

# Pager 동작 시간 향상을 위한 POCSAG Signal Decoder의 설계

正會員 崔 鍾 文\*, 金 永 大\*\*, 韓 廷 滋\*

## Design of POCSAG Signal Decoder for Operating Time Improvement in Pager

Jong-Moon Choi\*, Young-Dae Kim\*\*, Jung-Il Han\* *Regular Members*

### 요 약

본 논문에서는 Pager의 동작 시간을 향상시키기 위한 POCSAG Signal Decoder를 설계하였다. 송신단에서 보내어진 POCSAG 신호 형식을 제시하였고, 이 디코더의 동작에 대하여 언급하였다. 그리고 이 디코더를 이용한 Pager는 광역 기능이 지원되는 것을 보였으며 이러한 동작을 구현하는 하드웨어를 설계하였고, ASIC 칩으로 구현하였다. 구현된 ASIC 칩의 기능을 검증하고 성능을 테스트하여, 낮은 전압으로 구동되고 전력 소모가 적음을 확인하였다.

### ABSTRACT

In this paper, we designed POCSAG Signal Decoder to improve operating time in pager. We showed POCSAG Signal Pattern sent by transmitter and operation of this decoder. We also showed that the Pager using this decoder was equipped with Wide Area Signal Detection and designed the hardware which realizes this operation and implemented it with ASIC chip. As we inspected the function of the ASIC chip and tested the performance, we could find that the chip operated in low voltage and that power dissipation was low.

### I. 서 론

현재 이동통신 분야는 수요가 폭발적으로 증가하고 있는 추세이다. 그중에서도 Pager는 현재의 이동통신 분야를 대표할만큼 널리 보급되어 있다. Pager는 송신단에서 보내어진 POCSAG(Post Office Code Standardization Advisory Group) 신호를 수신하여 자신의 ID인지 판단하는데, 수신된 값이 자신의 ID 코드이면 Pager는 소유자에게 자신이 호출받은 사실이 있음

\*延世大學校 電子工學科  
Dept. of Electronic Engineering, Yonsei University  
\*\*LG 반도체 ASIC 설계실  
論文番號: 96301-0917  
接受日字: 1996年 9月 17日

을 알리고 받았던 메시지를 소유자에게 보여준다<sup>11)</sup>. 이때 Pager 내부의 Processor(MCU)는 순간순간 수신한 값이 자신의 ID인지 살펴보고 수신된 값을 처리하기 위해 항상 동작중에 있어야 하지만 이 경우 건전지 소모량이 많아지게 된다. 이러한 문제 때문에 RF 수신 회로와 MCU 사이에 항상 동작중에 있으면서 RF 회로에서 넘어오는 수신된 값을 처리하고 자신이 호출된 경우에만 MCU를 작동시켜 필요한 처리를 하는 장치의 개발이 요구된다<sup>11)</sup>. 이러한 역할을 수행하기 위해 POCSAG Signal Decoder를 설계하였다.

Pager에서 송신단에서 보내는 신호는 POCSAG 신호인데 이것은 Preamble과 여러 개의 Batch로 나뉘어진다. Batch는 Synchronization 코드와 여러 개의 Frame으로 나누어진다. 이러한 POCSAG 신호를 입력으로 받아들이면서 Pager 디코더의 초기화가 이루어지고 Preamble과 Synchronization 코드를 사용하여 동기를 맞추고 Frame에 들어있는 ID 코드와 메시지를 인식하게 된다. POCSAG Signal Decoder는 출력된 BS3, BS2, BS1이 RF 회로의 전력 스위치쪽으로 연결되어 RF 회로에 간헐적으로 전력을 공급하여 Pager의 전력 소모를 줄여주는 역할을 한다.

본 논문에서는 POCSAG 신호 형식과 POCSAG Signal Decoder의 기본 동작을 기술하였으며, 이에 적합한 하드웨어를 설계하고 ASIC 칩으로 구현한 결과를 나타내었다. 소모 전류를 최소화하기 위해 BCH 에러정정 부분을 효율적으로 설계하여 게이트 수를 적게 하여 칩 크기를 감소시켰고, 출력 버퍼 및 오실레이터 부분은 기존의 ASIC용 라이브러리에서 적은 Loading과 구동 능력을 가지는 시스템에 사용하기 적합한 라이브러리가 없어 본 ASIC 칩에 적합한 라이브러리를 HSPICE로 시뮬레이션하여 새로 구성하여 사용하였다<sup>10)</sup>.

## II. POCSAG 신호 형식

POCSAG 신호의 구성은 그림 1에 나타나 있고, 처음에 Preamble이 몇 bit 동안 나온 후 각각 17개씩의 워드로 되어 있는 Batch들이 나온다. N개의 Batch 후에는 Preamble이 다시 나오게 되며 이러한 과정은 끝없이 계속된다<sup>11)</sup>.

각 Batch의 구성은 처음 32개의 기본 bit로 구성된

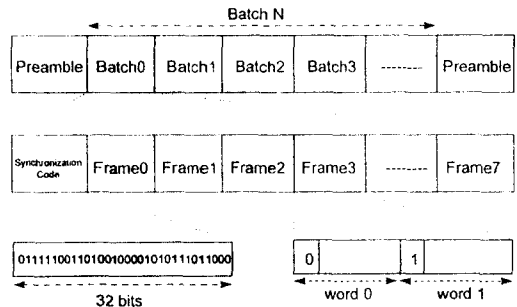


그림 1. POCSAG 신호의 구성  
Fig. 1 The Formation of POCSAG Signal.

