

# 레이더의 반 대응 능력을 위한 M-시퀀스 코드 기반의 펄스반복간격 지터 코더 구현

## Jittered Pulse Repetition Interval Coder Based on M-sequence Codes for Counter-Countermeasure of a Radar

표 순 오\*, 서 동 선\*\*, 조 준 용\*\*, 이 재 철\*\*  
Sun-Oh Pyo\*, Dong-Sun Seo\*\*, Jun-Yong Jo\*\*, Jae-Cheol Lee\*\*

### Abstract

In this paper, a novel pulse repetition interval (PRI) jittering coder based on quasi-random M-sequence codes is proposed for improvement of counter-countermeasure capability in a radar. Each of the proposed jittered 256 PRI codes has a unique code chip combination with 256 code chips, such that any set of three consequent code chips (4 pulses) from any code appears only once among the entire code chip sequences of the codes. This indicates that only 4 of received pulses are enough to determine uniquely the exact timing position of the incoming pulse train (or code chip sequence) required for counter-countermeasure, as well as the identity of the transmitted code. To prove the proposed idea experimentally, the jittered PRI coder is implemented and demonstrated.

### 요 약

본 논문에서는 레이더의 반대응 능력을 개선하기 위한 의사 랜덤 M 시퀀스 코드 기반의 새로운 펄스반복간격 (PRI) 지터 코더를 제안한다. 제안된 256개의 지터 PRI 코드 각각은 256 코드 칩의 유일한 조합으로 이루어져 있어서, 임의의 코드에서 선택된 어떠한 3개의 연속된 코드 칩(4개의 펄스)의 조합이라도 코드들의 모든 코드 칩 시퀀스들 중에서 오직 한 번만 존재한다. 이는 4개의 펄스만 수신하면 전송된 코드 식별은 물론이고 반대응을 위해 요구되는 수신 펄스열(또는 코드 시퀀스)의 정확한 타이밍을 결정할 수 있다는 것을 의미한다. 제안된 아이디어를 실험적으로 입증하기 위해, 상기 지터 PRI 코더를 구현하고 시연한다.

*Key words : pulse repetition interval modulation, counter-countermeasure, M-sequence code, pulse coder*

### 1. 서론

레이더(radar) 시스템에서 감시망 안에 들어온 표적의 위치 파악 및 추적을 위해 일반적으로 레이더

펄스를 반복적으로 사용한다. PRF(pulse repetition frequency) 코딩은 펄스의 반복률을 특정 주파수로 고정시킨 방법으로서, 출력의 안정성 및 디코더 구현 측면에서는 유리하나 정확한 측정이 가능한 최대 측정거리(unambiguous range)가 짧고 목표물로부터의 지능적인 진파 방해(jamming)에 취약하다. 그러므로 목표물이 여러 개이거나 정확한 측정이 어려운 영역이 존재하는 상황에서는 PRF 코딩 대신 더 정밀한 PRI(pulse repetition interval) 코딩이 사용된다[1].

지터(jittered) PRI 코딩은 펄스의 반복률(또는 펄스 간격)이 불규칙하게 변하는 코딩방식으로서 지능적인 진파 방해가 있는 환경에서 은밀하게 동작하는 레이

\* 明知大學校 電子工學科  
(Dept. of Electronics Engineering, Myongji University)  
\*\* 韓花綜合研究所  
(HanWha R&D Center)  
★ 교신저자 (Corresponding author)  
※ 본 연구는 한국연구재단(#2011-0002769)의 지원에 의해 수행되었음.

接受日: 2011年 6月 2日, 修正完了日: 2011年 6月 30日

더에게 중요하다[1,2]. 기존의 연구에서는 4개의 코드 칩을 이용하여 코드 주기가 4인 지터 PRI 코드 4개를 생성하는 방법이 연구되었으나[4], 코드 주기가 짧고 코드 수도 적어서 여전히 지능적인 전파 방해에 취약할 수 있다.

본 논문에서는 코드 주기 및 코드 수가 256개인 지터 PRI 코드를 생성하기 위해 의사 랜덤 M-시퀀스 코드 기반의 새로운 PRI 코딩 기법을 제안하고, 제안한 코딩 기법을 하드웨어적으로 구현할 수 있는 코드 생성기를 개발한다. 이 때 각 PRI 코드로 변조된 펄스는 표적이 대응(countermeasure)하기 어렵도록 불규칙적인 펄스열이어야 하고, 표적으로부터 대응하는 방해 펄스를 효과적으로 제거하기 위해 표적에서 반사된 펄스를 피도록 빠른 시간 내에 인지(acquisition) 및 추적(track)이 가능해야 한다. 이러한 지터 PRI 코드는 긴 주기를 지닌 유일한 코드 칩 조합으로 구성된 코드로서, 표적에게 불규칙적인 펄스열로 인식시키는 반면, 레이더 수신기는 코드의 일부만 수신해도 전송된 코드번호 및 코드열의 위치가 식별이 가능하여 추후 수신 예정 펄스의 시간위치 예측에 의한 반대응 능력을 높일 수 있다.

본 논문의 2장에서는 앞서 언급한 바와 같이 불규칙적인 펄스열 생성 및 빠른 디코딩이 가능하도록 지터 PRI 코드 규격(코드 길이, 코드 수, 하나의 코드를 구성하는 코드 칩 수, 디코딩 시 필요한 펄스 개수)을 도출하고, 3장에서는 이를 만족하는 새로운 PRI 코딩 기법을 제안한다. 4장에서는 제안한 기법을 이용하여 코드 생성기를 하드웨어로 개발하고, 개발한 코드 생성기의 성능을 실험적으로 입증하기 위해 코드 생성기를 내장한 지터 PRI 인코더/디코더 시험모듈을 개발하여 그 연동 시험 결과를 제시한다. 그리고 마지막 장에서는 결론을 제시한다. 일반적으로 레이더에서는 인코더/디코더가 하나의 모듈 내에 존재하기 때문에, 연동 시험 시에 PRI 디코더는 PRI 인코더에서 보낸 코드를 미리 알고 있다고 가정한다.

