

RFID GEN2 태그 표준의 VHDL 설계

준회원 장 일 수*, 종신회원 양 훈 기**

VHDL Implementation of GEN2 Protocol for UHF RFID Tag

Il Su Jang* Associate Member, Hoon Gee Yang** Lifelong Member

요 약

본 논문에서는 UHF 대역 RFID 수동형 태그의 디지털 회로부 구현을 위한 VHDL 설계과정을 보인다. 태그의 동작은 EPCglobal Class1 Gen2 표준을 지원하며 합성과 구현과정을 거친 타이밍 시뮬레이션 결과를 통해 검증하였다. 수 미터의 인식거리로 인해 Frame-Slotted Aloha를 사용하는 환경에서, 단위시간당 태그 인식률을 향상시키기 위해서는 리더 명령에 대한 빠른 처리와 응답을 할 수 있는 디지털 회로 설계가 필요하다. 본 설계는 Pipeline 처리 구조를 기반으로 직렬 입력 신호에 대한 응답 지연의 최소화를 목표로 하였다. 또한, 효율적인 다중 접속 명령들의 처리와 태그의 데이터 전송 속도의 오차를 낮추기 위해 리더의 Preamble과 PIE 디코딩을 위한 샘플링 과정을 제안하였다. FPGA 검증을 위한 Place & Route 후 다중 태그 상황을 감안한 테스트 벤치 시뮬레이션 결과, 표준상의 최대 송수신 데이터 전송 속도에서 디코딩 및 인코딩을 위한 최소 요구 시간 보다 빠른 처리 결과를 확인할 수 있었다.

Key Words : RFID, GEN2, FPGA, VHDL, Passive tag

ABSTRACT

This paper presents the VHDL implementation procedure of the passive RFID tag operating in Ultra High Frequency. The operation of the tag compatible with the EPCglobal Class1 Generation2(GEN2) protocol is verified by timing simulation after synthesis and implementation. 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 an interrogation rate. In order to satisfy linking time, the pipe-line structure is used, which can minimize latency to serial input data stream. We also propose the sampling strategy to decode the Preamble, the Frame-sync and PIE symbols in reader commands. The simulation results with the fastest data rate and multi tags environment scenario show that the VHDL implemented tag performs faster operation than GEN2 proposed.

I. 서 론

RFID 시스템은 무선 환경으로 리더와 태그로 구성된다. 현재 널리 사용되고 있는 13.56MHz 대역의 접촉식 RFID 환경과 달리 수 미터의 인식 거리를 갖는 UHF대역(860-960MHz)의 RFID 환경에서

는 다수의 리더와 다수의 태그들로 이루어진 무선 네트워크 환경이 된다. 특히 물류 분야를 비롯한 물품의 재고 관리와 같은 응용분야에서, 단위 시간동안 인식되는 태그의 수가 시스템의 성능을 좌우한다. 따라서 UHF 수동형 RFID 시스템에서는 다중 태그 인식률에 크게 영향을 주는 데이터 충돌 방지

※ 본 연구는 2007년도 「서울시 산학연 협력사업」의 「나노IP/SoC설계기술혁신사업단」 과 정보통신부 및 정보통신 연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2007-C1090-0701-0038)

* 광운대학교 전파공학과 석사과정 (ilssujjang@nate.com), ** 광운대학교 전파공학과 교수 (hgyang@kw.ac.kr)

논문번호 : KICS2007-06-285, 접수일자 : 2007년 6월 28일, 최종논문접수일자 : 2007년 12월 4일

프로토콜과 이를 처리하기 위한 프로세서가 태그마다 필요하게 된다.

수동형 태그의 구조는 크게 안테나, 아날로그 모듈, 디지털 프로세서와 메모리로 구성된다. 특히, 디지털 프로세서는 RFID의 응용분야가 넓어짐과 동시에 더욱 복잡한 설계가 요구되며 충돌 방지와 메모리 컨트롤 등의 역할로 인해 그 중요성이 크다. 일반적으로 디지털 설계는 크게 설계된 칩의 면적, 단위 시간당 처리량과 같은 성능과 저 전력 설계를 목표로 설계 목적이 구분된다. 기존의 수동형 태그 설계는 인식거리 증가를 위해 저전력 설계에 중점을 두고 있으며, 태그 가격과 밀접한 칩의 면적 및 설계에 사용된 게이트 수 감소를 위한 새로운 설계 방식을 제안하고 있다^[2-9]. 또한 태그의 보안을 위해 암호화 기법을 추가한 설계 역시 이슈가 되고 있다^[10,11].

그러나 일 대 일 통신 방식이 아닌 이상 인식거리의 증가는 리더가 인식할 태그 수의 증가와 비례하게 되며, 리더는 태그들을 인식하기 위해 단일 태그 상황과 달리 훨씬 많은 횟수의 명령어 전송을 해야 한다^[13,14,15]. 결국 태그 역시 처리해야 명령어 많아지게 되므로 빠른 데이터 변복조 성능과 응답 상태 변화는 RFID 시스템의 인식 성능을 개선시킬 수 있을 것이다.

