

무선 통신에서 DC 바이어스를 최소화하는 화이트닝 블록 설계

문상국

목원대학교 전자공학과

Design of a Whitening Block Module for Minimizing DC Bias in Wireless Communications

Sangook Moon

Mokwon University, Department of Electronic Engineering

E-mail : smoon@mokwon.ac.kr

요 약

블루투스과 같은 무선통신의 경우, 베이스밴드에서는 송신단과 수신단의 데이터의 모뎀 인터페이스를 통과한 데이터에 대해 보안성 및 회로의 안정성을 위해 DC 바이어스를 제거해주어야 한다. 송신단에서는 송신할 데이터를 랜덤하게 섞어 에러 정정 모듈에 보내주어야 하며 수신단에서는 랜덤하게 평균화된 데이터들을 원래 상태로 복구하여야 한다. 이러한 화이트닝 블록은 필터링을 위한 고유다항식을 목적에 맞게 선택하는 것이 매우 중요하다. 본 논문에서는 하드웨어의 효율성을 높이고 면적을 줄이기 위하여 고유다항식을 $g(D) = D^7 + D^4 + 1$ 로 선택하여 화이트닝 블록을 설계하였다. 설계한 하드웨어 화이트닝 블록은 Verilog HDL로 기술하고 검증하여 자동 합성방식으로 합성하였다. 합성된 화이트닝 블록은 기준으로 삼는 베이스밴드 마이크로컨트롤러의 동작주파수인 40MHz에서 정상적으로 동작하였다.

ABSTRACT

In wireless communications such as Bluetooth, Baseband should be able to minimize the DC bias of the data which passed the modem interface of either transmitter or receiver for the reliability of the circuit and the integrity of the data. The transmitter scrambles the data to send randomly to the error correction block and the receiver recovers the randomly spread data as they have been. To design the whitening block, it is important to select the prime polynomial for the filtering. In this paper, we designed a optimal whitening block using the prime polynomial $g(D) = D^7 + D^4 + 1$ for hardware and area efficiency. The proposed hardware whitening block was described and verified using Verilog HDL and later to be automatically synthesized. The synthesized whitening block operated at 40Mhz normal clock speed of the target baseband microcontroller.

키워드

무선통신, 베이스밴드, 화이트닝, whitening

1. 서 론

현존하는 무선 통신 기술로는 ZigBee, Wi-Fi, 블루투스, 무선 USB, 또한 여러 종류의 핵심 기반 기술로 대두된 UWB (Ultra WideBand) 등 여러 종류가 존재하며, 각기 다른 응용 영역 상에서 무선 네트워크의 표준을 차지하기 위해 치열한 경쟁

을 벌이고 있는 상황이다. 각각의 무선 통신을 수행하기 위해서는 송신단과 수신단의 분리는 필수적이며 양 단에서 신뢰할 수 있는 데이터를 주고 받을 수 있어야 한다. 무선 채널을 사용하여 통신의 수단이 되는 데이터는 0 혹은 1이 변조되어 채널에 실리는데, 이 데이터의 값이 0이 계속되든지 1이 계속 되든지 하는 경우가 발생할 수 있다. 이

러한 경우에, 편중된 연속된 값을 가지고 있는 데이터는 노이즈에 영향을 받기가 쉽기 때문에 이러한 데이터의 편중성을 최소화하기 위한 방안이 필요하다. 무선통신에서 그 역할을 해주는 데이터 처리부가 화이트닝/디화이트닝 블록이다.본 연구에서는 이러한 데이터 화이트닝 처리 모듈을 설계하였는데, 이는 특별히 블루투스 통신방식에 적합하다.

II. 블루투스 베이스밴드

그림 1은 블루투스 베이스밴드의 일반적인 블록 다이어그램이다. 베이스밴드의 세부 블록은 아래와 같이 세분화된다.