## II. 코드 규격

### 1. 요구 사항

PRI 코드는 미리 정해진 펄스 간격에 따라서 특정 코드 칩을 부여하고 이들을 조합하여 코드를 만드는 방식이다[4,5]. 반 대응 능력이 높은 PRI 코드를 생성하기 위해서는 다음의 특징을 만족해야 한다[1,4,5].

첫째, 펄스폭은 펄스 간격에 비해 가능한 짧게 한다. 이는 송신기의 전력 소비를 최소화할 수 있을 뿐만 아니라 거리 측정 분해능을 향상시킬 수 있다. 또한 이러한 PRI 코드는 시간 축에서 임펄스의 나열처

럼 보이므로, 스펙트럼 검출기를 통해 검출을 하면 평탄한 주파수 스펙트럼을 가진 잡음처럼 보인다. 그러므로 지능적인 전파 방해가 있는 환경에서 은밀하게 동작하는 것이 가능하다.

둘째, 하나의 코드는 제한된 코드 칩 규격 내에서 최대한 다양한 코드 칩으로 긴 주기를 갖도록 조합한다. 이는 코드 칩 조합이 불규칙하게 보이도록 함으로써 지능적인 전파 방해나 코드 정보 유출을 막을 수 있다.

셋째, 코드 수가 많아야 한다. 다양한 코드는 활용 폭을 넓히고, 지능적인 전파 방해를 막을 수 있다.

넷째, 수신기가 코드의 일부(즉 몇 개의 코드 칩)만 수신하여도 디코딩이 가능해야 한다. 지능적인 전파 방해를 막기 위해서 불규칙하고, 긴 주기를 갖도록 코드를 구성했어도 빠른 디코딩이 가능해야 한다. 여기서, 디코딩이 가능하다는 것은 코드 식별 및 추적이 가능함을 의미하는데, 특히 식별된 코드를 추적하기 위해서는 자기 상관 값이 시간 지연이 없는 경우에만 높게 나와야만 한다. 이는 측정 거리를 모호하게 만드는 반사파 신호(range-ambiguous echoes)의 영향을 줄여서 정확한 측정이 가능한 최대 표적 측정 거리를 늘릴 수 있다.

이상의 요구 조건을 만족하도록 인코더에서는 256개의 다양한 코드 칩을 조합하여 긴 주기(256개의 코드 칩)를 가지는 펄스(256 종류의 코드)를 발생시키고, 디코더에서는 4개의 수신 펄스(3개의 코드 칩)만으로도 수신된 코드가 정합한지를 인지할 수 있는 시스템을 구현한다.

### 2. 코드 검증을 위한 조건

앞서 기술한 코드 요구 사항을 만족시키는 코드는 코드 검증을 위해 다음의 기준을 만족해야 한다.

첫째, 하나의 코드는 코드 칩을 불규칙적으로 조합한 랜덤 시퀀스(random sequence)로 구성된다. 랜덤 시퀀스로 코드 칩 열을 구성하기 위해서 그림 1에서 보는 바와 같이 서로 다른 펄스 간격을 의미하는 각 코드 칩을 0~255 사이의 256개의 숫자로 표현한다. 이 때 코드 칩 간 펄스 간격 변화는 일정하다고 가정한다.

둘째, 하나의 코드를 구성할 때 다양한 코드 칩을 사용하기 위해서 256개의 코드 칩 중 최소 150개 이상을 사용한다. 또한 3개 이상 반복되는 코드 칩은 존재하지 않도록 코드를 설계한다.

셋째, 디코더에서 4개의 펄스(3개의 코드 칩)만을 수신하여도 수신 코드와 선택된 코드가 정합하는지를 인지할 수 있고, 정합되었다면 추적할 수 있도록 코드의 시작점(즉, 현재의 코드 칩 시퀀스의 위치)을 알

수 있도록 코드를 설계한다. 이는 임의의 지점으로부터 연속된 코드 칩 조합이 시퀀스 상에 오직 하나만 존재하도록 설계해야 한다. 즉, 코드 칩 블록 크기를 3으로 했을 때, 부분 교차 상관(PCC: partial cross-correlation)과 부분 자기 상관(PAC: partial auto-correlation)이 모두 0 이어야 한다.

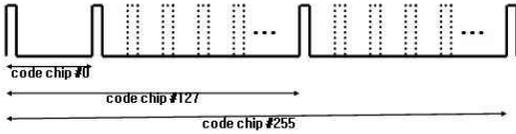


Fig. 1. 256 code chips with different pulse intervals  
그림 1. 펄스 간격이 다른 256개의 코드 칩

가. 부분 교차 상관 특성

교차 상관은 서로 다른 코드 간의 상관관계를 나타내기 때문에, 부분 교차 상관 특성은 디코더의 코드 식별을 위해 필요한 코드 칩의 수를 테스트한다. 시간 경과  $t$ , 주기  $T$ , 시간 지연  $k$ 인 두 코드  $C_1(t)$ ,  $C_2(t)$ 의 부분 교차 상관은 식 (1)과 같이 정의한다.

$$PCC = \sum_{k=1}^{k=T} \sum_{t=1}^{t=T} \sum_{p=0}^{p=P-1} (C_1(t+p) \oplus C_2(k+t+p)) \quad (1)$$

여기서  $p$ 는 칩 블록의 크기를 나타낸다. 코드 개수가 256개이기 때문에 전체 코드를 테스트하기 위한 교차 상관의 총 횟수는  $32640 (=255 \times (255+1)/2)$ 이 된다. 칩 블록의 크기를 3으로 했을 때 모두 0이 나와야만, 디코더가 4개의 수신 펄스(3개의 코드 칩)만으로도 코드를 식별할 수 있게 된다.

나. 부분 자기 상관 특성

디코더가 통상 비동기로 동작하므로, 코드 식별 능력뿐만 아니라 코드의 시작점(코드 시퀀스 상의 특정 위치) 인식 능력도 테스트해야 한다. 자기 상관은 자기 코드 간의 상관 관계를 나타내므로, 부분 자기 상관은 코드의 시작점 인식을 위해 필요한 코드 칩의 수를 테스트한다. 시간 경과  $t$ , 주기  $T$ , 시간 지연  $k$ 인 코드  $C(t)$ 의 부분 자기 상관은 식 (2)와 같이 정의한다.

$$PAC = \sum_{k=1}^{k=T-1} \sum_{t=1}^{t=T} \sum_{p=0}^{p=P-1} (C(t+p) \oplus C(k+t+p)) \quad (2)$$

코드 개수가 256개이기 때문에 자기 상관의 총 횟수는 256번으로, 칩 블록의 크기를 3으로 놓았을 때 모두 0이 나와야만, 4개의 수신 펄스(3개 코드 칩)만

으로도 코드의 시작점을 알 수 있다.