Synchronization 코드가 있고 각각 2개씩의 워드로 구성된 8개의 Frame으로 되어 있다. 마지막 8번째 Frame 뒤에는 다음 Block의 Synchronization 코드가 위치하게 되는데 이 값은 전 영역에 걸쳐서 0111110011010010001010111011000인 같은 값이 사용된다.

N개의 Batch 다음에는 다시 Preamble이 나타나며, 실제 사용자가 입력하는 값들은 각각의 Frame에 실리게 된다. 각 신호의 특성을 구체적으로 살펴보면 다음과 같다.

### 2.1 Preamble 및 Synchronization 코드

Preamble은 송신측과 수신측의 동기를 맞추기 위해 송신측에서 보내는 신호로 각 통신 속도에 따라 0과 1의 연속적인 반복으로 나타난다. 수신측에서는 이 신호가 나타날 때 자신의 클럭을 송신측의 클럭에 맞춘다.

Preamble은 Synchronization 코드가 나타남으로써 끝나게 되는데 이 신호는 Preamble의 끝임을 알림과 동시에 앞에서 나온 Preamble에 의해 동기화된 신호의 동기가 계속 맞고 있는지 확인하기 위해 쓰인다. 즉, 수신측에서는 매 일정 시간 Synchronization 코드가 나타나는 시간을 기억하며 제 시간에 이 신호가 나타나는지 확인하여 만약 나타나지 않으면 자신의 수신 시스템이 송신측과 동기가 맞지 않는 것으로 판단하여 다른 조치를 취하게 되는 것이다.

### 2.2 Batch

각 Batch는 Synchronization 코드 1개와 8개의 Frame

으로 구성되어 있다. Synchronization 코드는 전 영역에 대해서 같은 값이 나오며 각 Frame은 2개의 워드로 구성되어 있는데, 각 워드는 32개의 bit들로 구성되어 있다. 각 워드는 어드레스, 메세지, Idle 코드로 나타난다.

어드레스 워드의 32 bit 중 처음 ID\_CODE bit를 제외하고 연속되는 (18 + 2) bit가 실제 Pager의 호출번호가 된다. (18 + 2) bit 중 앞의 18 bit가 호출번호, 이어서 나타나는 2 bit는 Pager의 Tone 형식을 송신측에서 결정해 송신함으로써 결국 Pager의 수신 형태를 송신측에 제공하기 위한 bit이다. 앞의 21 bit 뒤로 연속되는 10 bit는 송수신중 잡음 및 오수신에 의해 잘못 수신된 bit들의 정정 및 수신된 데이터의 포기를 판단하는데 쓰이며 여기에서는 BCH coding 방식이 쓰인다. 마지막 1 bit는 Parity bit이다.

메세지 워드에서 첫째 bit는 메세지인지 어드레스인지 판단하기 위한 ID\_CODE bit이고 연속되는 20 bit가 실제 메세지이며 다음 10 bit가 BCH coding에 의해 늘어난 bit이고 마지막 1 bit는 Parity bit이다.

ID\_CODE bit가 '0'일 때는 어드레스 워드를 나타내고 '1'일 때는 메세지 워드를 나타낸다.

Idle word는 01111010100010011100000110010111로써 일정하다.

2.3 에러정정

1) BCH 에러정정

어드레스 워드와 메세지 워드는 32 bit로 구성되어 있는데, 앞의 21 bit는 유효한 데이터이고 나머지 11 bit는 checker bit이다. BCH 코드에서 앞의 유효한 21 bit의 데이터를 생성 다항식에 의해 나누어진 나머지 값을 BCH checker bit라 한다. 디코더에서의 수신은 31 bit의 어드레스 워드나 메세지 워드를 생성 다항식

에 의해 다시 나누어 남은 나머지 값에 의해 에러가 난 bit의 위치를 알아내고 이를 정정한다<sup>10-13)</sup>. BCH 에러정정은 단지 1 bit 또는 2 bit의 에러에 대해서만 에러정정을 제대로 수행한다. 그외의 에러에 대해서는 보장하지 못한다.

2) Parity 에러

본 논문에서는 Even Parity bit를 두어 전체 bit의 1의 합계를 구함으로써 에러를 검출하고 있다. Parity check는 BCH 에러정정 후에 이루어진다. 만일 Parity를 제외하고 3 bit 이상의 에러가 발생하면 어드레스 워드는 무시되고 메세지 워드인 경우 무시되거나, 에러 코드를 CPU로 보낸다.

사용된 에러정정은 (31, 21, 2)의 BCH 에러정정으로, 이때의 생성 다항식은  $x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1$ 이다.

본 논문에 사용된 에러정정 회로는 다음과 같은 원리로 설계되었다.

그림 2에서 31 bit 중 1 bit의 에러가 일어날 경우의 수는 31가지이고 31 bit 중 2 bit의 에러가 일어날 경우의 수는 465가지이다. 그러므로, 총 에러가 일어날

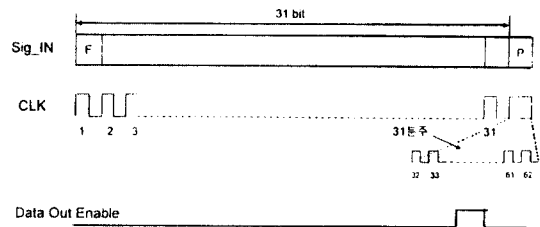


그림 2 BCH Decoding 블록의 타이밍도  
Fig. 2 Timing Diagram for BCH Decoding Block.