따라서 본 논문의 수동형 태그는 UHF RFID 무선 규격인 EPCglobal Class1 Generation2 표준을 지원하며, RFID 시스템 인식률 개선을 위해 가변 송수신 데이터 속도의 처리 방법과 응답 요구시간을 단축시키기 위한 명령 프레임 처리과정을 제안한다. EPC Class1 Generation2와 같은 Slotted-ALOHA 기반의 시분할 랜덤접속 환경에서 리더 명령의 처리와 태그의 응답 및 상태 변화가 효율적인 디지털 프로세서의 설계는 다수의 태그들이 보다 빠르게 인식 될 수 있게 할 것이다. 설계 언어는 VHDL이며, 통신 시스템 신호처리와 ASIC prototype 등 전반적으로 설계 량이 크며 다양한 기능을 포함하고 있는 FPGA 검증을 위한 다중태그 상황을 고려한 타이밍 시뮬레이션 결과를 보일 것이다.

II. EPC Class1 Generation2 표준

2.1 GEN2 표준의 특징

국제 표준인 EPC global Class1 Generation2 (ISO 18000-6C) 프로토콜은 현재 가장 많이 사용되고 있는 UHF 대역의 RFID 표준이다. Class0와 Generation1과 같은 이전 버전의 표준에 비해 더욱

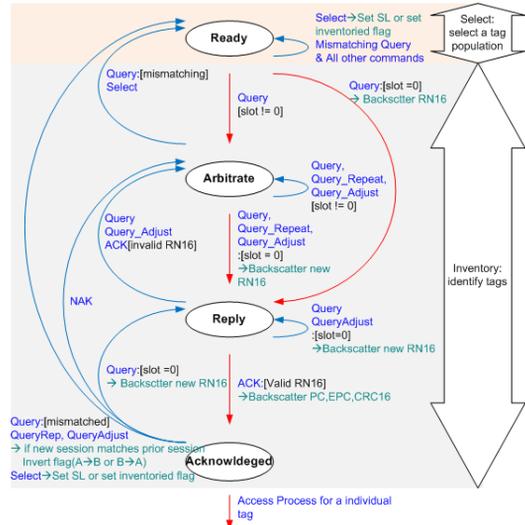


그림 1. EPC GEN2 태그의 응답 상태도

빠른 데이터 전송속도를 지원하며 높은 신뢰도로 태그들을 인식할 수 있도록 한다. 또한, 태그들을 빠르게 인식하기 위한 방법으로 환경에 따라 리더와 태그의 데이터 전송률을 변화시킬 수 있으며 초당 1000개 이상의 태그 인식률을 목표로 하고 있다.

태그들은 Dynamic-Frame-Slotted Aloha라고 하는 시분할 다중 접속 방식을 사용한다. 기본 원리는 Slotted Aloha와 같은 랜덤 접속 방식이지만 슬롯수를 변화 할 수 있는 프레임 내에서 태그들이 응답할 수 있는 슬롯이 정해져 있으며 리더 명령어중 Query의 Q값에 의해 응답 순서를 결정하기 때문에 Q알고리즘이라고도 한다. 이전 버전인 Gen1에서는 리더에 인식되었거나 인식이 불필요한 태그들을 구분하기 위해 Sleep 모드를 사용하는데, 이때 태그를 Sleep모드로 설정하거나 해제하기 위해서 리더의 명령이 더 많이 필요하게 된다. 하지만 GEN2에서는 Symmetry라는 방식으로 태그들을 크게 A와 B로 분류하여 인식할 태그와 인식된 태그를 구분한다. 이때 특정한 리더의 명령을 사용하지 않아도 되므로 단위 시간당 태그 인식률을 향상시킬 수 있다^[11,18,19].

2.2 GEN2 리더와 태그의 동작 및 상태

리더는 태그들과 통신하는 과정을 Select, Inventory, Access 세 단계로 구분 짓는다. 태그는 다중 접속과 메모리 접근을 위해 Ready, Arbitrate, Reply, Acknowledged, Open, Secured, Killed 상태를 갖으며, 상태에 따른 응답과 응답 대기 상태가 정의되어 있다. 리더의 안테나 방사 패턴안의 태그들은 일

정 레벨의 전력을 공급받아 Ready 상태가 되고, 리더는 태그를 인식하기 위한 Select 과정과 Inventory 동작을 진행한다.

리더의 Select 동작에서는 태그의 메모리 값을 이용하여 태그들을 선별하거나 인식거리내의 태그들을 인식하기 위해 Select 명령을 사용하며, 태그들은 Ready 상태에 머문다.

리더는 Select 동작 후 Inventory 동작에서 실제 태그들과 일대일 통신을 하기 위한 충돌방지알고리즘을 진행한다. 사용 되는 명령어는 Query, Query-Adjust, Query-Repeat, ACK, NAK 명령이며, ACK와 NAK를 제외한 명령들은 세션 필드를 포함하여 다중 리더 상황에서 세션별로 명령들을 구별한다. 태그의 응답 속도를 결정하며, 태그 인식 과정에서 가장 중요한 명령어인 Query의 역할은 4비트의 Q 값을 전송하는 것이다. Query는 태그들이 갖고 있는 RNG(Random Number Generator)를 재생성 함과 동시에 $2^Q - 1$ 이내의 범위에서 발생된 랜덤 값을 태그마다 하나씩 갖고 있는 응답 순서인 슬롯카운터에 저장하게 한다. Query-Repeat는 4비트의 가장 짧은 명령어로 명령어 구분을 위한 2비트와 세션 정보 2비트만을 전송한다. 해당 세션의 태그들은 이 명령을 받으면 태그들의 슬롯카운터에서 무조건 1감소시킨다. Query-Adjust는 태그들이 갖고 있는 Q 값에서 1증가 혹은 1감소하거나 현재 Q값을 유지 시킨 채로 RNG로부터 변화한 Q값에 해당하는 랜덤 값을 RNG로부터 슬롯카운터에 새롭게 저장한다. 리더는 인식거리 이내에 태그 수를 알 수 없기 때문에 Q 범위가 인식 될 태그들의 수와

많은 차이를 보이는 상황에서 Query-Adjust 명령을 사용하여 태그 응답의 충돌을 피하거나 Query-Repeat의 전송 횟수를 감소시켜 빠른 인식을 가능하게 한다.