### III. PRI 코딩 기법

#### 1. M(Maximal-length) 시퀀스[6]

PN(Pseudo Noise) 시퀀스는 랜덤하게 결정된 칩의 나열을 주기적으로 반복하는 신호로서 샘플된 백색 잡음의 통계적 성질을 지니고 있다. 또한 피드백 탭(feedback tap)의 조합을 바꿈으로써 직교 특성을 가진 여러 개의 코드를 생성할 수 있다. PN 시퀀스는 위와 같이 랜덤 특성, 우수한 상관 함수 특성을 가지고 있을 뿐만 아니라 아주 쉬운 방법으로 발생시킬 수 있는 장점이 있다. 본 논문에서는 PN 시퀀스 중 가장 대표적인 시퀀스인 LFSR(Linear Feedback shift-register) 시퀀스 중 M 시퀀스를 PRI 코드에 적용할 알고리즘으로 사용한다.

피드백 시프트 레지스터 시퀀스(feedback shift-register sequence)는 PN(Pseudo Noise) 시퀀스의 일종으로 DSSS(Direct Sequence Spread Spectrum) 시스템에서 이용하고 있다. 이 시퀀스는 피드백 시프트 레지스터들의 출력을 조합함으로써 생성되어진 주기적인 이진 시퀀스이다. 피드백 로직(feedback logic)이 그림 2에 보인 바와 같이 전부 modulo-2 adder(exclusive OR gate)로 구성되어지면, 피드백 시프트 레지스터와 생성된 출력 시퀀스를 선형이라고 부른다. 그림 2-(a)는 LFSR의 표준 형태이고, 그림 2-(b)는 고속 스피드 형태이다. 각 레지스터의 초기 값과 피드백 로직은 다음 클럭의 레지스터 값과 출력을 결정하므로 중요한 요소이다.

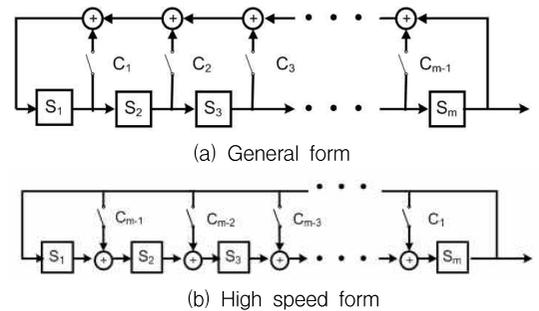


Fig. 2. PN sequence generation based on LFSR form.  
그림 2. LFSR 형태 기반의 PN-시퀀스 발생

그림 2의 다이어그램을 수식으로 표현하면 식 (3)과 같다.

$$\begin{aligned} s_j(i) &= s_{j-1}(i-1) \oplus c_{m-j+1} s_m(i-1), \quad i \geq 1, 2 \leq j \leq m \\ s_1(i) &= s_m(i-1), \quad i \geq 1 \end{aligned} \quad (3)$$

위 수식에서  $S_j$ 는 시프트 레지스터의 상태,  $C_m$ 은 피드백 탭의 상태를 나타낸다. 또한  $j$ 는 시프트 레지스터의 위치이고  $i$ 는 경과 시간을 의미한다.

LFSR의 상태가 모두 0이면, 그것의 값들은 언제나 0 이므로 출력으로 주기 함수가 만들어지지 않는다. 그러므로  $m$ 개의 LFSR은  $2^m-1$  개보다 작은 상태를 가지기 때문에 출력 시퀀스의 주기는  $2^m-1$ 를 초과할 수 없다. 출력 시퀀스의 주기가  $2^m-1$ 이면 M-시퀀스라고 부르고, 이 때 M-시퀀스의 주기는 초기 값에 영향을 받지 않는다. M-시퀀스의 초기 값은 출력을 천이 시키는 역할을 하며 이를 M-시퀀스의 천이 특성이라고 부른다.

**2. M-시퀀스 코드 기반의 PRI 코드 생성**

본 연구에서는 COOLEY 전자 연구실에서 개발한 M-시퀀스 생성기를 만들기 위한 primitive 다항식을 이용한다[7].

칩 당 8비트 레벨의 256(코드 길이, 하나의 코드를 구성하는 코드 칩 수) × 256(코드 수) PRI 코드 테이블을 생성하기 위해서는  $2^{3+8+8}=2^{19}$  주기를 갖는 M-시퀀스가 필요하다. LFSR이 primitive 다항식으로 생성된다면  $m$ 개의 시프트 레지스터를 갖는 LFSR의 출력으로 주기  $2^m-1$ 인 M-시퀀스를 생성시키므로, 본 연구에서 사용하는 M-시퀀스는 19차 primitive 다항식으로 생성한 M-시퀀스이다. 그림 3에 보인 바와 같이 19차수에도 다항식의 0이 아닌 계수의 개수에 따라 많은 수의 primitive 다항식이 존재하는 것을 알 수 있다. 다항식의 0이 아닌 계수의 개수는 LFSR의 피드백 탭의 수와 일치한다.

Equation Degree																				
Nonzero Coefficient	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
3	1	5	2	2	2	4	0	2	2	5	0	0	0	0	6	2	2	0	0	
5							4	4	10	12	16	20	44	19	66	42	82	82	152	72
7								4	4	12	28	39	90	96	236	238	470	360	1050	718
9									2	10	50	96	264	339	720	812	2674	2396	6996	6996
11										2	10	50	96	264	339	720	812	2674	2396	6996
13											4	10	140	450	640	2696	2910	10238	1774	1774
15												4	10	140	450	640	2696	2910	10238	1774
17														6	156	1008	1470	6766	6766	1772
19																8	122	284	1772	1772
20																	4	24	180	180
Total	1	2	2	6	6	18	16	48	60	176	144	630	756	1980	2048	7710	7776	27984	27984	

Fig. 3. The number of primitive polynomials for M sequence generator.