표 1. 에러정정의 특성

Table 1. Characteristics of Error Correction.

BCH error	Correction	Parity error	Address	Message	Error code	
					E	P
0 bit	NO	YES/NO	수신	Error code 전송	0	1/0
1 bit	YES	NO	수신	Error code 전송	0	0
1 bit	YES	YES	수신	Error code 전송	0	1
2 bit	YES	NO	수신	Error code 전송	0	0
2 bit	YES	YES	수신	Error code 전송	0	1
기타	YES/NO	YES/NO	무시	Error code 전송	0/1	0/1

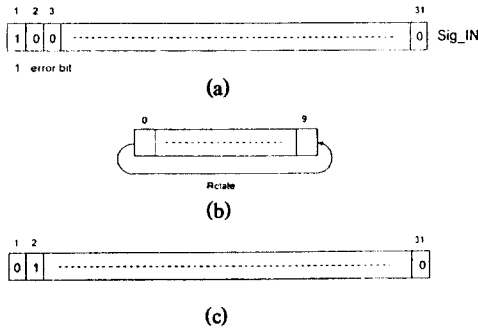


그림 3. 에러정정의 예  
Fig. 3 The example of Error Correction.

경우는 496가지이므로 이 경우의 수에 대한 신드롬을 구하여 에러정정을 수행하면 되겠지만 다음과 같은 방법을 사용하면 회로를 간단하게 설계할 수 있다.

예를 들어, 1 bit 에러인 경우의 수 31가지의 에러에 대해서 1개의 신드롬으로 발견하는 방법을 나타내면 다음과 같다.

그림 3(a)와 같이 첫 bit가 에러인 경우 32 클럭 때 신드롬은 0110111011이다. Sig\_IN으로 입력되는 데이터는 31 클럭 후에는 마스크 되어진다. 그러므로, 계산회로(checker 회로)의 Register Map을 그려 보면 그림 3(b)와 같다.

31 클럭 동안 생성 다항식에 의해서 나누어진 나머지(신드롬) 값이 Shift Right Rotate 하면서 31 부클럭(BCH Decoder Block이 Main 1 Clk timing) 동안 다시 나누어진다.

그림 3(c)와 같이 2번째 bit가 에러인 경우 위와 같은 데이터를 BCH 디코딩하면 32 클럭 때 1000000000의 신드롬 값이 남는다. 33 클럭 때의 나머지는 0110111011이다.

첫 bit가 에러 bit인 경우는 32 클럭 때 나머지 값이 0110111011이고, 2번째 bit가 에러인 경우 33번째 나머지 값이 0110111011이다. 이때 31번째 bit가 에러인 경우 62번째 클럭 때 0110111011의 나머지 값이 남는다.

이와 같은 원리를 이용하여 31가지의 1 bit 에러가 일어날 경우 정정하는 회로는 신드롬 0110111011에 대해서만 고려하면 되므로 ROM Table 없이 간단하게 설계할 수 있게 된다.

이 방법을 이용하여 2 bit 에러일 때 경우의 수 465 가지 중에서 30개의 신드롬 값에 의해 모두 에러를 정정할 수 있다.

### III. 기본 동작

각 핀에 해당되는 입출력 신호의 사양은 표 2와 같다. POCSAG Signal Decoder의 블록도는 그림 4와 같

표 2. 디코더의 입출력 신호  
Table 2. I/O Pin Description of Decoder.

Pin Number	Pin Name	I/O	Pin Description
1	NC		Not Connection
2	ADD_DET B	O	Address Detection시 "Low" 출력
3	RX_CLK B	O	수신된 데이터를 CPU로 전송시 사용되는 클럭
4	RX_DATA	O	수신된 데이터를 CPU로 전송하는 출력
5	SY_VALID	O	Sync Code Detection시 "High" 출력
6	RESET B	I	하드웨어 리셋 입력
7	ROM_CLK	I	ID ROM의 데이터를 읽기 위한 클럭
8	ROM_DATA	I	ID ROM의 데이터를 읽어 들이는 입력
9	TEST	I	Test Pin(정상 동작시 개방되어 있다.)
10	VSS		Ground
11	NC		Not Connection
12	MTSTOP	I	ID ROM의 데이터 중지 신호
13	XIN	I	외부 클럭 입력 Pin
14	XOUT	O	외부 클럭 출력 Pin
15	WADB	I	광역 모드에서 사용된다.(정상동작시 "H")
16	SIG_IN	I	RF 단의 신호 입력 Pin
17	BS1	O	Battery Saving을 위한 출력
18	BS2	O	Battery Saving을 위한 출력
19	BS3	O	Battery Saving을 위한 출력
20	VDD		Power 입력 Pin

다. 전체 입출력 신호는 표 2에 나타난 바와 같고, 내부의 PLL 블록은 RF 블록으로부터 들어오는 데이터 SIG\_IN으로부터 입력되는 데이터를 취하는데 기본적으로 1 bit의 데이터가 32 분주의 주 클럭으로 동작하므로 디지털 PLL 역할을 하는 로직을 사용하여 입

력 데이터를 취하여 SIN 신호를 에러정정 블록에 출력시킨다. 에러정정 회로는 수신된 SIN 신호를 2.3에서 나타난 바와 같이 BCH 에러정정을 수행하여 ECCOUT으로 출력시킨다. Rom Data Register 블록은 외부의 IDROM으로부터 들어오는 데이터를 저장하는 레지스터이다. Timing Control 블록은 전체 디코더의 타이밍을 제어하는 블록이며, RST 신호를 출력으로 내보내어 Rom Data Register 블록을 초기화시킨다. Preamble & Sync Code Detection 블록은 입력된 ECCOUT 신호로부터 Preamble과 Sync를 확인한 후, LastID(SIG\_IN)를 출력하여 Data Comparator 블록의 입력이 되도록 한다. Data Comparator 블록은 SelID와 LastID를 비교하여, 일치하면 MCU로 모든 출력을 내보낸다.

