FPGA를 이용한 RFID Gen2 protocol의 구현 및 검증

제영대 · 김재림 · 장일수 · 양훈기

광운대학교

Implementation & Verification of RFID Gen2 Protocol on FPGA Prototyping board

Young-Dai Je · Jae-Lim Kim · Il-Su Jang · Hoon-Gee Yang

*Kwangwoon University

E-mail : shine702@kw.ac.kr

요 약

본 논문에서는 EPCglobal Class1 Gen2 Protocol의 UHF(Ultra High Frequency) 대역 수동형 태그의 디지털 프로세서를 VHDL로 설계하고, FPGA를 사용한 프로토타이핑 보드를 통해 구현하여 수동형 태그의 인식률을 제고하여 본다. 제한된 인식거리로 인해 다수의 태그들을 Frame-Slotted Aloha 방식으로 인식해야하는 UHF 대역의 RFID(Radio Frequency IDentification) 환경에서, 태그들의 효율적인 응답 및 상태 천이가 가능하도록 구현하였다. 또한 C로 설계한 리더와의 UART 통신을 통하여 Gen2 Protocol의 Inventory Round가 원할히 동작함을 검증 하였고, 640Kbps 태그 응답 속도에서 약 1.4 ms만에 96bit EPC코드를 리더에게 전송하는 일련의 과정을 확인 할 수 있었다.

ABSTRACT

This paper presents the VHDL implementation procedure of the passive RFID tag in Ultra High Frequency RFID system. The operation of the tag compatible with the EPCglobal Class1 Generation2(GEN2) protocol is verified by timing simulation after synthesis and implementation on prototyping board. Due to the reading range with relatively large distance, a passive tag needs digital processor which facilitates faster decoding, encoding and state transition for enhancement of the interrogation rate. Also with UART communication, verify a Inventory Round in Gen2 Protocol. The verification results with the fastest data rate, 640Kbps, and multi tags environment scenario show that the implemented tag spend 1.4ms transmitting the 96bits EPC to reader.

키워드

RFID, FPGA, Prototyping board, EPC, Generation2

Ⅰ. 서 론

EPCglobal의 Gen2(ISO 18000-6C)는 단일 국제 표준 프로토콜로써 종전의 Gen1에서 제공되던 단일 리더 모드를 포함한 다중 리더 모드와 밀집 리더 모드를 지원한다. 이러한 여러 모드로 인하여 제한된 범위 내에 존재하는 여러 개의 리더가 충돌 없이 태그를 식별할 수 있다. 각 태그는 4개의 세션에서 동작 할 수 있으며 각 리더는 특정 세션의 태그만을 선택하여 식별함으로써 리더간

"본 연구결과물은 「2005 산업자원부 성장동력산업」의 「유비쿼터스 홈 네트워크 범용 침입대응 시스템 개발사업」의 지원과 정보통신부 및 정보통신연구진흥원의 「대학 IT연구센터 지원사업」(IITA-2008-C1090-0801-0038) 지원으로 수행 되었음"

의 충돌을 방지한다. 태그 메모리는 필드에서 프로그램이 가능하게 되어 여러 가지 상황에 유동적으로 대처할 수 있게 되었다. 데이터 전송속도 역시 Gen1에 비해 4.5배에서 8배 빠른, 최대 640kb/s까지 사용할 수 있게 되어 태그 인식속도를 빠르게 향상 시켰다. Kill 및 액세스 패스워드의 길이도 32bit로 확장됨에 따라 고객의 프라이버시 문제나 태그의 보안성을 크게 향상시켰다.