그림 3. M-시퀀스 생성기를 위한 Primitive 다항식 개수 이 primitive 다항식 중 '19차 5 nonzero coefficient primitive polynomial 2000047'을 사용하여 본 연구의 규격에 맞는 PRI 코드를 생성하는 방법은 다음과 같다.

primitive 다항식은 increasing order인 8진수로 표현되어 있다. 그러므로  $2000047_8 = 010\ 000\ 000\ 000$

$000\ 100\ 111_2$  이고 이를 식 4와 같이 표현할 수 있다.

$$P(x) = x^{19} + x^5 + x^2 + x + 1 \tag{4}$$

그림 4는 이 다항식을 LFSR의 고속 스피드 형태로 나타낸 것이다. 코드 주기는 M-시퀀스 주기 + 1 즉  $2^{8+3+8} = 2^{19} = 524288$ 로 하고, 초기 값은 레지스터 19 만 1이고 나머지 레지스터는 0으로 한다.

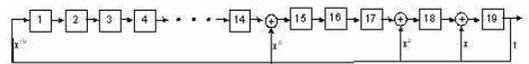


Fig. 4. LFSR of ' $P(x) = x^{19} + x^5 + x^2 + x + 1$ '

그림 4. ' $P(x) = x^{19} + x^5 + x^2 + x + 1$ '의 LFSR

그림 5는  $2^{19}$  길이의 이진 M-시퀀스를 PRI 코드 테이블로 바꾸어주는 방법을 나타내고 있다. 먼저 코드 수를  $2^8=256$ 으로 만들기 위해 M-시퀀스를 가로로 256등분한다. 그 후에 이진수 8개를 십진수로 변환한다. 이렇게 변환한 0~255 사이의 특정 숫자는 코드를 구성하는 각 코드 칩이 된다. 이와 같은 과정을 반복하면 256(코드 길이 즉 하나의 코드에 사용되는 코드 칩 수) × 256(코드 수)의 PRI 코드 테이블을 구성할 수 있다.

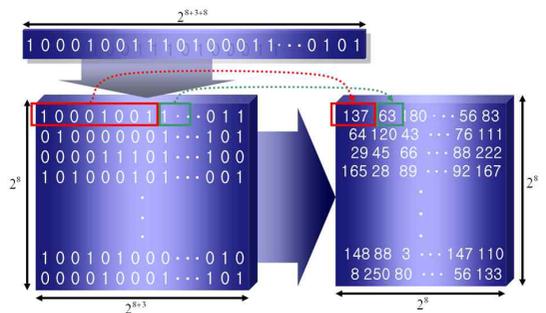


Fig. 5. Schematic method for PRI code table generation.

그림 5. PRI 코드 테이블 생성 개략 방법

**3. 생성된 PRI 코드 검증**

MATLAB을 이용하여, 생성된 PRI 코드 테이블의 코드 검증 결과를 제시한다. 본 시스템의 요구 사항을 만족하는 PRI 코드는 코드 규격뿐만 아니라 제시된 코드 검증 기준을 모두 만족해야 한다.

검증 결과 생성된 각 PRI 코드들은 최소 152개 및

최대 176개의 코드 칩을 사용했고, 평균 162개의 코드 칩을 사용하였다. 또한 3개 이상 반복되는 코드 칩은 존재하지 않았다. 그리고 설계한 PRI 코드들의 부분 교차 상관(코드의 식별 능력)과 부분 자기 상관(코드의 시작점 인식 능력)이 칩 블록 크기가 3일 때 모두 0이 나왔다. 이는 디코더에서 4개의 펄스(3개 코드 칩)만 받아 들여도 코드 식별 및 코드의 시작점을 알 수 있는 코드라는 것을 의미한다.

위 방식의 가장 큰 장점은 본 논문에서 제시한 요구 사항을 가진 PRI 코드들을 여러 가지 피드백 탭으로 생성 가능하다는 점이다. 비록 19차 primitive polynomial들이 모두 요구 조건을 만족하는 PRI 코드를 생성하지는 않지만, 실험적으로 확인해 본 결과 '19차 7 nonzero coefficient primitive polynomial 2101231'도 본 연구의 요구 조건을 만족하는 PRI 코드를 생성한다. 이는 PRI 코딩 알고리즘이 누출되어도 피드백 탭을 알지 못하면 사용하는 코드를 알 수 없으므로 보안성이 유지됨을 의미한다.

#### IV. PRI 코드 생성기 개발 및 검증

##### 1. PRI 코드 생성기 개발

그림 6은 PRI 코드 생성기의 기능 블록 다이어그램이다. 이 그림에서 M-시퀀스 생성기는 본 논문에서 설명한 '19차 5 nonzero coefficient Primitive polynomial 2000047'을 사용하여 생성한 LFSR이다.

코드 생성기가 동작 상태가 되면, M-시퀀스 생성기에서 매 클럭 당 M-시퀀스를 형성시키는 비트가 한 개씩 생성되고 3비트 카운터(counter)에서 이를 카운트(count)한다. 3비트 카운터이므로 M-시퀀스 8개가 발생할 때마다 오버플로우1(OVERFLOW1)이 발생하고 첫 번째 8비트 카운터에서 이를 카운트한다. 첫 번째 8비트 카운터가 카운트하기 전에 코드 테이블에 8비트 큐(queue)에 들어 있는 M-시퀀스 8개 즉, 코드 칩을 저장한다. 이 때 메모리 주소는 코드 칩 카운트가 된다. 첫 번째 카운터는 8비트 카운터이므로 코드 칩 카운트가 256이 되면 오버플로우2(OVERFLOW2)가 발생하고 두 번째 8비트 카운터가 이를 카운트 한다. 비교기에서 코드 카운트가 스위치로 입력된 코드보다 크면 코드 테이블이 완성되었다는 신호를 보내고 M-시퀀스 생성기는 동작을 멈춘다. 시험 모듈은 메모리의 낭비를 막기 위해서 8비트 메모리인 코드 칩이 256개밖에 존재하지 않는다. 이는 매 코드 카운트마다 코드 칩을 갱신하여 MCU에서 필요한 코드 정보를 인코더나 디코더는 해당 코드의 코드 칩들만 갖출 수 있도록 개발한다.