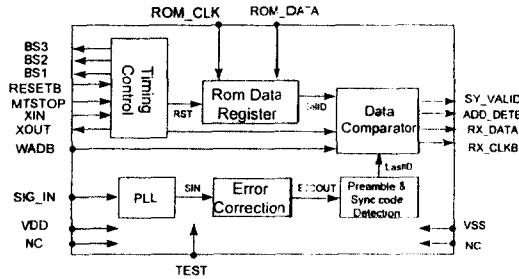


그림 4. POCSAG Signal Decoder의 블록도  
Fig. 4 Block Diagram of POCSAG Signal Decoder.

본 논문에서 구현한 Pager 디코더의 동작 순서는 그림 5와 같이 순서도(Flow chart)로 나타내었다.

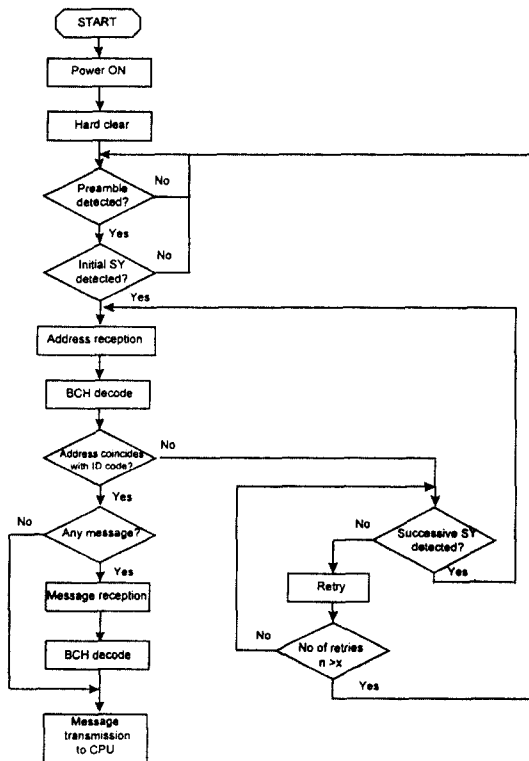


그림 5. Pager 디코더의 순서도  
Fig. 5 Flow Chart of Pager Decoder.

### 3.1 초기화

처음 RESETB 핀에 '0'이 입력되면 특정 Flip/Flop 및 Latch들은 특정값으로 초기화된다. 특정 기간 리셋 동작이 일어난 후 ROM\_DATA는 그림 6처럼 ROM\_CLK 핀을 통해 들어온 클럭의 Rising Edge에 동기되어 내부 직렬 Register 열 즉, 내부 Flip/Flop에 차례로 저장된다.

ROM\_DATA에 입력되어 내부 Flip/Flop에 저장된 값들은 디코더에 다음 리셋이 나타날 때까지 디코더의 모든 동작 방식을 결정하며, ROM\_CLK에 120 클럭이 공급되면 내부의 ROM\_DATA 쪽에 마스킹이 일어나게 되어 이후 ROM\_CLK과 ROM\_DATA에 입력된 값들은 무시된다.

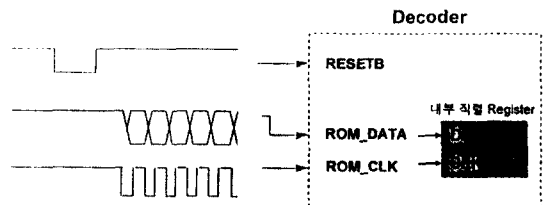


그림 6. 디코더의 초기 동작  
Fig. 6 Initial Operation of Decoder.

### 3.2 POCSAG 신호 Preamble 수신

디코더는 크게 두 개의 동작 모드를 가지고 있다. Preamble 수신 모드와 Batch 수신 모드가 있다. 우선 Preamble 수신 모드는 그림 7에 나타난 바와 같이 디코더의 출력핀 BS3, BS2, BS1이 RF 회로의 전력 스위치쪽으로 연결되어 RF 회로에 간헐적으로 전력을 공급하며 이러한 동작은 결국 Pager의 전력 소모를 줄여준다.

BS3, BS2, BS1이 '1'일 때 RF 회로에 전력이 공급되고, 이 RF 회로에서 복조된 POCSAG 신호는 디코더의 입력 SIG\_IN 핀의 입력이 된다. 디코더는 SIG\_IN에 입력된 값 중 11개 이상의 bit가 0과 1을 규칙적으로 반복할 때 그 입력된 신호를 Preamble로 간주하여 BS3, BS1이 '0'으로 되더라도 '0'으로 되지 않고 Synchronization 코드를 발견할 때까지 계속 '1'로 남아 있다. 이에 따라 RF 회로에 전력이 계속 공급됨으로써 RF 회로는 신호를 계속 받아 복조를 하며 입력되는 값이 Synchronization 즉, 01111100110100100001010111011000일 때 Synchronization 코드의 마지막 bit를 받고 자신의 출력핀 SY\_VALID 핀을 '0'에서 '1'로 바꾸어 준다.