본 논문에서는 이 Gen2 프로토콜에 따른 리더의 명령어 처리와 태그의 응답 및 상태 변화를 효율적으로 처리할 수 있는 디지털 프로세서의 설계 및 구현 통하여 다수의 태그들이 보다 효율적이고 빠르게 인식되는 것을 검증 할 것이다.[1]

II. Class 1 Generation 2 UHF Air Interface Protocol(Gen2)

RFID Gen2 태그는 그림 1에서 보는 바와 같이 수동형으로써 역방향 링크에서 후방산란(Backscatter)이라는 일종의 커플링 방식으로 리더에게 응답한다. 후방산란 방식에서 리더는 태그에게 정방향 링크에서 조금 더 많은 에너지를 보내기 위해 Pulse Interval Encoding(PIE)을 사용하여 태그의 전력을 효율적으로 공급하게 된다. 태그는 공급받은 이 에너지의 일부를 리더에게 되돌려 보냄으로써 응답하게 된다. 태그에서 반사하는 신호는 리더의 반송파 주파수 신호에 태그의 ID 정보(EPC)를 변조해서 보내게 된다. 이러한 일련의 과정을 후방산란이라고 한다.

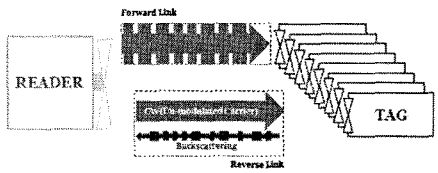

그림 1. UHF 대역 수동형 RFID 시스템

후방산란을 할 때에 태그는 FM0와 밀러 서브캐리어라는 두 가지 변조 방식을 택하게 된다. 전자는 ISO 표준에서 효과적으로 사용되고 있는 포맷으로 빠르지만 간섭받기 쉬운 특성을 가지고 있고 후자는 느리지만 노이즈 환경에서 효율이 더 좋다. 하여 밀러 서브캐리어 방식은 FM0방식이 여러 리더들이 존재하여 간섭이 심한 환경에서 태그 인식을 실패했을 때 인식 속도는 느리지만 태그 인식 을 가능하게 해준다. 본 지에서는 리더가 태그를 최고속도인 640Kbps 즉, FM0방식을 통하여 인식하게 설정하여 설계 및 구현 하였다.

RFID 리더와 태그를 인식하는 동작에 있어서 리더는 Select, Inventory, 그리고 Access 이렇게 세 단계로 구분하며, 태그는 Ready, Arbitrate, Reply, Acknowledged, Open, Secured, 그리고 Killed 이렇게 7개의 상태로 구분한다.

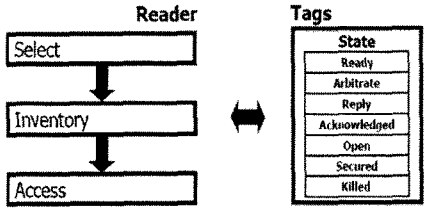

그림 2. 리더 동작 단계와 태그들의 상태

리더의 세 가지 단계에 따른 명령어들의 의해서 태그는 위의 7개의 상태를 천이하며 명령어들을 수행해 나간다.

리더가 태그를 인식하기 위한 준비 단계인 Select 단계에서 리더는 인식 가능한 태그들의 범위를 설정하기위해 태그들에게 Select명령을 전송하고 전력을 공급하게 된다. 이에 전력을 공급받은 태그들은 Ready 상태가 되며, 리더의 명령어들을 받을 준비를 한다. 태그의 EPC 값을 얻기 위한 리더는 두 번째 Inventory 단계로 넘어 가게 된다. 이 단계에서는 태그들의 충돌을 최소한으로 줄이면서 효율적으로 인식하기 위한 Gen2의 알고리즘인 FS-Aloha를 사용하기 위한 명령어들을 전송하게 된다. 이 때 사용되는 명령어는 Query, QueryAdjust, QueryRep, ACK, 그리고 NAK가 있다. 태그는 4개의 세션필드를 가짐으로서 여러 리더가 선점할 수 있도록 다중 리더 환경에서의 효율을 높였다.