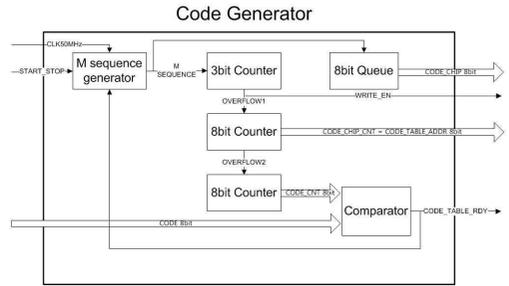


Fig. 6. Functional block diagram of the code generator.  
그림 6. 코드 생성기의 기능 블록 다이어그램

##### 2. PRI 인코더/디코더 시험모듈 개발

그림 7은 PRI 인코더와 인코더 제어기의 기능 블록 다이어그램이다. 시스템이 동작 하고 있지 않을 경우 START/STOP 스위치를 누르면, MCU는 FPGA의 특정 메모리로 코드를 전송하고, FPGA는 코드 생성기를 통해 코드 칩 열을 생성한다. 이후 FPGA 클럭을 이용하여 순차적으로 코드 칩에 해당하는 펄스 간격을 갖는 펄스를 발생시킨다.

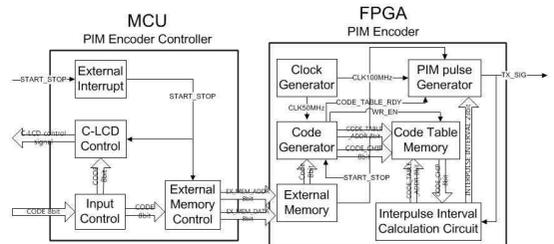


Fig. 7. Functional block diagram of the PRI encoder.  
그림 7. PRI 인코더 기능 블록 다이어그램

그림 8은 PRI 디코더와 디코더 제어기의 기능 블록 다이어그램이다. PRI 디코더는 PRI 인코더와 같은 코드 생성기가 존재하여 수신 받고자 하는 코드를 선택할 수 있고, PRI 디코더는 디지털 코드 상관기도 가지고 있어서 수신 펄스와 미리 선택된 코드로 변조된 PRI 펄스가 정합한지를 인지하는 능력을 갖는다. 초기 상태는 다음에 오는 PRI 펄스를 예상할 수 없는 Search 상태이고, 다음에 오는 PRI 펄스의 위치(PRI 펄스 간격)를 예상할 수 있다면 Track 상태가 된다. PRI 디코더는 수신된 PRI 펄스와 선택 코드로 변조된 펄스가 정합된다고 판단되면 Search 상태에서 Track 상태로 전환된다. 수신 포트 정보와 디코더를 통해 알아낸 정합 상태 여부 및 측정된 펄스 간격은 캐릭터 LCD와 RS232 통신을 통해 PC의 하이퍼터미널(hyperterminal)에 표시된다.

Search 회로는 연속되는 펄스 4개를 수신한 후 3개의 펄스 간격을 계산하여 코드 테이블의 연속되는 코드 칩 3개의 펄스 간격과 비교하는 회로로 디지털 코드 상관기[8]를 이용하여 설계한다. 만일 비교했을 때 같은 것이 있다면 설계한 알고리즘 특성으로 인해 이후에 들어오는 펄스의 간격은 코드 테이블의 코드 칩으로 계산된 펄스 간격과 연속적으로 일치하게 된다. 그러므로 비교했을 때 같은 것이 있다면 Track 회로로 전환한다.

Track 회로는 매 입력되는 펄스 간격과 예상되는 펄스 간격을 비교하는 회로이다. 인코더에서 코드를 바꾸지 않거나 외부의 어떠한 방해 요인이 없다면 계속적으로 같은 것으로 예상되지만, 그렇지 않을 경우 Search 회로로 전환한다.

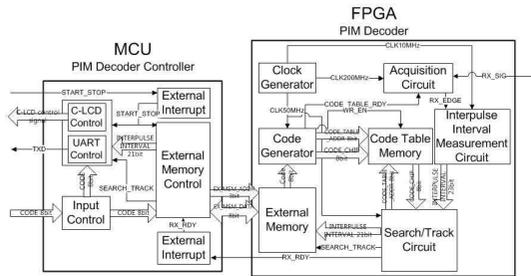


Fig. 8. Functional block diagram of the PRI decoder.  
그림 8. PRI 디코더 기능 블록 다이어그램

3. PRI 인코더/디코더 시험모듈 연동 시험

시험 절차는 다음과 같다. PRI 인코더에서 발생시킨 PRI 펄스를 PRI 디코더에서 받는다. 디코더는 현재 상태를 판단한다. 현재 상태가 Search 상태이면 0으로, Track 상태이면 1로 표현한다. 현재 상태가 판단되면 MCU는 PRI 디코더로부터 측정된 펄스 간격과 현재 상태를 읽어 들이고 측정된 펄스 간격은 해당하는 8비트 코드 칩으로 변환한다. 이후 코드 칩과 현재 상태를 16진수로 바꾼 다음, RS232 통신(시리얼 통신)을 이용하여 PC로 전송한다. 전송된 결과는 하이퍼터미널에 디스플레이 한다.

그림 9는 PRI 인코더/디코더 연동시험 결과이다. 디코더에서 펄스 간격을 측정할 때마다 한 쌍의 데이터(측정 펄스 간격과 현재 상태)를 PC의 하이퍼터미널에서 디스플레이 하였다. 디스플레이 된 펄스 간격은 인코더에서 PRI 변조된 것으로 MATLAB으로 코드 테이블을 생성했을 때 해당하는 코드의 코드 칩이 의미하는 펄스 간격과 일치했다. 상태는 인코더와 디코더가 같은 코드로 설정했을 경우에는 3개의 코드 칩을 수신하였을 때 0(Search)에서 1(Track)으로 바뀌는 것을 확인할 수 있었고, 서로 다른 코드일 경우에는 항상 0(Search)이 나왔다.