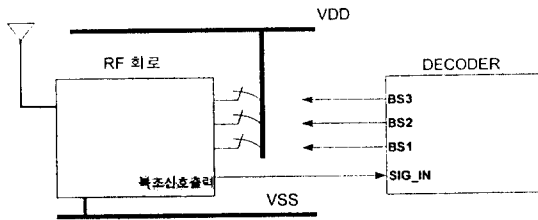


그림 7. Preamble 수신  
Fig. 7 The Receipt of Preamble.

### 3.3 POCSAG 신호 Batch 수신

Batch 수신 모드에서는 Synchronization 수신과 Frame 수신으로 구분된다.

#### 1) Synchronization 코드 수신

비동기 통신에서의 큰 문제점 중의 하나가 바로 송신측과 수신측과의 동기화이다. 따라서 POCSAG 신호에서는 수신측에서 자신과 송신측과의 동기를 확인하기 위해 송신측에서 매 Batch의 머리 부분에 동기 코드를 첨가하여 송신한다.

디코더는 송신측과 수신측의 타이밍이 제대로 맞았을 때 Frame 수신 모드와 같이 동작하지만 타이밍이 제대로 맞지 않았을 때 즉 각 Batch의 시작 부분에 있는 Synchronization 코드를 매번 확인하고 틀렸을 때 다음과 같이 동작한다.

먼저 디코더는 초기화 후 순서도에서와 같이 ROM\_DATA를 통해 사용자가 입력한 사용자 정의 bit들 중 R3-R0에 있는 값에 지정된 Retry 횟수만큼 Synchronization 코드를 찾고 마지막까지 없으면 Preamble 찾기 모드로 돌아가 Preamble을 다시 찾는다.

표 3에 쓰여진 횟수만큼 Synchronization 코드를 찾고 결국 Synchronization 코드가 없으면 다시 Preamble을 찾는다. 수신된 값 중 Synchronization 코드 역시 Noise 및 오수신에 의해 값이 바뀔 수도 있다. 디코더는 수신된 전체 코드를 Synchronization 코드와 비교하여 2개 bit까지 값이 틀리더라도 그 코드가 Synchronization 코드와 같으면 받았던 값은 Synchronization 코드로 인식을 한다.

표 3. 사용자 지정 Retry Bit  
Table 3. Retry bit assigned by user.

R3	R2	R1	R0	Retry 횟수
0	0	0	0	2
0	0	0	1	3
0	0	1	0	4
0	0	1	1	5
0	1	0	0	6
0	1	0	1	7
0	1	1	0	8
0	1	1	1	9
1	0	0	0	10
1	0	0	1	11
1	0	1	0	12
1	0	1	1	13
1	1	0	0	14
1	1	0	1	15
1	1	1	0	16
1	1	1	1	17

#### 2) Frame 수신

디코더가 Preamble을 찾은 후 첫 Synchronization 코드를 받을때부터 Frame 수신 모드인데 이는 BS3, BS2, BS1과 밀접한 관계가 있다.

Frame 수신 모드에서는 직접 송신된 데이터를 받는 부분으로 전 Frame 구간 동안 데이터를 받는 것이 아니고 사용자 정의 bit의 Frame 부분(즉, 처음에 디

코더를 초기화할 때에 내부직렬 Register에 저장된 부분에 사용자가 지정한 시간에 BS1, BS3를 '1'로 만들어 RF 회로를 활성화시켜 데이터를 받을 수 있도록 한다. 그림 8은 사용자 정의 Frame 부분이 0부터 7중 3인 경우로 만약 받을 데이터가 계속되면 점선의 부분처럼 BS 신호는 계속되어 RF 회로로 하여금 계속 데이터를 받을 수 있게 한다. 만약, 사용자가 지정한 Frame의 값이 0일 때 BS1, BS3는 Synchronization 코드를 받아 '1' 상태에서 직접 '0'으로 되지 않고 그 Frame 동안 계속 '1'을 유지하여 이어지는 데이터를 놓치지 않고 계속 받을 수 있도록 한다.

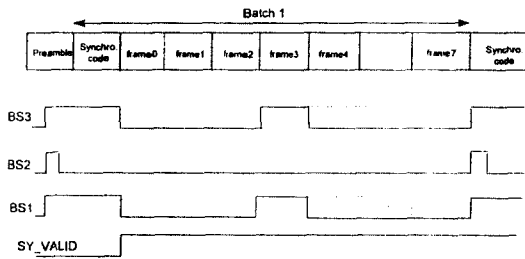


그림 8. Frame 수신  
Fig. 8 The Receipt of Frame.

그림 8에서 BS3, BS1의 점선으로 된 부분은 수신되는 데이터의 양에 따라 그 지정된 Frame이 지나도 계속 '1'로 되어 있다는 것을 나타내며, Batch1에서와 같은 상황은 Batch2, Batch3, ...으로 계속된다.

### 3) 어드레스, 메시지 수신

디코더는 출력핀 SY\_VALID를 '1'로 바꾼 후 사용자가 명시한 Frame에 BS3과 BS1을 '1'로 바꾸어 주며 RF 회로를 활성화시킨다. RF 회로에 의해 복조된 신호는 디코더 내부의 비교기에 의해 자신의 어드레스 코드인지 확인된다.

