

# RFID 시스템에서 Orthogonal Code를 사용한 태그 인식 메커니즘

김성준<sup>○</sup> 정영환 안순신  
고려대학교

{sjunil<sup>○</sup>, youngh, sunshin}@dsys.korea.ac.kr

## Tag-Identification Mechanism using Orthogonal Code in RFID System

Sungjun Kim<sup>○</sup> Younghwan Jung Sunshin Ahn

Computer Network Lab. Dept. of Electronics and Computer Eng., Korea University

### 요약

본 논문에서는 별도의 충돌 방지 메커니즘이 필요 없는 orthogonal code를 태그 ID로 부여함으로써 태그 인식을 간단하게 하는 메커니즘에 대해 설명한다. 첫 번째 적용 메커니즘으로 Local환경에서 Orthogonal code가 적용될 수 있는 여러 가지 응용 분야를 제안하고 두 번째로 Orthogonal code를 기존 ID 체계의 일부분으로 사용하여 anti-collision algorithm의 효율을 높이는 방법을 제시한다. 마지막으로 orthogonal code ID의 개수가 많아짐에 따라 code bit수가 길어지는 문제를 해결하기 위하여 code를 블록화 하여 사용 bit를 줄이는 메커니즘을 제시한다.

### 1. 서론

최근 주목을 받고 있는 RFID 기술은 RF 신호를 사용하여 물품에 부착된 전자태그를 인식하는 기술로서 물류 산업의 지속적인 성장과 태그의 저가격화에 힘입어 다양한 분야에서 필요한 기술로 전망되고 있다.

또한, RFID 기술은 USN(Ubiquitous Sensor Network)의 구축에 대한 실제적인 솔루션을 제공하고 언제 어디서나 제한 없는 접속이 이루어지는 유비쿼터스 환경 구축에 있어 없어서는 안될 중요 기술로 주목받고 있다. 이러한 RFID 기술은 유통/물류뿐만 아니라 의료, 약품, 환경 등의 우리 주변의 거의 모든 분야에 적용 가능하다[1].

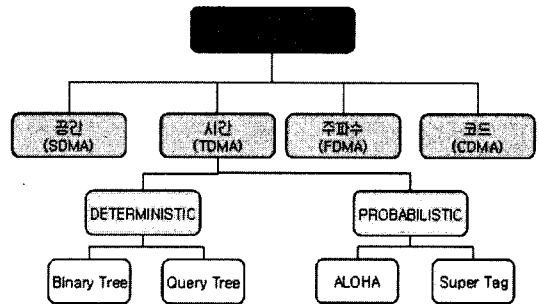
RFID 기술의 중요성에 대한 인식이 전 세계적으로 확산됨에 따라 RFID 기술 확보 및 국가 경쟁력 확보를 위해 많은 국가들에서는 막대한 투자와 연구가 진행 중이다. 이러한 RFID 기술의 적용과 확산을 위해서는 태그의 저가격, 저전력, 초소형화 문제, 보안 및 프라이버시 문제, 여러 개의 태그 식별 문제 등의 해결해야 할 사항이 많이 남아있다. 그중 여러 태그의 식별 문제는 anti-collision algorithm으로 해결할 수 있다.

### 2. 관련 연구

#### 2.1 Anti-collision Algorithm

하나의 RFID 리더의 판독 범위 내에 여러 개의 태그가 있을 때, 리더의 요청 시 여러 개의 태그는 동시에 응답하게 된다. 이때 여러 태그들의 신호는 서로 간섭을 일으켜 collision이 일어나게 된다. 이러한 태그의 응답 신호간의 간섭을 해결하여 각 태그의 응답을 충돌 없이 인식하도록 하는 방법이 필요하게 되는데 이것이 anti-collision algorithm이다.

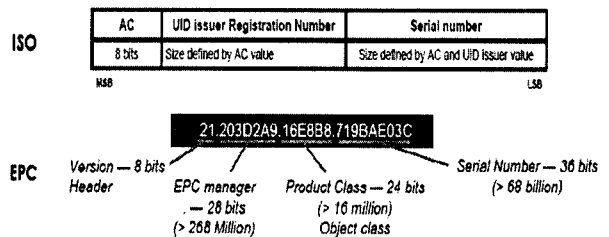
현재, 다양한 anti-collision algorithm들이 제시되고 있으며 이러한 알고리즘은 크게 deterministic 방법과 probabilistic 방법으로 나뉘며 가장 많이 쓰이고 있는 방식은 ALOHA 방식과 Binary Tree 방식이 있다. Anti-collision 방식은 다음 그림1과 같이 분류 될 수 있다[2].