시험 모듈 테스트 결과 인코더에서는 입력 코드에 따른 긴 주기를 가진 불규칙적인 PRI 펄스가 발생하였다. 또한 디코더에서는 4개의 PRI 펄스(3개의 코드 칩)만 수신하여도 수신 펄스와 디코더에서 선택한 코드로 변조된 펄스가 정합한지를 인지할 수 있었고, 정합의 경우 코드 추적이 가능했다. 이는 디코더가 인코더에서 보낸 코드를 미리 알고 있다고 가정하면, 방해 전파 신호와 목표물로부터 반사된 전파 신호를 구분할 수 있다는 것을 입증한다.

```

80 0 02 0 70 1 08 1 2A 1 27 1 B2 1 40 1
0A 1 F8 1 2D 1 7D 1 AD 1 B8 1 BE 1 E7 1
9C 1 88 1 A8 1 D5 1 C8 1 12 1 0F 1 47 1
9C 1 85 1 2F 1 37 1 15 1 31 1 4E 1 7F 1
49 1 10 1 3B 1 AE 1 FA 1 F2 1 F7 1 50 1
45 1 D7 1 2C 1 78 1 09 1 4D 1 22 1 B8 1
76 1 90 1 80 1 90 1 72 1 3F 1 E9 1 28 1
79 1 9A 1 CB 1 F3 1 A1 1 B5 1 CA 1 56 1
07 1 AC 1 5C 1 79 1 C3 1 4A 1 A4 1 34 1
3F 1 41 1 6A 1 9D 1 68 1 CF 1 80 1 72 1
75 1 E2 1 2E 1 BD 1 2D 1 D2 1 CF 1 6C 1
80 1 D2 1 22 1 DB 1 56 1 ED 1 4F 1 A5 1
63 1 5A 1 C8 1 7B 1 AE 1 C2 1 F2 1 02 1
53 1 C8 1 B8 1 4D 1 84 1 0A 1 03 1 AE 1
24 1 F6 1 0C 1 03 1 74 1 51 1 8F 1 F7 1
53 1 A5 1 D9 1 D8 1 46 1 C 1 02 1 11 1 08 1
98 1 C5 1 1A 1 5F 1 01 1 8F 1 A6 1 AB 1
54 1 87 1 47 1 EF 1 85 1 4F 1 C6 1 52 1
92 1 EE 1 54 1 C8 1 2E 1 EF 1 85 1 AB 1
C5 1 B6 1 FD 1 DB 1 E9 1 47 1 D6 1 E2 1
BF 1 9F 1 59 1 A6 1 F7 1 15 1 C4 1 FA 1
38 1 34 1 35 1 C1 1 46 1 2D 1 C2 1 EF 1
22 1 2D 1 35 1 C1 1 46 1 2D 1 C2 1 EF 1
1E 1 C7 1 B0 1 14 1 0C 1 1F 1 78 1 12 1
35 1 63 1 44 1 C8 1 08 1 1F 1 24 1 ED 1
2C 1 64 1 59 1 1E 1 12 1 52 1 F3 1 E4 1
15 1 F1 1 DA 1 FD 1 C8 1 69 1 03 1 A7 1
EF 1 D1 1 4E 1 A3 1 E7 1 A2 1 B8 1 46 1
80 1 02 1 70 1 08 1 2A 1 27 1 B2 1 40 1
52 0 CB 0 DC 0 A1 0 10 0 58 0 AF 0 81 0
D7 0 D7 0 FB 0 BB 0 12 0 A8 0 65 0 0A 0
B3 0 4C 0 68 0 2B 0 19 0 3C 0 0E 0 00 0
3D 0 42 0 58 0 9F 0 45 0 A4 0 1A 0 F7 0 05 0
A9 0 F1 0 8C 0 70 0 01 0 5A 0 85 0 B9 0
71 0 80 0 0D 0 34 0 30 0 A1 0 50 0 E9 0
95 0 F5 0 CA 0 E8 0 05 0 29 0 57 0 DE 0
88 0 59 0 18 0 3D 0 89 0 E0 0 30 0 24 0
D2 0 C2 0 8C 0 83 0 86 0 3F 0 49 0 E2 0
B8 0 3F 0 45 0 EA 0 8C 0 98 0 86 0 9B 0
48 0 55 0 A2 0 63 0 84 0 6B 0 42 0 0F 0
11 0 F9 0 E9 0 5A 0 16 0 B8 0 96 0 C7 0
27 0 16 0 22 0 F4 0 D6 0 4A 0 9D 0 54 0
CF 0 06 0 F1 0 B9 0 DC 0 62 0 D3 0 2B 0
CB 0 92 0 20 0 59 0 5D 0 54 0 A7 0 07 0
72 0 BF 0 41 0 5A 0 95 0 BA 0 CC 0 FF 0
BF 0 6F 0 3A 0 3D 0 5C 0 23 0 22 0 11 0
85 0 88 0 57 0 36 0 68 0 69 0 C2 0 84 0
A3 0 A1 0 63 0 0C 0 90 0 4A 0 47 0 B5 0
47 0 80 0 44 0 03 0 2B 0 08 0 94 0 62 0
4C 0 2C 0 CC 0 2B 0 FC 0 35 0 8E 0 47 0
2C 0 66 0 09 0 38 0 63 0 08 0 B9 0 18 0
87 0 91 0 PR 0 99 0 45 0 5C 0 CB 0 21 0
F5 0 DB 0 62 0 4D 0 B9 0 4A 0 E4 0 15 0
07 0 19 0 7D 0 84 0 38 0 03 0 75 0 0C 0
DD 0 57 0 E7 0 0A 0 1C 0 92 0 B6 0 CF 0
CD 0 AE 0 9A 0 B3 0 52 0 6E 0 SE 0 E8 0
CD 0 72 0 F3 0 AD 0 55 0 CD 0 62 0 19 0
06 0 11 0 28 0 8A 0 49 0 1F 0 FE 0 54 0
4A 0 85 0 00 0 57 0 D2 0 53 0 SE 0 73 0
04 0 86 0 00 0 2A 0 40 0 B3 0 FA 0 A4 0
96 0 1A 0 A4 0 22 0 3F 0 15 0 2B 0 F9 0
52 0 CB 0 DC 0 A1 0 10 0 58 0 AF 0 81 0
    
```