본격적으로 리더는 태그들의 EPC코드를 읽기 위해 FS-Aloha 방식을 이용하여 태그를 읽기 시작한다. 처음으로 Query 명령어를 태그들에게 전송한다. 이 Query 명령어를 받은 태그들은 Ready 상태에서 Arbitrate상태로 천이 하게 된다. 이 명령어에는 4 bits의 'Q' 값이 포함되며, 이 'Q'값은 각 각의 태그가 가지고 있는 RNG (Random Number Generator)에서 발생되는 랜덤 넘버의 값의 범위를 제한하게 되고, 이때에 발생되는 랜덤 넘버는 그 태그의 슬롯 카운터 값이 된다. 슬롯 카운터 값의 범위는 '0'에서 '2^Q-1'까지 이다. 태그는 자신의 슬롯카운터 값이 '0'이 되었을 때, 리더에게 RNG에서 생성한 16 bits의 RN16 값을 전송하게 되며 Reply 상태로 천이 한다. 이에 리더는 ACK명령어를 태그에게 보내게 되고 이 ACK명령어를 받은 태그는 리더가 원하는 EPC코드를 전송하여 Inventory 단계를 끝맺게 된다. 하지만 동시에 슬롯 카운터 값이 '0'인 태그가 두 개 이상이 있다면, 양 태그가 동시에 리더에게 응답을 하게 되어 응답 충돌이 발생하게 되고 리더는 두 태그에게 NAK 명령을 보내게 된다. 여기서 NAK를 받은 두 태그들은 최하위 응답 순위로 밀려나게 된다.

리더는 슬롯 카운터 값이 '0'인 태그가 존재 하지 않을 때에는 범위안의 모든 태그의 슬롯 카운터 값을 줄이기 위해 QueryRep 명령어를 태그들에게 전송하게 된고, 이 명령어를 받은 태그들은 자신의 슬롯 카운터 값을 '1'씩 줄이게 된다.

슬롯 카운터의 값을 Q값에 의해 범위가 결정되므로 리더는 인식 범위 안에 있는 모든 태그의 수에 따른 적절한 Q값의 설정을 위해 QueryAdjust명령어를 사용한다. 이 명령어는 Q값을 수정하므로 태그들이 현재 가지고 있는 슬롯 카운터 값을 효과적으로 조정하는 결과를 가져온다.

리더의 Access 단계는 Inventory 단계를 마친 태그와 일대일로 통신을 하는 단계이다. 이 단계에서 리더는 태그의 메모리 추가 정보를 읽거나 제공되는 메모리 공간에 쓰기 명령을 사용할 수 있으며, 해당 태그의 암호를 입력하여 영구적으로 Kill 할 수 있다. 본 저에서는 태그 인식률에 논점이 있으므로 EPC코드를 얻기 위한 과정인 Inventory 단계까지 태그를 구현하였다[2].

III. 프로토콜 처리기 설계

리더와의 통신에 있어 핵심이 되는 수동형 태그의 디지털 회로부에 대한 블록 도는 명령어 처리 및 전달 지연을 최소화 하는 것을 목표로 그림 3과 같이 구성 하였다.

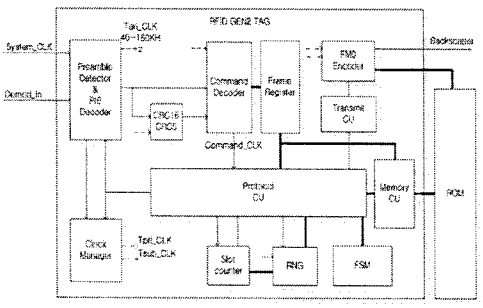

그림 3. 태그 프로토콜 처리기의 블록도

리더의 명령들은 그 종류에 따른 길이와 처리 방식이 모두 다르다. 따라서 직렬 입력되는 명령의 버퍼링과 동시에 명령어 프레임이 끝나기 전에 명령어를 구분하는 것은 물론, 메모리의 특정 주소의 값과 비교하거나 CRC와 같은 데이터 오류검사를 병행해야 하므로 파이프라인 방식의 설계를 통해 전달 지연을 최소화 할 수 있다[4].