[그림 1] Anti-collision algorithm의 분류

#### 2.2 기존 ID 부여 방식 및 문제점

기존의 Tag ID 부여 방식은 다음과 같다. ISOLA EPC Global과 같은 표준화 단체에서는 그림2와 같이 태그의 ID를 계층적인 구조로 구성하고 binary value를 ID로 부여하고 있다. 태그에서는 이러한 binary value를 FM0, Manchester encoding 방법과 같은 line coding 방법을 사용하여 [3,4,5] 전송하게 되는데 이때, 여러 태그들의 응답 RF신호의 상호 간섭으로 인하여 충돌이 발생하여 식별할 수 없는 신호가 수신된다.



[그림 2] ISO, EPC의 ID 체계

이러한 신호의 충돌을 방지하고 ID를 제대로 인식하게 하기 위해서는 특별한 anti-collision 메커니즘이 필요하게 되는데 기존 메커니즘은 ID 인식 속도가 상당히 느리다. 그렇기 때문에 anti-collision mechanism이 필요 없이 빠르게 ID를 인식할 수 있는 방법이 필요하다. 본 논문에서는 anti-collision mechanism이 필요 없는 orthogonal code를 사용하는 방법에 대해 살펴볼 것이다.

### 3. Orthogonal Code의 원리 및 예시

#### 3.1 Orthogonal Code의 특징

Orthogonal이란 성질은 두 신호 간 상관성이 전혀 없어 서로에게 어떤 영향도 끼치지 않는 것을 얘기한다. 즉 두 신호가 orthogonal하다는 말은 각자의 독립성이 지켜진다는 것이다. Orthogonal Code는 각 코드가 동시에 송신될 때, 서로에게 간섭을 일으키지 않고 자신의 정보를 유지할 수 있다. 수신 단에서는 이러한 orthogonal code가 여러 개가 동시에 들어오더라도 orthogonal 성질을 이용하여 각각 분리가 가능하다[6].

#### 3.2 Tag의 ID로 Orthogonal Code의 이용

기존의 EPC, ISO에서 사용하는 binary value가 아닌 orthogonal code를 별도의 관리 기관이 부여 (예시: local환경에서는 local 환경 관리자가)하고 태그는 리더가 request하면 자신의 ID (orthogonal code)를 특별한 코딩 없이 전송한다.

여러 태그의 응답들은 각각 orthogonal하기 때문에 서로 충돌 없이, 정보의 손실 없이 독립적으로 전달이 되어 리더에서는 각 응답을 분리할 수 있다.

리더는 단 한번의 request로 여러 tag의 ID를 인식 가능하기 때문에 특별한 anti-collision algorithm이 필요하지 않고 매우 빠르게 인식할 수 있다.

#### 3.3 Orthogonal Code의 동작 원리 및 예시

##### 3.3.1 RFID 에서의 Orthogonal Code의 이용

RFID에서는 CDMA에서 사용되는 orthogonal code와 유사하지만 메시지가 없고 자신의 ID (orthogonal code)만을 전송하여 ID를 인식한다. ID를 전송할 때는 별도의 line coding은 하지 않고 modulation만을 하여 전송해준다.

그림 3과 같이 리더의 요청에 태그는 자신의 ID  $c_n(t)$ 를 전송하고 여러 태그가 동시에 응답하게 되면 리더에서 수신된 신호는  $c_1(t) + c_2(t) + c_3(t)$ 이 된다.

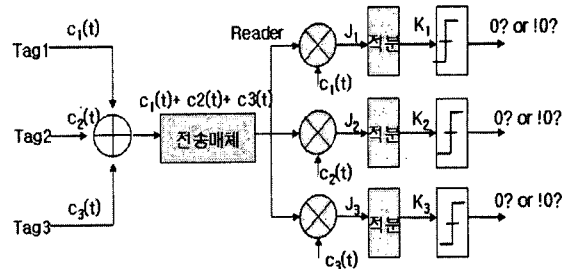
수신된 신호를 가능한 모든 code와 각각 곱하여 한 주기 동안 적분하거나 더하여 나온 수  $K_n$ 을 이용하여 태그의 존재 여부를 판단하는데 orthogonal code의 특성상  $K_n$ 이 0이면 해당 태그는 존재하지 않고, 0이 아닌 값이 나오면 해당 태그는 존재한다. 예를 들어 수신된 신호  $c_1(t) + c_2(t) + c_3(t)$ 와 응답하지 않은 태그의 코드  $c_4(t)$ 를 곱하면 0이 나온다[7].

### 4. Orthogonal Code의 적용 메커니즘

#### 4.1 Orthogonal Code의 Local 환경에서의 적용

Orthogonal code는 비트 수에 따라 나올 수 있는 경우의 수가 한정되어 있기 때문에 소량의 Item이 취급되는 Local 환경 (enterprise level)에 적용될 수 있다.