(a) Coder : code 0 Decoder: code 0 (b) Coder : code 3 Decoder: code 0

```

52 0 CB 0 DC 1 A1 1 10 1 68 1 AF 1 81 1
D7 1 D7 1 FB 1 BB 1 12 1 A8 1 65 1 0A 1
9C 1 88 1 A8 1 D5 1 C8 1 12 1 0F 1 47 1
BD 1 42 1 93 1 53 1 F2 1 59 1 B2 1 93 1
A9 1 F1 1 8C 1 7D 1 01 1 5A 1 15 1 B9 1
71 1 80 1 0D 1 34 1 30 1 A1 1 50 1 B9 1
76 1 95 1 CA 1 1E 1 05 1 29 1 57 1 DB 1
98 1 99 1 18 1 3D 1 89 1 10 1 30 1 34 1
D2 1 C2 1 8C 1 83 1 86 1 FF 1 49 1 B2 1
B8 1 3F 1 45 1 EA 1 8C 1 98 1 96 1 9B 1
62 1 16 1 62 1 63 1 16 1 16 1 96 1 C7 1
11 1 F9 1 89 1 5A 1 16 1 B8 1 96 1 C7 1
27 1 16 1 62 1 F4 1 D6 1 4A 1 19 1 54 1
CF 1 06 1 F1 1 89 1 E2 1 52 1 F3 1 E4 1
C8 1 82 1 20 1 59 1 15 1 54 1 A7 1 07 1
72 1 BF 1 41 1 5A 1 9D 1 BA 1 CC 1 8F 1
BF 1 6F 1 2A 1 3D 1 5C 1 23 1 22 1 11 1
B5 1 98 1 57 1 56 1 60 1 69 1 C2 1 84 1
A3 1 A1 1 63 1 C9 1 4A 1 14 1 17 1 D5 1
47 1 B0 1 44 1 03 1 2B 1 8D 1 94 1 62 1
4C 1 2C 1 CC 1 2B 1 F7 1 35 1 98 1 47 1
2C 1 66 1 09 1 3E 1 13 1 0E 1 E9 1 1E 1
DD 1 91 1 FE 1 99 1 45 1 5C 1 EC 1 21 1
F8 1 DB 1 6E 1 4D 1 B8 1 4A 1 84 1 15 1
07 1 D9 1 7D 1 84 1 3B 1 03 1 75 1 0C 1
FD 1 57 1 B7 1 0A 1 1C 1 8E 1 B8 1 CF 1
06 1 72 1 F3 1 A0 1 55 1 CD 1 62 1 19 1
CD 1 A8 1 9A 1 E3 1 52 1 6E 1 5E 1 58 1
0A 1 11 1 2E 1 8A 1 43 1 1F 1 15 1 54 1
2C 1 05 1 28 1 19 1 97 1 23 1 8E 1 73 1
46 1 86 1 00 1 2A 1 40 1 B9 1 FA 1 AC 1
96 1 1A 1 84 1 22 1 F7 1 15 1 29 1 F9 1
52 1 CB 1 DC 1 A1 1 10 1 68 1 AF 1 81 1
80 0 02 0 70 0 08 0 2A 0 27 0 E2 0 40 0
D7 0 D7 0 FB 0 BB 0 12 0 A8 0 65 0 0A 0
9C 0 88 0 A8 0 D5 0 C8 0 12 0 0F 0 47 0
BD 0 42 0 93 0 53 0 F2 0 59 0 B2 0 93 0
A9 0 10 0 3B 0 AE 0 FA 0 F2 0 F7 0 50 0
45 0 D7 0 2C 0 78 0 09 0 4D 0 22 0 B8 0
76 0 90 0 80 0 90 0 72 0 3F 0 E9 0 28 0
79 0 9A 0 CB 0 F3 0 A1 0 B5 0 CA 0 56 0
07 0 AC 0 5C 0 79 0 C3 0 4A 0 A4 0 34 0
3F 0 41 0 6A 0 90 0 68 0 CF 0 80 0 72 0
76 0 82 0 0C 0 2B 0 ED 0 47 0 05 0
80 0 32 0 22 0 DB 0 86 0 D0 0 4F 0 A5 0
63 0 5A 0 C8 0 7B 0 A8 0 C2 0 F2 0 02 0
53 0 CB 0 E8 0 1F 0 45 0 A4 0 1A 0 F7 0 05 0
CF 0 F2 0 0C 0 33 0 18 0 1F 0 44 0 5E 0
8A 0 EE 0 1D 0 29 0 3E 0 DF 0 0C 0 7F 0
95 0 51 0 88 0 10 0 87 0 4C 0 B9 0 B8 0
10 0 B4 0 0C 0 A6 0 D6 0 75 0 DB 0 BC 0
24 0 F6 0 0C 0 00 0 74 0 81 0 FB 0 F0 0
53 0 A5 0 D9 0 D8 0 46 0 0C 0 21 0 09 0
98 0 C5 0 1A 0 59 0 01 0 8F 0 A5 0 A3 0
54 0 87 0 47 0 8F 0 85 0 4F 0 C6 0 62 0
F3 0 EE 0 54 0 8C 0 26 0 0F 0 85 0 AF 0
C5 0 B6 0 FD 0 DB 0 89 0 4F 0 D6 0 E2 0
8F 0 9F 0 59 0 A6 0 87 0 15 0 C4 0 FA 0
38 0 34 0 35 0 C1 0 46 0 2D 0 C2 0 0F 0
22 0 2D 0 35 0 6C 0 A4 0 F2 0 7C 0 1D 0
18 0 C7 0 B0 0 14 0 02 0 5D 0 88 0 94 0
35 0 63 0 44 0 C8 0 08 0 0F 0 24 0 02 0
12 0 64 0 C9 0 34 0 BB 0 22 0 78 0 A4 0
2C 0 F1 0 DA 0 FD 0 CB 0 63 0 09 0 87 0
8F 0 D1 0 4E 0 A3 0 E7 0 A2 0 B8 0 46 0
80 0 02 0 70 0 08 0 2A 0 27 0 E2 0 40 0
    
```

(c) Coder : code 3 Decoder: code 3 (d) Coder : code 0 Decoder: code 3

Fig. 9. Experimental results for performance verification of the PRI encoder/decoder.