· 3.1 프리앰블 & 프레임 싱크 제거기(PFR)

리더는 모든 명령 프레임 앞에 12.5μs의 Delimiter와 데이터 '0'의 길이인 Tari와 Tari의 2.5배에서 3배 사이의 길이를 가지는 RTcal, 이 세 가지 심벌이 합쳐진 Frame-sync를 붙여 태그에게 전송한다. 예외적으로 역방향 링크의 태그 응답 속도를 지정하기 위해 Query 명령어 앞에는 Frame-Sync 심벌에 RTcal의 1.1배에서 3배 사이의 길이를 가지는 TRcal 심벌이 더해진 Preamble을 덧붙여 전송한다.

Preamble & Frame-Sync Remover(PFR) 블록은 Delimiter를 감지하는 상태, Tari를 알아내는 상태, 그리고 PIE 신호를 디코딩 하는 상태로 구분하여 동작한다. PFR 블록은 명령 프레임이 시작되면 Decoder를 활성화 시키며, 그 끝은 Decoder로부터 프레임 종료 신호를 받아 Delimiter를 감지하는 초기 상태로 돌아가게 된다. PFR 블록은 복조된 PIE신호를 1.92 MHz의 시스템 클록으로 샘플링 하여 Low와 High를 구분하여 각각의 심벌을 읽는다.

· 3.2 프레임 디코더

리더의 명령어는 그 종류에 따라 프레임의 길이 각기 다르며, 명령어 구분을 위한 헤더 필드도 각기 길이가 다르다. 명령어가 들어오면 128비트 레지스터에 버퍼링을 함으로 써 각각의 길이에 맞는 샘플링을 통해 해석한다. 그리고 각 명령어 프레임의 끝에는 오류 체크를 위한 5비트 또는 16비트의 CRC가 존재하므로 CRC검출기를 통해 오류를 확인한 다음 참일 경우에만 해당 명령을 컨트롤 유닛으로 전송하도록 구현하였다.

· 3.3 프로토콜 컨트롤 유닛(PCU)

프로토콜 컨트롤 유닛(PCU)는 Finite State Machine(FSM), 슬롯 카운터, 전송 컨트롤러 및 메모리 컨트롤러를 관리하는 핵심 유닛으로 태그의 응답을 관장한다. 앞 장에서 설명한 태그 응답 충돌을 줄이기 위한 FS-Aloha 알고리즘을 PCU에서 처리한다.

Select 명령은 현재 인식단계에서 사용될 세션 및 그림 4와 같은 세션에 따른 인식표(Inventory Flag) 값과 선택 표(Seleted Flag)의 값을 취하기 위해 메모리를 참조하여 그 값들을 설정한다. 표준에서는 이 값들의 물리적 유지 시간을 정하여 다중 리더 상황에서 세션을 구분하여 인식하도록 한다[3].

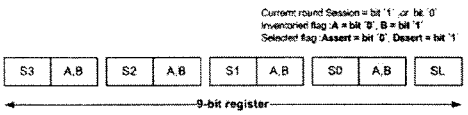

그림 4. 세션과 Flag 값 구분을 위한 레지스터

Query명령어의 각 필드의 값들을 처리하며

FSM에 자신의 상태가 Arbitrate임을 알려준다. RNG로부터 Q값에 따른 슬롯 카운터 값을 가져온다. 슬롯 카운터 값이 '0'이 되면, FSM에 Reply 상태임을 알려주고 RNG로부터 RN16 값을 받아 레지스터에 저장해 둔다. 이 후 리더로부터 ACK를 받아 레지스터에 저장된 RN16과 ACK명령에 실린 RN16 값을 비교하여 같을 경우 FSM을 Acknowledged 상태임을 알리고, FSM은 EPC 값 전송을 위해 전송 컨트롤러와 메모리 컨트롤러를 활성화 시킨다.