태그는 Query, Query-Repeat, Query-Adjust 명령을 받으면 Ready 상태에서 Arbitrate 상태가 되어 슬롯 카운터가 0이 될 때까지 리더에 응답을 대기한다. 태그는 각 명령의 처리 후 슬롯카운터 값이 0이 되면 Reply 상태가 되어 리더에게 일대일 통신을 요청하기 위해 RN16이라는 16비트 랜덤수를 전송한다. 리더는 RN16을 받으면 RN16을 포함한 ACK를 전송하며, 응답한 RN16 값과 ACK의 RN16이 같은 태그는 Acknowledge 상태가 되어 태그 정보인 메모리의 EPC코드를 전송한다.

리더의 Access 동작 단계는 Inventory 단계를 마친 태그와 일대일 통신을 하는 단계이다. 이 동작 단계에서는 태그 메모리의 추가 정보를 읽거나 제공되는 메모리 공간에 쓰기 명령을 사용할 수 있으며, 해당 태그의 암호를 입력하여 영구적으로 Kill 할 수 있다⁶⁾.

III. 프로토콜 처리기의 설계

리더와의 통신에 있어 핵심이 되는 수동형 태그의 디지털 회로는 그림 2와 같이 구성 할 수 있다. 리더의 명령 프레임들은 각각 길이와 처리 방식이 모두 다르기 때문에 직렬 입력되는 명령 프레임의 버퍼링과 동시에 명령어 프레임이 끝나기 전에 명령어를 구분하는 것은 물론, 메모리의 특정 주소의

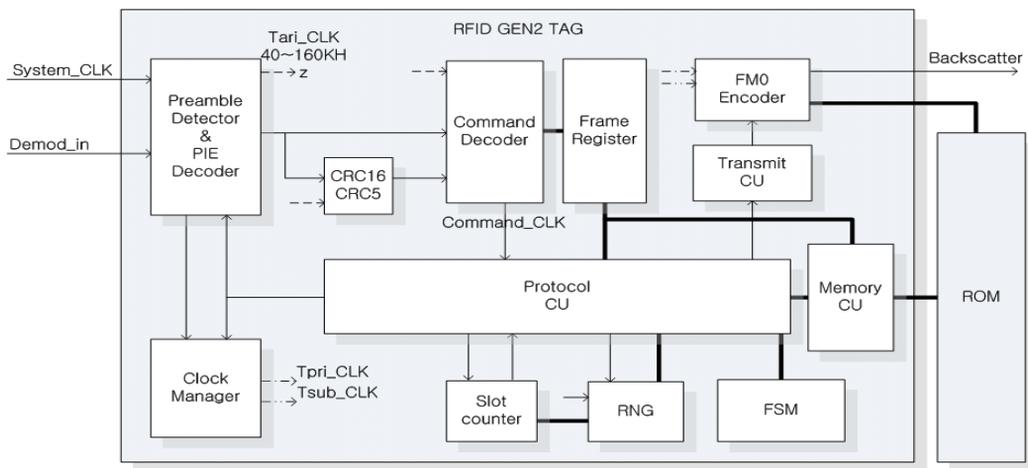


그림 2. 태그의 프로토콜 처리기 블록도

값과 프레임 필드 값을 비교하거나 CRC와 같은 데이터 오류검사를 병행해야 하므로 병렬 처리와, Pipe line 설계를 통해 응답 지연을 최소화 할 수 있다^{6,7)}.

각 블록들이 동작하기 위한 클록은 이전 블록에서 생성된 클록을 사용하거나 Clock Manager 블록의 출력 클록과 enable 신호를 gating하여 사용한다. 이는 디지털 회로의 소비 전력은 로직 게이트를 이루고 있는 트랜지스터들의 스위칭 속도가 빠를수록 커지게 되므로, 결국 각 블록의 클록 사용량을 줄이기 위해 gated-clock의 사용과 표준 요구사항을 만족하는 최소 주파수의 시스템클록을 사용하여 소비 전력을 최소화 할 수 있으며, 본 설계에서 태그의 시스템 클록 주파수는 1.92MHz이다^{6,8,19)}.

3.1 Preamble detector & PIE decoder

리더로부터 Continuous Wave(CW)를 통해 동작 전력을 공급 받는 태그는 지속적인 high 레벨 신호를 감지한다. 따라서 리더는 태그에게 명령 프레임 시작을 알리기 위해 12.5us 동안의 low 레벨(delimiter)을 시작으로 비트 '0'의 심볼 Tari와 비트 '0'과 '1'을 구분하기 위한 기준인 RTcal(Reader to Tag Calibration) 심볼로 이뤄진 Frame-sync를 전송한다. 특별히 Query 명령에서는 태그 응답 속도를 설정하기 위해 Frame-sync에 TRcal(Tag to Reader Calibration) symbol을 추가한 Preamble을 사용한다.

GEN2 리더는 CW 동안은 물론 명령 전송 중에도 태그의 동작 전압을 유지시키기 위해 비트 '0'과 '1'을 시간 길이로 구분하여 인코딩하는 Pulse Interval Encoding(PIE)을 사용한다.