그림 9. PRI 인코더/디코더 시험 검증 결과

V. 결론

본 논문은 펄스 레이더 시스템에서 반 대응 능력을 향상시키기 위해서, 의사 랜덤 M-시퀀스 코드 기반의 새로운 PRI 코딩 기법을 제안하였다. 제안한 PRI 코딩 기법은 M-시퀀스를 적용하여 상대적으로 긴 주

기(하나의 코드에 256개의 코드 칩 조합)를 갖고 불규칙하게 구성된 다수의 코드(코드 256개)를 생성할 수 있었다. 하드웨어적 제약으로 코드 칩 개수는 256개로 제한을 두었고, 하나의 코드는 최소 150개 이상의 코드 칩 조합으로 구성하였다. 또한 3개 이상 반복되는 코드 칩은 존재하지 않았다. 그리고 설계한 PRI 코드들의 부분 교차 상관(코드의 식별 능력)과 부분 자기 상관(코드의 시작점 인식 능력)이 칩 블록 크기가 3일 때 모두 0이 나왔다. 이는 디코더에서 4개의 펄스(3개 코드 칩)만 받아 들여도 코드 식별 및 코드의 시작점을 알 수 있는 코드라는 것을 의미한다.

제한한 코딩 기법을 이용하여 코드 생성기를 하드웨어로 개발하였고, 개발한 코드 생성기의 성능을 실험적으로 입증하기 위해 코드 생성기를 내장한 PRI 인코더/디코더 시험모듈을 개발하였다. 시험 모듈을 테스트한 결과 인코더에서는 입력 코드에 따른 긴 주기를 가진 불규칙적인 PRI 펄스 시퀀스를 발생하였고, 발생된 시퀀스의 펄스들 간의 간격은 MATLAB으로 생성된 코드 테이블 상의 해당 코드의 코드 칩 시퀀스가 의미하는 펄스들 간의 간격과 정확히 일치함을 확인하였다. 디코더에서는 4개의 PRI 펄스(3개의 코드 칩)만 수신하여도 수신 펄스와 디코더에서 선택한 코드로 변조된 펄스가 정합하는지를 인지할 수 있었고, 수신될 펄스 시퀀스의 시간적인 배열 위치 예측에 의한 정합 코드만의 추적이 가능하였다. 정합 코드의 추적 능력은 여타의 대응 또는 잡음 펄스를 효과적으로 제거할 수 있는 반대응 능력을 의미한다.

이 같은 PRI 인코딩/디코딩 시험모듈 검증 결과는 인코딩된 신호를 RF 주파수 신호로 반송파 변조하여 목표물에 방출하고, 반사된 전파를 안테나를 통해서 수신 후 반송파 복조한 다음, PRI 디코딩하는 실제 레이더 시스템에서도 그대로 적용될 것으로 예측된다. 또한, 지터 PRI 레이더뿐만 아니라 지능적인 전파 방해가 있는 환경에서 운용되는 다른 응용 분야에서 유용한 반대응 기법으로 활용될 것으로 판단된다.

### 참고문헌

[1] J. W. McCorkle, "Random number generator for jittered pulse repetition interval radar systems," U.S. Patent 5,847,677, Dec. 8, 1998.  
[2] G. E. Matich, David Henry Ramsey, Raymond John Wall, "Ranging systems," U.S. Patent 7,259,713, Aug. 21, 2007.  
[3] 류영진, 김환우, "자기상관관계를 이용한 레이더 신호의 PRI 변조형태 인식 기법," 전자공학회 논문지 vol. 43 No. 3, pp. 61-67, 2006. 5.

[4] Thomas E. Bayston, "Pulse code recognition method and system," U.S. Patent 5,023,888, Jun. 11, 1991.  
[5] K. Miwa, "Pulse modulation method," U.S. Patent 5,640,160, Jun. 17, 1997.  
[6] D. J. Torrieri, "Principles of Spread-Spectrum Communication Systems," pp. 55-128, Springer Publishing, 2004.  
[7] K. Metzger, J. R. J. Bouwens, "An ordered table of primitive polynomials over GF(2) of degrees 2 through 19 for use with linear maximal sequence generators," Tech. Memo. No. 107 004860-2-M, Cooley Electron. Lab., Dept. of Electrical and Computer Engr., U. of Michigan, Jul. 1972.  
[8] J. P. A. Moreau, "Digit Sequence Correlator," U.S. Patent 3,598,979, Aug. 10, 1971.

### 저 자 소 개

#### 표 순 오 (학생회원)



2009년 : 명지대학교 전자공학과 졸업 (공학사)  
2009년 ~ 현재 : 명지대학교 대학원 전자공학과 (석사과정)  
<주관심분야> 센서 신호처리  
회로설계, 디지털 모뎀 설계,  
디지털 필터 설계

#### 서 동 선 (중신회원)



1980년 : 연세대학교 전자공학과 졸업 (공학사)  
1985년 : 연세대학교 대학원 전자공학과 (공학석사)  
1989년 : 뉴멕시코 주립대학교 졸업 (공학박사)  
1990년 2월~현재 : 명지대학교 교수 재직

<주관심분야> ultra short pulse technology, optical CDMA, microwave photonics, optical communications, optical measurements

**조 준 용** (비회원)

1998년 : 성균관대학교 전자공학과 졸업 (공학사)

2000년 : 성균관대학교 대학원 전기전자컴퓨터공학과 (공학석사)

2005년 : 성균관대학교 대학원 전기전자컴퓨터공학과 (공학박사)

2003년~2007년 : 고등기술연구원 포토닉스센터 근무

2008년~현재 : (주)한화 종합연구소 재직(책임연구원)  
<주관심분야> 광학센서, 레이저응용분야, 광통신소자, 광섬유소자

**이 재 철** (비회원)

1980년 : 연세대학교 전자공학과 졸업 (공학사)

1982년 : 연세대학교 대학원 전자공학과 졸업 (공학석사)

1990년 : Univ. of Rochester, Institute of optics (공학박사)

1982년~2000년 : 대우중공업 근무

2001년~2007년 : 고등기술연구원 포토닉스센터 근무

2003년~현재 : 아주대학교 시스템공학과 (겸임교수)

2008년~현재 : (주)한화 종합연구소 재직 (상무보)  
<주관심분야> 광전자, 광학센서, 항법센서, 레이저응용분야