· 3.4 슬롯 카운터 및 RNG

태그의 응답 순서를 결정하는 슬롯카운터는 RNG의 값이 얼마나 랜덤한지에 따라 태그 응답 충돌을 피할 수 있다. 하여 태그가 작동 되는 순간부터 종료 될 때 까지 RNG는 시스템 클록에 따라 랜덤 수를 발생하도록 하였다. 16비트 LFSR(Liniear Feedback Shift Register)로 구현한 RNG는 레지스터들의 값이 모두 '0'이 되면 더 이상 값을 생성할 수 없다. 또한 RNG 레지스터의 초기 값이 같은 태그가 있을 경우도 태그 응답 충돌의 발생 확률이 높아진다. 따라서 RNG의 시프트레지스터가 모두 '0'이 될 때와 초기 값은 각각의 태그의 유일한 값인 EPC의 16비트 CRC 값으로 초기 값을 설정하였다.

Ⅳ. 시뮬레이션

(GEN2 표준을 지원하는 태그의 디지털 구현 검증을 위해 Xilinx VERTAX 4가 장착된 FPGA 프로토 타이핑 보드를 사용하였다. 검증하기 위한 환경은 다음 그림 5에서 보는 것과 같다.

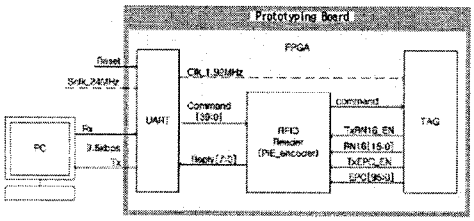

그림 5. RFID Gen2 Protocol의 검증 환경

검증의 목표는 리더의 Select단계에서 태그의 EPC코드를 얻을 수 있는 Inventory 단계까지 한 주기를 실행하여 태그의 인식 시간을 확인 하는 것이다. 하여 그림 5에서 보는 것과 같이 PC에 C로 설계한 리더에서 Select 명령어로 Flag값을 설정한 후 Q값이 '0'인 Query 명령을 전송하여 태그의 슬롯 카운터 값을 '0'으로 만든다.

이후 태그로부터 RN16 값을 받은 후 ACK 명령을 전송하여 태그 메모리의 EPC 값을 확인하는 것이다.

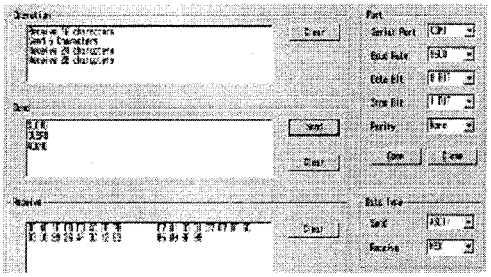

그림 6. 리더와 태그의 통신

위의 그림 6에서도 보시는 바와 같이 검증 환경을 만들기 위해서 직렬 통신 규격인 UART(Universal Asynchronous Receiver Transmitter)설계를 하였으며, 태그에 명령을 전달하기 위해 리더의 PIE 인코딩 블록의 설계를 추적으로 해주었다.

Ⅴ. 결 론

UHF 대역의 태그들은 수 미터의 인식거리로 인하여 다중 태그 상황을 고려한 충돌 방지 알고리즘을 지원해야 한다. 본 검증에서는 RFID 표준인 Gen2 프로토콜의 리더와 태그를 스펙의 요구사항을 만족하는 시스템을 PC와 프로토 타이핑 보드위에 구현 및 검증을 통해 태그의 Inventory Round의 동작 과정에 대하여 고찰 하였고, 그 시간을 측정해 봄으로 써 태그 인식률을 높이기 위한 개선방안을 찾는데 도움을 주고자 하였다.

참고문헌

[1] 안재명, 이종태, 오해석, "EPCglobalNetwork 기반의 RFID 기술 및 활용", Global, 2007

[2] EPC Global, "*EPC Radio-Frequency identity Protocols class-1 generation-2 UHF RFID protocols for communications at 860MHz~960MHz version 1.1.0*," EPC global, 2007

[3] Daniel M. Dobkin, "Gen2 Inventory Operation And Multiple Session.", *www.rfidsolutionsonline.com*, 2006