확인 중 만약 자신의 ID이면 그림 9처럼 출력핀 ADD\_DETБ를 '1'에서 '0'으로 바꾸어 준다. 디코더는 일단 자신의 어드레스 코드가 발견되면 자신의 메시지를 다 받을때까지 RF 회로의 전력을 끊지 않으며 ADD\_DETБ핀도 계속 '0' 상태로 남게 된다. 그림 9는 디코더의 사용자 정의 Frame bit가 "010"으로 되어 있을 경우이며 첫 Synchronization 코드를 받고 A

부분에서는 자신의 어드레스 코드가 아닐 때 이 부분은 무시하고 B 부분에서 자신의 어드레스 코드를 발견하여 메시지 코드가 끝날때까지 신호를 수신하는 경우를 나타내었다.

Batch는 Synchronization 코드와 8개의 Frame으로 구성되고 Frame은 어드레스, 메시지 및 Idle 코드로 구성된다. 어드레스 코드는 그림 10처럼 구성되어 있다.

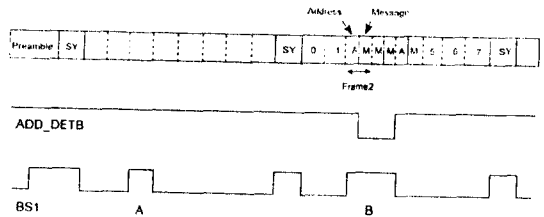


그림 9. 어드레스, 메시지 수신  
Fig. 9 The Receipt of Address and Message.

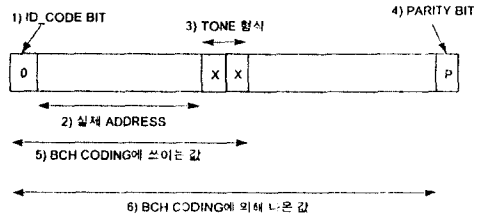


그림 10. 어드레스 코드의 구성  
Fig. 10 The Formation of Address Code.

그림 10중 1)번과 2)번의 내용은 초기 Reset 이후 외부에서 ROM\_CLK 핀에 공급된 클럭에 따라 ROM\_DATA 핀에 직렬로 입력된 값 중에서 어드레스 부분을 나타낸다. 이 입력된 값은 내부 Register에 저장되어 그림 9처럼 각 Batch에서 자신의 Frame에 자신의 어드레스 코드가 있는지 내부에서 계속 비교하게 된다. 실제 어드레스는 18개의 bit이고 앞의 1)번 1 bit는 자신의 ID를 나타내며 3)번은 메시지 수신 여부를 알리는 방법을 지정하기 위한 bit이다. 21개의 bit가 BCH Coding에 쓰이며 이러한 BCH Coding에 의해 나온 bit 수는 29개이고 4)번은 Even Parity로서 전체 Parity를 맞추기 위해 쓰인다.

메세지는 그림 11처럼 구성되어 있으며 실제 메세

지 20 bit와 ID\_CODE 1 bit을 합하여 BCH Coding에 이용하고 BCH Coding에 의해 나온 값 29 bit와 Parity 1 bit을 합하여 메세지 코드로 이용한다.

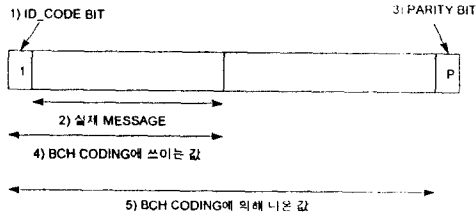


그림 11. 메세지 코드의 구성  
Fig. 11 The Formation of Message Code.

디코더는 지정된 Frame에 자신의 어드레스 코드가 맞을 때 메세지를 받아들인다. 만약 연속하여 2개 이상의 메세지가 수신될 때 BS1과 BS3를 계속 '1'로 유지시켜 주며 이것은 처음 받은 코드의 첫부분, 즉 ID\_CODE bit이 '0'이 나올 때까지 지속된다.

또한 메세지 수신 중 Synchronization 코드가 있어야 될 때, 받았던 값이 Synchronization 코드인지 확인하고 계속 메세지 수신을 하며 이때 받았던 Synchronization 코드는 수신 메세지 내용에서 제외된다.

#### IV. 광역기능

Pager가 광역 기능을 지원하기 위하여 WADB 신호를 사용한다<sup>1)</sup>. WADB 신호는 "Low"로 활성화되

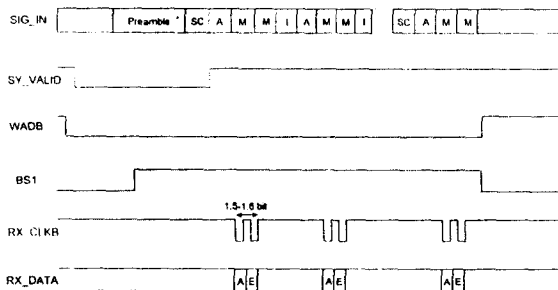


그림 12. 광역 기능에서의 Timing도  
Fig. 12 Timing Diagram for operating in Wide Area.