Local 환경에서 orthogonal code로 ID를 부여하게 되면 Local 환경 내에서 사용되는 리더는 anti-collision algorithm이



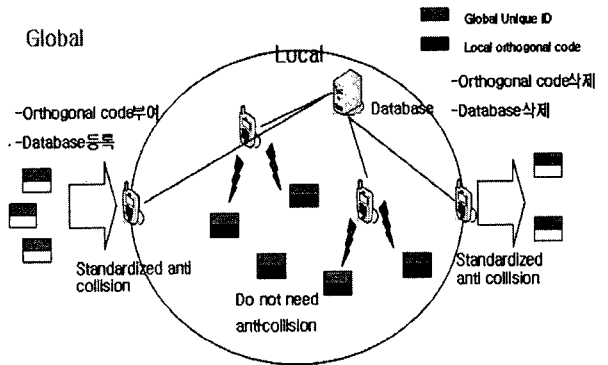
[그림 3] Orthogonal Code의 동작 원리

필요치 않아 간단하고 값 싸게 만들 수 있으며, 인식 속도 또한 빨라진다.

Global한 환경에 있던 Item이 Local한 상황으로 편입될 때 orthogonal code를 부여하고 database에 등록한다. 이때 Item을 인식하는 리더는 기존의 표준화된 충돌 방지 메커니즘을 이용하여 인식한다.

Local 환경 내에서 Item을 인식할 때는 충돌 방지 메커니즘 없이 orthogonal code를 읽어 들여 등록된 database를 이용하여 사용, 관리한다. 이 때 Item을 인식하는 리더는 orthogonal code를 인식할 수 있는 리더를 이용해야 한다.

Item이 local환경에서 벗어날 때에는 item의 orthogonal code ID를 삭제하고 database에서 해당 ID를 삭제하여 orthogonal code의 재사용이 가능하도록 한다. 그림 4는 이러한 메커니즘을 설명하고 있다.



[그림 4] Orthogonal Code의 local 환경에서의 적용

##### Local Orthogonal ID의 부여

Tag가 Local환경으로 편입될 때 Global환경과 Local환경의 경계에 있는 리더는 database에서 미리 생성된 local orthogonal ID 중 사용되지 않는 ID를 할당받아 새로운 태그의 메모리에 저장한다. Database는 사용되지 않는 orthogonal ID, 사용되는 orthogonal ID의 목록을 가지고 있으며 사용되고 있는 orthogonal ID 목록에는 orthogonal ID와 global ID의 mapping 정보를 가지고 있다.

##### Local 환경에서 ID의 인식

Local 환경 내에서 리더는 orthogonal ID를 요청하고 태그들은 자신이 부여 받은 orthogonal ID를 응답한다. 리더는 DB에 등록되어 사용되고 있는 orthogonal code를 이용하여 수신된 신호를 분석하여 수신신호 내에 들어있는 ID를 분리해 낸다.

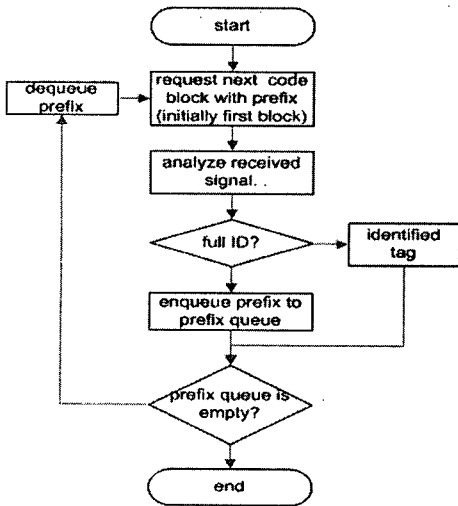
4.2 Orthogonal Code의 블록화

보통의 orthogonal code는 개수가 늘어나면 비트수가 늘어난다. 보통 n개의 orthogonal code를 사용하려면 n비트가 필요하다. orthogonal code를 블록화 하여 생성하면 필요한 비트수가 줄어든다.

블록화를 하지 않고 nk개의 orthogonal code를 만들려면 보통 nk 비트가 필요하지만 n비트의 orthogonal code를 사용하여 k개의 블록으로 만들면 nk개의 orthogonal code가 만들어진다.(ID의 길이는 k\*n 비트) 각 블록의 code는 서로 orthogonal 하다. database에 등록된 orthogonal code를 참조하여 사용하면 더 빠른 인식이 가능해진다

4.2.1 블록화된 orthogonal code의 인식

그림 5는 블록화된 orthogonal code를 인식하는 과정을 보여주고 있다. 가장 먼저 리더는 prefix와 함께 orthogonal code의 블록을 요청한다. 태그는 prefix가 일치하면 요청된 code의 블록을 전송하고 리더는 수신된 신호에서 블록화된 code에서 orthogonal code를 분리해 낸다. 분리해낸 orthogonal code들을 앞서 전송한 prefix와 합쳐 새로운 prefix 혹은 full ID를 생성한다. Full ID일 경우 태그는 인식된 것이고 prefix일 경우는 prefix 큐에 넣는다. prefix 큐가 빌 때까지 반복하면 모든 태그가 인식된다. 이러한 과정으로 블록화된 orthogonal code를 인식할 수 있게 된다.



