충돌방지 방법에 대한 태그 인식 시간 특성을 비교한 그래프이다.

도면

도면1



도면2



도면3