Preamble detector & PIE decoder블록은 delimiter를 감지하는 초기 상태, Tari 값을 알아내는 상태, RTcal을 감지하는 상태와 PIE 신호를 디코딩 하는 상태로 동작한다. 이 블록은 복조된 신호를 1.92MHz의 시스템 클록(SCLK)으로 샘플링 하며 동작원리는 다음과 같다.

복조신호가 12.5us인 시스템 클록 기준으로 24 클록 동안 Low 레벨이 유지되면 delimiter를 감지한다. Delimiter이후 Tari를 측정하기 위한 카운터와 복조신호의 Low 레벨 구간인 Pulse Width(PW) 값을 측정하는 카운터를 동작하여 Tari 길이를 알아낼 수 있다. 표준에서 리더 명령의 비트 '0' 길이인 Tari 값은 6.25us, 12.5us, 25us이다. 이때 Tari를 구하기 위해 샘플링 한 신호레벨이 high는 bit '1'

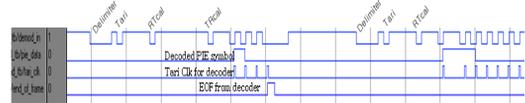


그림 3. Preamble과 Frame-sync의 복조 및 PIE 디코딩 출력파형(Tari=6.25us)

표 1. Tari와 시스템 클록 트리거 수

Tari (us)	Tari*Systemclock	Tari (clocks)	PW (clocks)
6.25	6.25us × 1.92MHz	12	6
12.5	12.5us × 1.92MHz	24	12
25	25us × 1.92MHz	48	24

Low는 bit '0'이므로, 이 값들을 더하여 12클록부터 우선순위를 두어 샘플 값의 합이 6이하이면 Tari는 6.25us, 24클록 후 12이하이면 12.5us, 48클록 후 24이하이면 25us로 결정하며 그에 해당하는 PW 값을 알 수 있다.

Tari 길이를 알아내면 RTcal 길이를 측정위한 상태가 된다. RTcal 값은 (1.5~3)*Tari 값이기 때문에 Tari를 구하면 자동적으로 RTcal 값의 최대 최소 길이를 알 수 있다. 이 역시 복조신호가 high로 유지 되다 low로 변하는 순간부터 PW 카운터를 동작하도록 한다. 3*Tari-PW이후까지 신호레벨이 low가 되지 않으면 delimiter를 구하기 위한 블록의 초기 상태로 돌아간다. 그전에 low 신호가 PW 구간 동안 유지되면 RTcal이 측정된다.

RTcal 이후 Preamble 이라면 TRcal이 오지만, Frame-sync 라면 PIE 신호가 오기 때문에 바로

PIE 디코딩에 들어가야 한다. 이 역시 PW구간이 앞서 구한 Tari/2 이내에 오면 데이터는 '0'이며, 최대 1.5*Tari 이내에 PW가오면 데이터 '1'로 디코딩이 가능하기 때문에 표준에서 (1.5~2)*Tari 값을 갖는 비트 '1'의 길이에 대해 모두 디코딩이 가능하다. TRcal의 길이는 (1.1~3)*RTcal의 값을 갖기 때문에 최소 1.65*Tari 이전에 PW가 오지 않으므로 PIE 디코딩과 TRcal 복조가 동시에 가능하다.

리더의 명령은 비트 '0'을 기준으로 40, 80, 160Kbps의 데이터 전송속도를 갖는다. 또한 PIE 방식은 데이터 '0'과 '1'의 길이 가 다르기 때문에 일정한 트리거를 발생하는 클록으로 프레임 디코딩이 어렵다. 따라서 PIE디코더는 Command decoder가 데이터를 버퍼링하는데 필요한 Tari_CLK을 생성하여 전달한다. 이는 CRC블록으로도 전달되어 리더 명령 프레임의 데이터 오류를 검사한다. Preamble

detector & PIE decoder 블록은 명령 프레임이 시작되면 Command decoder를 활성화 시키며, 명령의 끝은 Command decoder로부터 프레임 종료 신호를 받아 delimiter를 감지하는 상태로 초기화 된다.

3.2 클럭 분주기

GEN2 표준에서는 태그의 응답 속도가 40~640Kbps로 리더에 의해 설정 된다. 리더가 설정한 태그의 응답속도와 태그의 실제 응답속도가 다르면 리더는 태그의 응답을 해독할 수 없다. 리더의 인식률에 영향을 주게 되는 올바른 태그의 응답은TRcal이 얼마나 정확하게 측정에 되는지에 달려있다 [7]. 클럭 분주기는 태그의 응답인 FM0와 Miller 인코딩에 사용할 Tpri_clk과 Tsub_clk을 시스템 클럭(SCLK)으로부터 분주하는 역할을 하기 때문에 클럭 주파수에 따라 태그의 응답 오차율이 달라지므로 칩 설계 시 시스템 클럭 주파수 선택은 칩 설계 시 고려해야 할 가장 큰 변수 이다.

태그의 응답 속도(Link Frequency)는 시간 길이인 Preamble의 TRcal 값과 Query 명령의 DR(Divide ratio) 값에 의해 식 (1)과 같이 계산된다.