어 있고, SY\_VALID 신호가 rising edge가 될 때 광역 detect mode가 된다. 광역 detect mode는 WADB 신호가 활성화되어 있는 동안 전체 frame 내의 모든 어드레스를 RX\_CLKB와 RX\_DATA 핀을 통하여 CPU로 에러정정된 어드레스를 전송한다. 이때 수신된 어드레스는 에러가 없거나 에러정정이 완료된 데이터이고, "High"이면 CPU에서 이 데이터를 무시한다. 광역 기능 해제는 WADB 신호가 "High"가 되면 수행된다. 이의 timing은 그림 12와 같다.

#### V. ASIC 칩 구현

본 ASIC 칩은 앞에서 설명한 기능을 구현하고자, 각 사양에 따라 COMPASS Tool을 사용하여 논리 회로를 설계하고 시뮬레이션하여 기능을 검증한 후 칩을 제작하였다. 이때 사용한 Technology는 0.8 $\mu$ m CMOS로 CBIC(Cell Base IC) 방식으로 Layout을 수행하였으며, chip size는 1770 $\mu$ m $\times$ 2270 $\mu$ m이며, package type S\_Dip(Shrink Dip)를 사용하여 제작하였다. 이와 같이 제작된 칩은 그림 13과 같이 응용 회로를 기관 보드로 제작하여 실장 실험을 수행하였다. 그림 14는 제작된 Pager 디코더를 장착한 전체 Pager 보드 모양을 나타내는 사진이다. 원하는 모든 사양이 정상 동작함을 확인하였다. 그림 15는 측정된 Pager 디코더의 동작 특성을 나타낸다. 512bps, 1200bps, 2400bps 각각의 경우에 대한 측정은 POCSAG 인코더에서 측정 속도에 맞게 송신하여 주고, 디코더에서 측정하고자 하는 속도의 모드를 미리 ROM 데이터에 의해 모드 설정을 하고 동작을 확인하였다. 휴대용 기기에서 가장 중요한 소비 전류가 동작 전압 3.0V에서 최대 100 $\mu$ A 이하의 좋은 특성을 얻을 수 있었다. 512bps에서 수정 발진기 38.4kHz를 사용하여 25 $\mu$ A, 1200bps에서 수정 발진기 76.8kHz를 사용하여 32~33 $\mu$ A, 2400bps에서 수정 발진기 153.6kHz를 사용하여 66~68 $\mu$ A의 측정 결과를 얻을 수 있었다. 본 칩의 특성은 다음과 같다.

- 동작 전압: 1.7V~5.0V(3.0V typ)
- 소비 전류: 100 $\mu$ A 최대(3.0V)
- 512, 1200, 2400 bps 지원
- 광역기능 지원
- 건전지 절약을 위한 3개 Power Save Mode 판



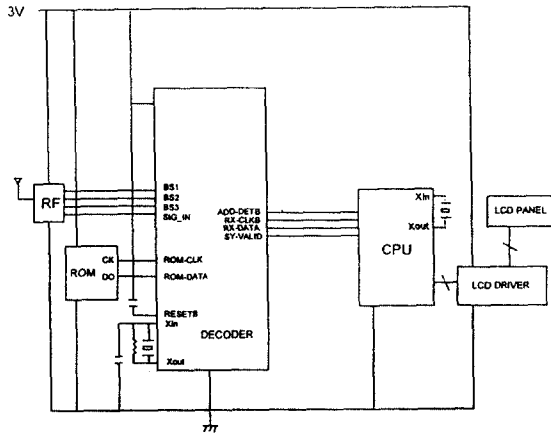


그림 13. 응용 보드 회로도  
Fig. 13 The Circuit of Application Board.

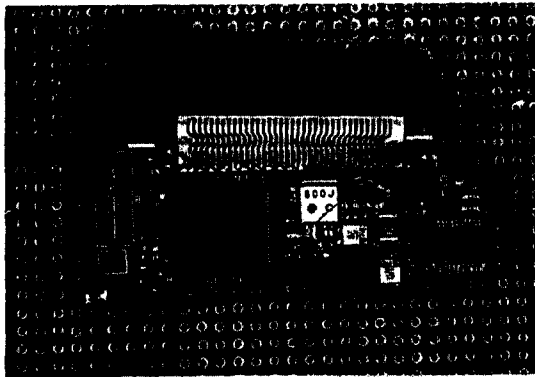
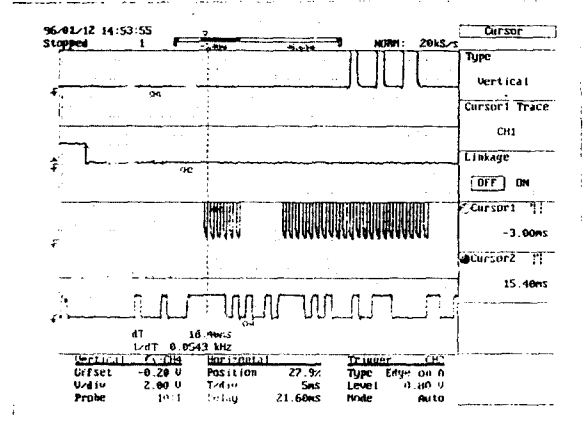


그림 14. 전체 Pager 보드 사진  
Fig. 14 The Photograph of the whole Pager Board.