- RF 인터페이스
- LMP 인터페이스
- 홉 선택 제어기
- ACL & SCO 링크 제어기
- HEC, FEC 제어기
- 액세스코드 상관기
- 데이터 화이트닝
- 암호화
- 인증
- 오디오 인터페이스
- 클럭 복원과 동기화 모듈
- 채널 제어기
- RX/TX 버퍼와 레지스터

그림 오른쪽에는 RF 모듈이 존재하여서 2.4 GHz ISM 밴드의 주파수를 변조하는 역할을 수행하고, 이로 받은 데이터들은 아래 부분의 타이밍 복원 회로에 의해 1MHz의 샘플링 형식에 맞는 데이터의 스트림으로 입력된다. 입력된 데이터들은 먼저 로우패스 필터를 통과하여 노이즈를 제거하면서 신호의 수행 사이클에 충실히 번역되어 베이스밴드 내로 전달된다. 이 때 64비트 블록 단위로 전송되는 데이터는 데이터의 싱크 검색기에 의해 신호의 문자열이 '1010' 또는 '0101'인지를 감별하여 그것이 데이터 패킷의 시작인지를 인지한다. 성공적으로 인지되어 전달된 데이터는 일정한 블록 형태의 병렬 데이터로 변환되고 변환된 데이터는 헤더와 페이로드 부분으로 나뉘어 각각의 에러 정정 블록으로 전달된다. 헤더 에러 검출 블록에서는 스펙에 정의된 헤더의 에러를 검출하는 알고리즘을 이용하여 헤더의 에러를 검출하여 이후를 진행하고, 페이로드 부분은 화이트닝/디화이트닝 블록에 의해 스크램블링 되었던 데이터들이

미있는 값을 가지면서 FEC (Forward Error Check)의 세가지 모드 (1/3, 2/3, 3/3)에 의하여 에러를 다시 검출하고, [1][2] 간단한 에러 복구 메커니즘에 의해 복구된 데이터는 헤더와 페이로드로 분리되어 베이스밴드 내 메모리에 데이터 블록으로 각각 저장된다. 통신 패킷의 종류는 SCO (Synchronous Connection Oriented)와 ACL (Asynchronous Connection Oriented) 두 종류로 나뉜다. SCO 패킷은 슬롯을 할당하여 주로 음성통신과 같은 어플리케이션에 사용되고, ACL은 유연성을 가지는 데이터 패킷들을 전송하는데 주로 사용된다.

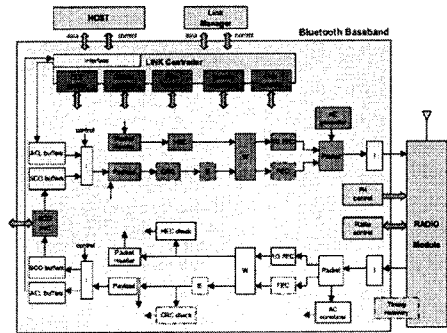


그림 1. 블루투스 베이스밴드 블록 다이어그램
Fig. 1. Bluetooth baseband block diagram

III. 데이터 화이트닝 블록

데이터 전송 전에, 데이터 패킷의 헤더와 페이로드는 화이트닝 seed를 사용하여 스크램블링 하여야 한다. 데이터 패킷을 인여화 하고 패킷 내의 DC 바이어스를 최소화하기 위해서다. 이 스크램블링은 FEC 인코딩을 하기 직전에 이루어져야 한다. 수신단에서는, 송신단과 같은 화이트닝 seed를 가지고 있어서 동일한 방법으로 디화이트닝을 수행한다. 이 디화이트닝은 FEC 디코딩 직후에 이루어져야 한다. 화이트닝과 디화이트닝의 시점은 그림 2와 같이 진행된다 [3].

효율적인 데이터 화이트닝을 위하여 본 논문에서는 블루투스 SIG에서 제안한 고유다항식 $g(D) = D^7 + D^4 + 1$ 을 택하였다. 이는 8진수 표현식으로 221로도 표기가 가능하다. 화이트닝 데이터를 생성하는 선형 귀환 쉬프트 레지스터의 블록 다이어그램은 그림 3과 같다.