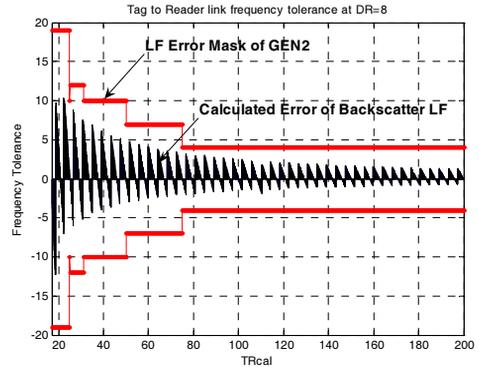
$$LF(Hz)=DR/TRcal(sec), DR=8 \text{ or } 64/3 \quad (1)$$

태그의 응답속도는 Query 명령에 의해 정해지기 때문에 Protocol Control Unit(PCU)로부터 Query 명령의 처리유무와 DR 값을 전달 받아 응답 속도를 계산한다. 클럭 분주기의 출력인 Tpri_CLK, Tsub_CLK는 태그가 Reply 상태에서만 출력되어 인코더로 전달하도록 PCU로부터 enable 신호를 받아 활성화 된다.

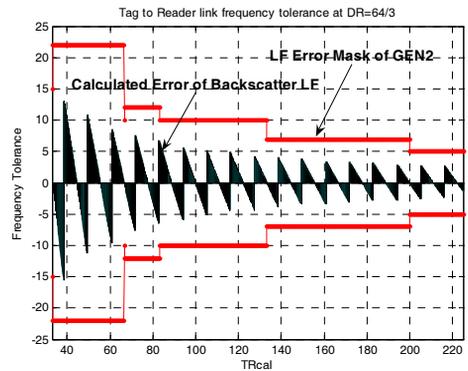
TRcal을 측정하면서 카운트된 시스템 클럭수를 TRcal_c라 하면, 실제 리더가 전송한 TRcal 값과 최대 시스템 클럭의 한주기 이내의 오차 (+0.52us

표 2. TRcal에 따른 태그 응답 주파수 오차 예

DR	TRcal (us)	TRcal_c	LF (KHz)	LFc =1.92*DR /TRcalc	LF 오차범위 (GEN2)	분주율
64/3	33.3	64	640	640	+/-15%	1/3
	50	96	426	384	+/-22%	1/5
	66.7	129	319	320	+/-10%	1/6
	83.3	160	256	240	+/-10%	1/8
	175	336	121	120	+/-7%	1/16
8	17.2	34	465	480	+/-19%	1/4
	25	48	320	320	+/-10%	1/6
	31.25	60	256	240	+/-10%	1/8
	60	116	133	128	+/-7%	1/14
	180	346	44	44.6	+/-4%	1/43



(a)DR=8



(b)DR=64/3

그림 4. 태그 응답 속도 오차율

@ 1.92MHz)를 갖게 된다.

오차를 줄이기 위해 시스템 클럭을 높은 주파수로 사용할 경우 전력소비가 커져 태그의 동작 전압이 높아지게 되고, 결국 UHF대역의 장점인 인식거리가 줄어들게 된다. 표준의 태그 응답 속도를 모두 지원하기 위해 LF를 구하는 과정에서 나눗셈 연산이 필요한데 분주 비율이 정수가 되지 않는 경우가 대부분이다. 본 설계에서는 Preamble detector & PIE decoder 블록에서 측정된TRcal_c 값에 따라 시스템 클럭으로부터 LF 분주비율을 식 (2)와 같이 계산하여 표준의 응답 주파수 오차 범위를 그림 4와 같이 만족하였다.

$$Backscatter_LF = SCLK/Round(TRcal_c/DR)$$

$$분주율 = SCLK/Backscatter_LF \quad (2)$$

3.3 프레임 디코더

리더의 명령어는 데이터 길이가 모두 다르다. 따라서 명령어의 종류를 알려주는 각 명령어의 첫 번

제 필드 값을 먼저 해독하여 각 명령어의 길이를 판단한다. 그러나 명령어의 종류를 알려주는 첫 필드 역시 2bit, 4bit, 8bit로 길이가 다르기 때문에, 명령어를 버퍼링함과 동시에 Preamble detector & PIE decoder로부터 전해지는 Tari_clk을 카운트하여 2, 4, 8 Tari_clk 마다 명령어 종류를 우선적으로 구분해야 프레임 끝을 결정 할 수 있다.

명령어의 종류가 판단되면 연속적인 필드들의 값은 128비트 레지스터에 버퍼링한다. 고정길이의 명령을 제외한 가변길이의 Select 및 Read, Write 명령들은 CRC16 검사를 통해 명령의 끝을 알 수 있다. 따라서 리더 명령 프레임은 디코더와 CRC검출기에 동시에 입력되며 오류검사가 참일 경우에만 해당 명령을 컨트롤 유닛으로 전송하도록 설계하였다. 이러한 설계 방식은 부가적으로 태그의 소비 전력을 낮추는 효과를 가져 올 수 있다^{5,7)}.

프레임 디코더는 해독된 명령어의 필드 값과 명령의 마지막 비트 처리 후 Command_clk을 생성하여 컨트롤 유닛으로 전송한다.

3.4 Protocol Control Unit(PCU)의 명령어 처리
Protocol Control Unit(PCU)은 Finite State Machine(FSM), 슬롯카운터, 전송 컨트롤러 및 메모리 컨트롤러를 관리하며 태그의 응답 여부를 판단하며, 프레임 디코더로부터 전달 받는 Command_clk의 상향 트리거에 동작한다.

Select 명령은 현재 인식단계에서 사용될 세션 및 그림 5와 같은 세션에 따른 인식표(Inventoried flag) 값과 선택표(Selected flag)의 값을 취하기 위해 메모리를 참조하여 그 값들을 설정한다. 표준에서는 이 값들의 물리적 유지시간을 정하여 다중 리더 상황에서 세션을 구분하여 인식하도록 한다^{11,18)}.