[그림 5] 블록화된 ID의 인식 과정

4.3 Orthogonal Code를 Tag ID의 일부분으로 사용

마지막으로, orthogonal code를 사용한 메커니즘은 orthogonal code를 Tag의 일부분으로 사용한다. Orthogonal code는 anti-collision algorithm이 필요 없다는 점에서 매우 빠른 태그의 인식을 가능하게 하나 ID의 길이의 비효율성을 가지기 때문에, 이를 태그 ID의 일부분으로 사용하여 anti-collision algorithm의 효율을 높일 수 있는 다양한 적용이 가능하다.

4.3.1 적용 예시

간단한 예로 태그 ID의 앞부분만을 orthogonal code로 사용

하는 메커니즘이 있다. 이 메커니즘은 한번의 request로 orthogonal code 블록을 읽어 들일 수 있다. 다음에 읽어 들인 orthogonal code의 block은 anti-collision algorithm의 적용 전 태그의 그룹화를 할 수 있어 anti-collision algorithm의 효율을 높일 수 있다.

또 다른 적용 예로서 태그의 모든 정보가 필요 없고 특수한 block만을 필요로 할 때 사용할 수 있는 방법이 있다. 예를 들어 생산자의 정보를 읽어 들이기 위해 생산자의 정보가 든 code block을 orthogonal code로 할당하여 빠르게 읽어 들이는 방법이다. 또한 이 방법은 serial number만을 주로 읽어 들일 때는 serial number 부분을 orthogonal code로 할당하여 빠르게 읽어 들일 수 있다.

그 밖에 충돌이 일어난 비트들만을 orthogonal code로 mapping하여 전송하는 방법이 있다. 일반적인 anti-collision algorithm을 이용하였을 때, 수신된 신호를 분석하여 충돌이 일어난 위치의 비트만을 orthogonal code로 대체하여 충돌 없이 전송하는 방법을 사용하면 좀 더 빠른 태그의 인식이 가능해진다.

5. 결론

기존의 일반적인 binary value ID를 사용하는 것과 다르게 orthogonal code를 태그의 ID 혹은 ID의 일부분으로 사용하는 방법은 기존 anti-collision 방법과는 전혀 다른 접근 방법이다. Orthogonal code를 Tag의 ID로 사용하면 단 한번의 request로 모든 태그들을 동시에 읽어 들일 수 있어 anti-collision algorithm이 불필요하게 되고 획기적인 성능향상을 가져온다.

본 논문에서 제시한 local orthogonal ID부여 방법은 Local 환경에서 global하게 부여된 길고 비효율적인 ID를 사용하지 않고, 짧고 효율적인 ID를 다시 부여하여 태그의 인식속도를 향상 시킬 수 있고, 개체 관리의 효율성을 높일 수 있다.

Orthogonal code를 블록화 하거나 태그의 ID의 일부분으로 사용하는 방법은 기존 anti-collision algorithm의 효율성을 높일 수 있다.

본 논문에서 제시한 orthogonal code를 사용한 메커니즘은 Orthogonal ID를 부여할 수 있는 태그와 이를 읽을 수 있는 리더를 새로 개발해야 한다는 단점이 있지만 효율적인 물류관리의 측면에서는 매우 필요한 메커니즘이 될 것이다.

6. 참고 문헌

- [1] S.E. Sarma. "Towards the five-cent tag". Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001. <http://www.autoidcenter.org>
- [2] K. Finkenzeller, "RFID Handbook". John Wiley & Sons. 1999.
- [3] Sanjay E. Sarma, Stephen A. Weis, Daniel W. Engels, "RFID Systems, Security&Pprivacy Implications", Technical Report, MIT-AUTOID
- [4] "EPC class 1". EPC global standard documents <http://www.epcglobalinc.org/standards\_technology/specifications.html>
- [5] "ISO/IEC FDIS 18000-6". ISO standard document, <http://www.iso.org>
- [6] Garg, Vijay Kumar, "Wireless Network Evolution", Prentice Hall, 2002
- [7] Swarts, Francis, "CDMA techniques for third generation mobile systems", Kluwer Academic Publishers, 1999