FLEX 방식 고속무선호출 디코더의
VHDL 모델링 및 구현 연구

박 전*, 이 대원, 김 영철
전남대학교 전자공학과
e-mail : jpark@neuron.chonnam.ac.kr *

Study on the VHDL modeling and Implementation of
a FLEX high speed pager decoder

Jin Park*, Tae-Won Lee, Young-Chul Kim
Dept. of Electronic Eng., Chonnam National University
e-mail : jpark@neuron.chonnam.ac.kr *

Abstract
In this paper, we design a decoder for the FLEX high speed paging protocol. The decoder that we design consists of a synchronizer, a de-interleaver, an error corrector and a packet builder. In the FLEX protocol, a word is coded using a BCH algorithm. In this design, we do not use a look-up table in order to decrease a chip area of the BCH decoder. The simulation result shows that the decoder is correctly designed.

I. 서론

현재 국내에서는 POCSAG 방식과 FLEX 방식이 혼용되고 있는데, FLEX방식은 POCSAG보다 많은 데이터 전송할 수 있으며 고속이기 때문에 FLEX 단말기가 기존의 POCSAG 단말기보다 많은 장점을 가지고 있다고 할 수 있다.

본 연구는 한국과학재단 RRC 지원사업 연구비에 의한 전임

FLEX 방식 고속 무선호출 단말기는 Receiver, A/D Converter, Decoder, Host의 구성된다. 디코더는 무선호출 단말기의 핵심 부품이라 할 수 있으며, 대부분 알고리즘은 공개되지 않기 때문에 국내의 단말기 제조업체 제품이 자체적으로 알고리즘을 개발하여 단말기를 생산하고 있다.

FLEX 프로토콜은 제날상에서 발생한 오류를 정정하기 위하여 디코더에 BCH 이더 정정 알고리즘을 사용하며 최대 2비트의 이더를 정정할 수 있다. 2비트 이상의 이더가 발생했을 경우 checksum을 검사하여 오류를 감지할 수 있으며, 합수 채퍼티를 사용하여 채퍼티 검사를 수행한다.

본 논문에서는 FLEX 고속무선호출 프로토콜을 디코딩하기 위한 알고리즘을 VHDL을 이용하여 모델링 및 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 FLEX고속무선호출 프로토콜의 구조 및 특징에 대하여 알아보고, 3장에서는 프로토콜을 디코딩하기 위한 알고리즘에 대하여 설명한다. 4장에서는 시뮬레이션 결과를 그리고 5장에서 결론을 기술한다.

II. FLEX 무선호출 프로토콜
프레임은 그림 1과 같이 구성되며, 몇몇음향방식으로 1시각당 15개의 사이클이 진행되고, 1사이클은 128개의 프레임으로 구성된다. 한 프레임은 Sync와 11개의 Block으로 구성되며 Sync 정보가 전송된 후 11
개의 클럭이 인터리버 방식으로 전송된다.

그림 1. FLEX 프로토콜의 프레임 구조.

Sync1, Sync2는 FLEX 프로토콜의 동기화를 위해 동기 패턴을 포함하며, FInfo(Identity Information)는 주가, 프레임 번호, 플랫폼 정보를 전송한다. Sync1과 FInfo 정보는 형태가 1600BPS로 전송되고 Sync2가 11개의 Block은 1600BPS, 3200BPS, 6400BPS 속도로 전송될 수 있으며 이에 대한 정보는 Sync1에 전달된다.

BIW의 첫 번째 위치는 Address와 Vector field의 시작위치 정보를 포함하기 때문에 최소한 한 위치가 전송되며, 시장, 날짜와 같은 시스템 정보를 전송하기 위해 3개가 추가적으로 전송될 수 있다.

Address field는 해당 프레임이 전송되는 수소가 들어 있어서 단말기는 이 부분만 검사하면 자기 데이터 유무를 판단할 수 있다. Vector field에는 전송한 메시지의 시작위치와 길이가 포함되어 있고, Message field에는 전송하고자 하는 데이터가 들어가며 사용되지 않는 부분은 Idle field로 정의되어 일정한 패턴으로 채워진다.

III. 디코딩 알고리즘

Synchronizer는 A/D converter로부터 입력된 데이터 양에서 동기 패턴 A, C를 찾아내고 동기 상태에 들어 간다. 동기오도 상태에서 입력되는 데이터는 음이중화 되며, 오류 복구를 위해 Error Corrector에 입력된다.

마지막으로 이 데이터는 Packet builder에 입력되어 host에 전송하기 위한 32비트의 packet을 구성해내고 SPI 비례에 저장된다.

1. Synchronizer

\[
\begin{align*}
\text{clk} & \rightarrow \text{data_rate} \rightarrow \text{Frame Info} \\
& \rightarrow \text{exts} \rightarrow \text{exts} \rightarrow \text{A(0:30)} \\
& \rightarrow \text{BCH} \rightarrow \text{Corrector} \\
& \rightarrow \text{A(0:30)}
\end{align*}
\]

그림 2. Synchronizer.