Select 명령의 수신이 끝나기 전에 컨트롤러는 메모리의 해당 주소 값과 Select 명령의 마스크 값의 비교 결과를 알고 있어야 한다. 이를 처리하기 위해 디코더는 Select 명령으로 판단하면 버퍼링된 메모리주소, 마스크길이, 마스크 값을 메모리 컨트롤러에 전달함과 동시에 활성화 시킨다. 메모리주소는 Membank+Pointer, 마스크 길이는 Length로 Select 명령의 필드 값이다. PCU는 메모리의 값과 마스크 값을 비교한 1비트 신호를 참조하여 해당 세션의 Inventoried flag 값과 Selected flag 값을 설정한다.

태그를 인식함에 있어 가장 중요한 Query 명령은 Select 명령에서 설정된 flag 값들에 따라 처리

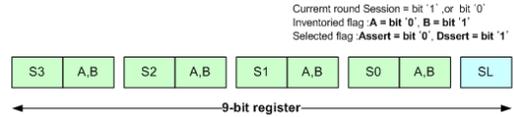


그림 5. 세션과 flag 값 구분을 위한 9비트 레지스터

되거나, Select 명령 없이 Query 명령을 바로 사용하여 리더의 Inventory 동작을 진행 할 수 있다. 이는 불특정 다수의 태그를 인식할 경우 메모리의 값을 모르므로 Select 명령을 사용하지 않고 리더 인식 거리내의 모든 태그를 인식할 때 사용할 수 있다⁶⁾. Query 는 22비트의 고정 길이 명령이지만 CRC5의 검사 결과로 그 명령의 끝을 결정한다. PCU는 Command_clk의 상향 트리거시에 버퍼링 되었던 Query의 명령의 각 필드 값을 처리하며, 하향 트리거시에 슬롯카운터는 RNG로 부터 Q범위($2^Q - 1$)의 값을 가져온다. Query 명령이 처리 되면 Q값과 슬롯 카운터를 활성화시키기 위한 1비트 신호를 정의한다. 다중접속을 위한 일종의 응답 순서인 슬롯 카운터 값이 '0'이면 바로 FSM은 Reply 상태가 되며, 그 외에는 Arbitrate 상태가 된다. FSM은 그림 1에 따라 동작한다.

Query Repeat 명령은 세션 검사 후 FSM이 Arbitrate 상태일 경우 슬롯 카운터의 값에서 1씩 감소한다. 슬롯 카운터 값이 0이었다면 $2^Q - 1$ 로 설정된다.

Query Adjust 명령은 세션검사를 마친 후 Query에서 전송 받아 4비트 레지스터에 저장해 둔 Q값에서 1증가 혹은 1감소시키거나 그 값을 유지 시킨 채 슬롯 카운터의 값을 RNG로부터 변화한 Q범위 내에서 새롭게 받는다.

슬롯카운터의 값이 0이 되면 FSM은 Reply 상태가 되어 전송 컨트롤러를 활성화 시킨다. PCU는

RNG로부터 16bit의 랜덤수를 추출하여 인코더에 16비트 버스로 전달함과 동시에 전송한 RN16으로

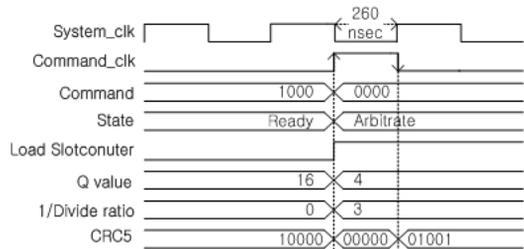


그림 6. Query 명령의 처리

레지스터에 저장해 둔다. 이 후 리더로부터 ACK를 받아 레지스터에 저장된 RN16과 ACK 명령에 실린 RN16을 비교하여 같을 경우 FSM은 Acknowledged 상태가 되고, FSM은 전송 컨트롤러와 메모리 컨트롤러를 EPC를 전송을 위해 활성화 한다.

3.5 슬롯카운터 및 RNG

태그의 응답 순서를 결정하는 슬롯카운터는 RNG의 값이 얼마나 랜덤한지에 따라 태그응답 충돌을 피할 수 있다. 따라서 태그가 작동 되는 순간부터 종료 될 때까지 RNG는 시스템 클록에 따라 랜덤수를 발생시킨다²⁾. 16비트 LFSR(Linear Feedback Shift Register)로 설계한 RNG는 레지스터들의 값이 모두 '0'이 되면 더 이상 값을 생성할 수 없다. 또한 RNG 레지스터의 초기 값이 같은 태그가 있을 경우 태그 응답 충돌의 발생 빈도가 높아 질 수밖에 없다. 따라서 RNG의 쉬프트레지스터가 모두 0이 될 때와 초기 값은 태그마다 유일한 값인 태그 EPC 메모리의 CRC16을 초기 값으로 설정하였다. 슬롯 카운터는 16비트 레지스터로 구성되며 Q값에 의해 RNG의 16비트 출력버스와 연결된다. Query와 Query Adjust 명령이 PCU로부터 처리 되면 Command_clk의 하향 트리거에 새로운 슬롯카운터 값을 설정한다.