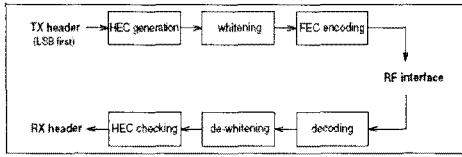


그림 2. 헤더 비트 프로세스
Fig. 2. Header bit process

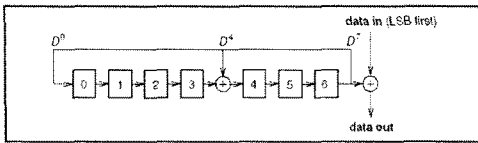


그림 3. 데이터 화이트닝 선형 귀환 쉬프트 레지스터

Fig. 3. Data whitening linear feedback shift register

블루투스의 경우, 그림 3의 쉬프트 레지스터는 전송 이전에 블루투스 표준의 CLK6-1이라는 초기값을 사용한다. 초기화 이후에는, 패킷 헤더와 페이로드가 쉬프트 레지스터를 통과하여 화이트닝 된다.

IV. 구현 및 시뮬레이션

데이터 화이트닝 회로는, 블루투스 표준에서 제안한 고유 다항식 $g(D) = D^7 + D^4 + 1$ 을 사용하여 구현하였다. 그림 4는 화이트닝 시퀀스에 따른 화이트닝 쉬프트 레지스터의 결과값이다.

회로는 HDL을 사용한 탑다운 설계방식을 사용하였다. 회로 소자의 개수가 많지 않기 때문에, 합성 옵션으로 ungroup 옵션을 사용하여 모든 세부 모듈을 최하위 게이트 레벨로 풀어서 합성하였다. FPGA는 Altera StratixII를 사용하였고, Altera에서 제공하는 QuartusII 자동 합성기를 사용하였다. 시뮬레이션의 최악 조건은 2.3V, 섭씨 100도이다.

검증은 C 프로그램으로 구현한 good model의 화이트닝 모듈의 출력과 HDL 시뮬레이터에서 발췌한 결과값이 일치하는지를 1만개의 테스트벡터로 확인하였다. 그림 5는 Altera 합성기를 사용하여 회로로 구현된 게이트와 로직 블록의 사용 회수를 나타낸다.

Whitening Sequence (=D7)	Whitening LFSR D7.....D0
--------------------------	--------------------------

1	1111111
1	1101111
1	1001111
0	0001111
0	0011110
0	0111100
1	1111000
1	1100001
1	1010011
0	0110111

그림 4. 화이트닝 시퀀스와 레지스터의 값
Fig. 4. Whitening sequence and value in the registers

Flow Status	Successful - Tue Oct 14 16:16:40 2008
Quartus II Version	8.0 Build 231 07/10/2008 SP 1 SJ Web Edition
Revision Name	whiten
Top-level Entity Name	whiten
Family	Stratix II
Met timing requirements	Yes
Logic utilization	< 1%
Combinational ALUTs	10 / 12,480 (< 1%)
Dedicated logic registers	9 / 12,480 (< 1%)
Total registers	9
Total pins	21 / 343 (6%)
Total virtual pins	0
Total block memory bits	0 / 419,328 (0%)
DSP block 9-bit elements	0 / 96 (0%)
Total PLLs	0 / 6 (0%)
Total DLLs	0 / 2 (0%)
Device	EP2S15F484C3
Timing Models	Final

그림 5. 합성된 회로의 로직게이트 사용율
Fig. 5. Logic gate usage of the implemented circuit

FPGA 소자로는 StratixII를 사용하였다. 쉬프트 레지스터가 포함되므로 FPGA 내의 레지스터 소자가 사용된 것을 볼 수가 있다. 합성한 회로를 포스트 레이아웃하여 게이트 시뮬레이션을 수행하였다. 시뮬레이션 결과는 그림 6과 같다. 클럭 주기는 100ns를 사용하였으며, 시뮬레이션을 위하여 fed header 부분을 활성화하여 시뮬레이션을 수행