2. De-Interleaver

\[
\begin{align*}
\text{clk} & \rightarrow \text{reset} \rightarrow \text{data_rate} \\
& \rightarrow \text{exts} \rightarrow \text{exts} \rightarrow \text{L281600} \\
& \rightarrow \text{L282200} \\
& \rightarrow \text{L482200} \\
& \rightarrow \text{L48400} \\
& \rightarrow \text{packet}
\end{align*}
\]

그림 3. 음이중화 작업

FLEX에서는 2회래 또는 4회래 FSI(Frequency Shift Keying) 방식 방식을 사용한다. 따라서 그림 3에서 같이 1600BPS 2회래, 3200BPS 2회래, 3200BPS 4회래, 6400BPS 4회래의 4가지 속도로 구분되며, 이러한 4가지 속도모드 모두 수용하도록 설계한다. 수신되는 정보는 음이중화되어 비례에 적합한 후 Error Correction을 수행한다.

3. Error Corrector
FLEX 방식 고속우선호출 디코더의 VHDL 모델링 및 구현 연구

하나의 워드는 4비트의 checksum, 17비트의 데이터, 10비트의 BCH 파리티, 마지막으로 1비트의 합수 패리티로 이루어진다. Checksum과 데이터는 모두 21비트이고, BCH 고정값과 생성된 패리티 10비트를 추가하면 31비트가 되기 때문에 BCH(31, 21) correction 이라 한다.

![diagram]

그림 4. Error Corrector.

에러 결정과 검증은 3단계의 과정을 통해 이루어진다. 2비트의 오류를 정확히 위해 BCH(31, 21) 디코디어 이루어지며, 합수 패리티 검사를 수행한다. 3비트 이상의 오류가 발생하였을 경우에는 BCH 단에서 정확하지 못하기 때문에 bch_error 출력을 1로 설정하고 패킷 생성시에 해당 오류패러미터를 생성하여 Host에 전송한다.

본 논문에서는 BCH오류 정정에 있어서 Look-up table을 사용할 경우 ROM이 필요하기 때문에 단말기의 소형화를 위해 Look-up table을 사용하지 않고 Galois field 공식, Chien 알고리즘을 이용한 오류정정 방법을 사용하였다.

4. Packet Builder

주파수에 담긴 16비트의 CRC와 함께 디코딩된 데이터를 포함한 로질리스 또는 데이터를 포함한 프레임(frame)과 위상(phase)을 찾아낸다. 수신된 어드레스가 프레임로부터 계산된 값과 다를 경우 비동기모드(sync=0)에 돌리며, 다음 프레임이 수신될 때까지 디코딩은 중단된다. 어드레스의 범위를 조사하여 어드레스의 종류에 관한 정보(LA: 높 어드레스, TO: 높 은 어드레스)를 Packet Builder 블록에 전달한다. 어드레스에 데이터(cor_data)로부터 해당파라미터의 값을 재임하고, 패킷 ID와, 워드 번호를 부여하면 32비트의 패킷이 생성된다.

IV. 시뮬레이션 결과

1. BCH(31, 21) 에러 정정 블록

BCH(31, 21) 에러 정정 블록은 신호함을 계산하기 위해 Galois field, 에러 위치 다항식 계수 계산, 에러 위치 계산부로 구성되어 최대 2비트의 오류를 정확하게 선정한다.

다음은 오류가 발생한 데이터를 복구해내는 시뮬레이션 결과이다. 정확한 데이터(31비트)가 모두 출력되기 위해서는 입력 데이터 값이 2배에 해당하는 큐터(31+2+1=63비트)가 필요하다.

![diagram]

그림 6. 1비트의 오류가 발생한 데이터 복구.

2. 디코더의 시뮬레이션

그림 7은 1600BPS 2배역 FSK 방식으로 출력된 바탕이 디코딩되어 신호에 입력되기까지의 시뮬레이션 결과이다. 통합단위로 처리하기 때문에 10000과 15000 부분에

- 375 -
시 SPL_PACKET의 값이 출력된다. SYNC매전을 디코딩 후 SYNC_MODE의 값이 HIGH 상태로 되며, frame information으로부터 우선순위 어드레스(PRIORITY_ADDR, NUMBER)의 수는 4, BIW(BLOCK_END)의 수는 2, VECTOR의 시작(VECTOR_START)은 웨도 13을 결과로부터 봤 수 있다. 그리고 벡터필드가 우선된 후 메시지(MSG_START)는 22에서 시작되며 길이(MSG_LENGTH)가 4임을 그림에서 16진수로 그 값을 보여주고 있다.


V. 결론

본 논문에서는 고속무선호출 프로토콜을 디코딩하기 위한 알고리즘을 VHDL로 모델링하였다. BCH에 의해 정해진 연산을 효율로 하기 위해 ROM이나 PLA를 사용하지 않고 디코딩 알고리즘을 분리하여 설계하는 방법을 제안한다. 현재 무선호출은 단방향에서 양방향 서비스로 변하고 있다. 양방향 디지털기의 경우 수신단면이 추가되어야 하기 때문에 성능을 높이는 방법이 필요하다. 그리고서와 MCU의 원 적용이 필요할 것으로 보인다. 제안한 알고리즘은 원 원 연구에 적합하게 사용될 수 있을 것이다.

참고문헌