3.6 FM0 & Miller 인코더

표준에서는 FM0와 MILLER 방식이 지원 되어야 하지만 시스템 클록이 1.92MHz이므로 지원 가능한 인코딩 방식은 모든 태그 응답 속도에서 FM0는 가능하며 LF=80KHz 이하에서 모든 Miller 인코딩 방식이 가능하다. 인코더의 동작 신호는 FSM으로부터 Reply와 Acknowledged 상태에서 활성화 된다. RN16 값은 FSM이 Reply 상태가 되면 RNG로부터 16비트 값을 PCU가 레지스터에 저장과 동시에 인코더에 전달하며, EPC값은 PCU가 메모리 컨

트롤러를 활성화하여 메모리로부터 96비트 EPC를 인코더에 16비트씩 SCLK 6클록으로 인코더의 96비트 FIFO에 저장한다. 그림 8은 FM0 인코딩을 위한 클록을 시스템 클록으로부터 분주한 파형이며 분주율이 홀수일 경우와 짝수일 경우의 Tsub_clk의 Duty Cycle은 다르지만 태그의 출력인 FM0 파형의 Duty Cycle은 Gen2표준의 50~55%를 만족한다.

IV. 시뮬레이션

VHDL로 설계하여 Xilinx ISE6.3i를 이용하여 합성 및 검증하였으며, Modelsim6.0으로 시뮬레이션 하였다. FPGA 구현을 위한 타겟 디바이스는 xc2v1000으로 약 3만 5천 게이트가 사용되었으나 LF를 구하기 위한 나눗셈 연산에서 2만 3천 게이트가 사용되었다. 테스트 벤치에서 리더 명령의 전송 속도는 160Kbps(데이터'0'기준 Tari=6.25us)이고, 태그는 표준 상 최대 640Kbps (DR=64/3)로 FM0 인코딩하여 응답한다. 시스템 클록(SCLK)은 1.92MHz 단일 클록이며 비동기 리셋 신호를 받는다.

리더의 명령과 명령 사이와 명령과 태그 응답 사이에는 표준에서 정한 링킹타이밍이 있으며 시뮬레이션에서 정의한 표준 상의 최소 링킹타이밍은 표 3과 같다. Select 명령 후 Query는 T4 시간 후 전송되는데, 태그의 응답이 있을 때 까지 T1+T3 간격으로 Query Repeat를 3회 전송하며, 그래도 태그의 응답이 없으면 Query Adjust 명령으로 Q값을 1씩 낮추도록 시뮬레이션 하였다.

그림 8은 Select 1회, Query 1회, Query-Repeat 8회, Query Adjust 2회 후 ACK가 전송되는 시나

표 3. 설계 요약

	4 input LUTs	estimated power consumption
Top design	3,893	2.685mW @1.5V

표 4. 명령어 전송간격 파라미터^[1]

(RTcal =16.65us, FT=15%)

Time	최소 명령어 간격계산(us)	설명
T1	Max(RTcal, 10Tpri) * (1-FT) - 2us	명령 후 태그 응답까지 시간
T2	3 * Tpri	태그 복조시간
T3	T3 >= (T4 - T1)	T1 이후 태그 응답 없을시 기다리는 시간
T4	2*RTcal	명령어 최소간격

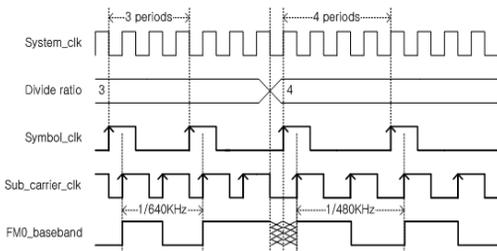


그림 7. 인코딩을 위한 Tpri 클록과 Tsub 클록 파형

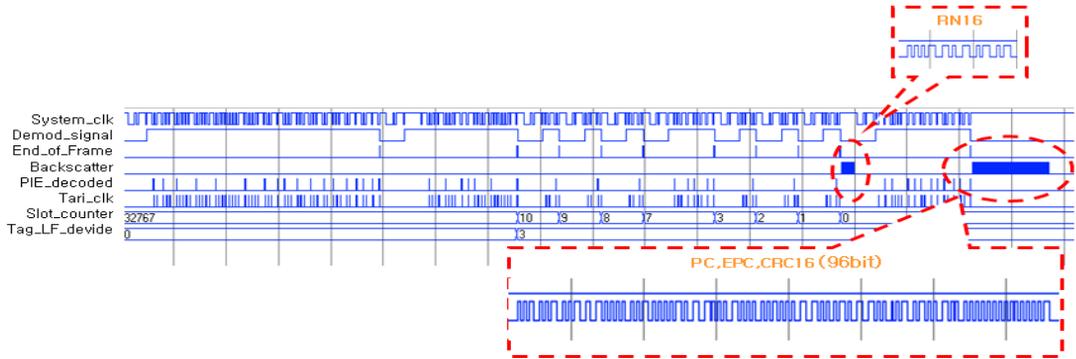


그림 8. 타이밍 시뮬레이션 파형

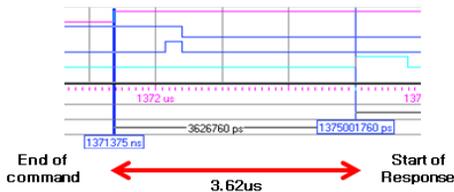


그림 9. 명령 복조 후 응답 지연시간

리오 상황에서 태그 처리과정을 보여주는 Place And Route(PAR) 후 타이밍 시뮬레이션 파형으로, 슬롯카운터가 0이 되면 RN16을 전송하며 ACK로부터 전송했던 RN16을 되받으면 메모리의 96비트 EPC 값을 전송함을 확인 할 수 있다. 그림 9는 리더 명령 처리 후 태그 응답인 FM0 신호 출력까지 시간이 표 4의 T1 시간 값의 21%임을 알 수 있다. 표 4의 링킹 타이밍 값들은 표준의 최소 요구 값이며 보통 더 큰 값을 설정하므로 설계된 태그는 단위 시간 당 더 많은 리더의 명령을 처리 할 수 있게 된다.