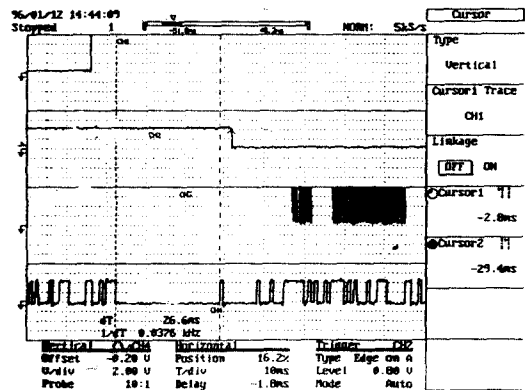
· 4 어드레스 사용자 정의 지원

VI. 결 론

Pager는 송신단에서 보낸 POCSAG 신호를 수신하여 자신의 ID인지 판단한 후, 수신된 값이 자신의 ID 코드이면 소유자에게 호출받은 사실이 있음을 알리고 메시지를 보여준다. 이때 Pager 내부의 Processor (MCU)는 수신된 값을 처리하기 위하여 항상 동작해



(a) 데이터 수신 모드



(b) 어드레스 수신 모드

그림 15. 측정된 Pager 디코더의 동작 특성 (a) 데이터 수신 모드 (b) 어드레스 수신 모드  
Fig. 15 The Operation Characteristics of the measured Pager Decoder. (a) Data Receive Mode (b) Address Receive Mode

야 하지만 건전지 소모량이 커지는 문제가 발생하기 때문에 RF 수신 회로와 MCU 사이에 항상 동작 중에 있으면서 RF 회로에서 출력된 값을 처리하고 호출된 경우에만 MCU를 작동시켜 필요한 처리를 하도록 하는 POCSAG Signal Decoder를 설계하였다.

Pager에서 중요한 건전지 수명을 길게 하기 위한, 즉 소비 전류의 최소화를 위해 에러정정에 사용한 BCH

정정 회로를 간단하게 구현하는 원리를 적용하여 게이트 수를 감소시켰고, 칩의 총전류 소모에서 가장 많은 부분을 차지하는 출력 버퍼 및 오실레이터의 셀을 기존 라이브러리를 사용하지 않고 재설계로 최적화하여 전류 소모를 적게 하도록 ASIC 칩을 구현하였다. 이렇게 구현된 칩은 실제 실험 결과 앞에서 제시된 모든 기능을 만족하였고, 소비 전류를 측정해 본 결과 512 bps에서는 25 $\mu$ A, 1200 bps에서는 32~33 $\mu$ A, 2400 bps에서는 66~68 $\mu$ A로 소비 전류가 측정되었다.

향후 반도체 공정이 저전압 소자를 위한 공정이 최적화 되어지고, 설계에 사용되는 라이브러리를 더욱 최소화하여 chip size를 작게 하면 더욱 소비 전류가 감소되는 효율적 디코더가 가능하리라 생각된다.

**참 고 문 헌**

1. Seiko Instrument, "PAGING DECODER IC(POCSAG) S-7040D Specification"
2. Motorola, "Motorola BRAVO Plus Series: Numeric Display Radio Pagers: Theory/Maintenance Manual. 68P81012B35-O"
3. Motorola, "Motorola Advisor Message Receivers: POCSAG and GSC: Alphanumeric Display Pagers: Theory/Maintenance Manual. 68P81011B75-O"
4. Motorola, "Motorola ADVISOR Message Receivers: GSC and POCSAG Alphanumeric Display. Service Manual for 138-174 MHz models"
5. Motorola, "Motorola BRAVO Plus Series: POCSAG Numeric Display Pagers. Service Manual for 929-932 MHz models"
6. Motorola, "Motorola BRAVO Plus Series: Numeric Display Pagers. Service Manual for 406-512 MHz models"
7. Neil J. Boucher, "The Paging Technology Handbook", Quantum Publishing, 1992.
8. Radio Design Group, "POCSAG Paging Encoder Product Specification JONE", 1995.
9. Rudolf F. Graf, "The modern oscillator circuit encyclopedia", Tab books, Blue Ridge Summit, 1992.
10. R. T. Chien, "Cyclic decoding procedures for

Bose-Chaudhuri-Hocquenghem codes," IEEE Trans. Inform. Theory, Vol. IT-10, pp. 357-363, Oct. 1964.

11. Rhee, Man Young, "Error-Correcting coding theory", McGRAW-HILL, 1989.
12. 강창언, "디지털 통신 입문", 북두출판사, pp. 201~204, 1996.
13. G. Forney, "On decoding BCH codes," IEEE Trans. Inform. Theory, Vol. IT-11, pp. 549-557, 1965.



**崔 鍾 文(Jong Moon Choi)**정회원  
 1981년 2월: 성균관대학교 전자공학과 졸업(공학사)  
 1983년 2월: 연세대학교 전자공학과 졸업(공학석사)  
 1983년 3월~현재: LG 반도체 ASIC 설계실 책임연구원

1992년 8월~현재: 연세대학교 전자공학과 박사과정  
 ※주관심분야: ASIC 설계, 이동통신, VLSI 설계



**金 永 大(Young Dae Kim)**정회원  
 1995년 2월: 숭실대학교 전자공학과 졸업(공학사)  
 1995년 3월~현재: LG 반도체 ASIC 설계실 연구원  
 ※주관심분야: ASIC 설계, 이동통신, VLSI 설계

**韓 廷 溢(Jung Il Han)** 정회원  
 1995년 2월: 연세대학교 전자공학과 졸업(공학사)  
 1995년 8월~현재: 연세대학교 전자공학과 석사과정  
 ※주관심분야: 에러정정, 이동통신, ASIC 설계