V. 결 론

UHF 대역의 태그들은 수 미터의 인식거리로 인하여 다중태그 상황을 고려한 충돌방지알고리즘을 지원해야 한다. 따라서 이를 처리하기 위한 디지털 컨트롤러가 태그의 핵심이라 할 수 있다. 본 설계에서는 900MHz 대역의 수동형 RFID 표준인 EPCglobal Class1 GEN2 프로토콜을 지원하는 기존 논문에서 언급되지 않았던 RFID 리더 명령과 태그 응답 사이의 링킹 타이밍을 최소화하기 위한 디지털 회로를 VHDL 설계와 파형 중심으로 기술하였다. 시뮬레이션 결과 표준의 송수신 데이터 최대 속도에서 최소 링킹타이밍을 80% 이상 단축시키는 성능을 보였다. 또한 수동형 RFID 태그는 응답을 위한 상

태와 응답 속도가 리더에 의해 설정되고 태그는 설정된 값에 따라 정확하게 동작해야만 리더에 인식되기 때문에 이를 처리하기 위한 PIE 디코더와 클럭 분주기 설계 방식을 제안하였으며 리더와 태그의 가변 전송 속도 시뮬레이션 결과 표준의 오차 범위 요구사항을 만족 하였다.

결국 RFID 시스템의 인식률은 리더뿐만 아니라 태그의 성능 역시 영향을 주게 되므로 GEN2 표준과 같은 시분할 랜덤 다중 접속 환경에서 최소화된 응답 지연 시간과 정확한 응답 속도를 지원하는 본 설계의 결과는 RFID 리더가 단위 시간동안 인식하는 태그의 수의 비율인 인식률을 개선시킬 수 있을 것이다.

참 고 문 헌

- [1] EPC Global, "EPC Radio-Frequency identity Protocols class1 generation2 UHF RFID protocols for communications at 860MHz~960MHz version 1.0.9," *EPC global*, 2005
- [2] Klaus Finkenzeller, "RFID Handbook 2nd Edition," *Wiley*, 2003
- [3] Rob Glidden, "Design of Ultra-Low_Cost UHF RFID Tags for Supply Chain Applications," *IEEE Communications Magazine*, Aug 2004.
- [4] Karthaus,U., Fischer, M., "Fully integrated passive UHF RFID transponder IC with 16.7-/spl mu/W minimum RF input power," *IEEE Journal of Solid-State Circuits*, Volume 38, Issue 10, pp. 1602-1608, Oct. 2003.
- [5] He Yan, "Design of Low-power Baseband-processor for RFID Tag," *International Symposium on Applications and the Internet*

Workshops (SAINTW'06), pp. 60-63, 2006.

[6] J.-W. Lee, "Design consideration of UHF RFID tag for increased reading range," *IEEE MTT-S International Microwave Symposium*, pp. 1588-1591, 2006.

[7] Alex K. Jones, "A Field Programmable RFID Tag and Associated Design Flow," *14th IEEE symposium on FFCM*, p.p165-174, 2006.

[8] Andrea Ricci, Matteo Grisanti, Ilaria De Munari, Paolo Ciapolini, "Design of a Low-Power Digital Core for Passive UHF RFID Transponder," *9th EUROMICRO Conference on Digital System Design (DSD'06)*, pp. 561-568, 2006.

[9] 이용주, "EPC RFID 프로토콜 제너레이션2 클래스1 태그 디지털 코텍 설계", *한국통신학회논문지*, Vol.31 No.3A, pp. 360-367, 2006

[10] P. Bernardi, "Design of an UHF RFID Transponder for Secure Authentication," *GLSVLSI2007: ACM 17th Great Lake Symposium on VLSI*, March 11-13, 2007.

[11] Yu Yu, "A novel design of secure RFID tag baseband," in *Proceedings of EU RFID Forum*, 2007.

[12] César Marcon, "A 915 MHz UHF Low Power RFID Tag," *SBCCI2007*, pp. 276-281, Sept. 2007.

[13] Jianwei Wang, "A Novel Anti-Collision Algorithm with Dynamic Tag Number Estimation for RFID Systems," *ICCT*, Nov. 2006.

[14] H. Vogt, "Efficient Object Identification with Passive RFID Tags," *Lecture Notes in Computer Science Springer-Verlag*, vol. 2414, 2002.

[15] Yuusuke Kawakita, "Anti-collision performance of Gen2 Air Protocol in Random Error Communication Link," *SAINT Workshop*, pp. 68-71, 2006.

[16] 김혁, "Real Xilinx FPGA World 8.1," 엔트 미디어, 2006.

[17] H. Roth, "Digital Systems Design using VHDL," *PWS Publishing Company*, 1998.

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

[19] www.impinj.com/files/MR_MZ_TB_00001_TagClockRate.pdf

장 일 수 (Il Su Jang)

준회원



2006년 2월 광운대학교 전자공학과(학사)

2006년 3월~현재 광운대학교 전자공학과(석사과정)

<관심분야> RFID 시스템, 무선통신 시스템

양 훈 기 (Hoon Gee Yang)

중신회원



1985년 연세대학교 전자공학과(학사)

1987년 SUNY at Buffalo ECE(석사)

1992년 SUNY at Buffalo ECE(박사)

1993년 3월~현재 광운대학교 전자공학과 교수

<관심분야> 무선통신시스템, UWB, RFID, 스펙트럼공학