

잡음 발생기의 저면적, 저전력 방안에 관한 연구 (A Study on Low Area/Power Schemes of Noise Generation System)

이 창 기(Chang-Ki Lee)¹⁾

요 약

통신 시스템의 성능은 여러 가지 요구 조건을 고려하여 측정되어야 한다. 이러한 목적으로 잡음 발생기는 주어진 특성을 갖는 잡음 신호를 생성하는데 사용되어진다. 최근의 연구에서 잡음 모델이 복잡한 PSD를 갖는 경우 DCT를 이용한 잡음 발생기가 기존의 잡음 발생기보다 우수한 성능을 보였다. 본 논문에서는 DCT 기반 잡음 발생기에서 DCT를 제외한 회로의 면적을 줄일 수 있는 구조와 전력을 최소화시킬 수 있는 구조를 제안한다. 모의실험에서 면적을 최소화하기 위한 구조는 61~64%의 면적을 줄였으며, 전력을 고려한 구조에서는 88~89% 정도의 전력소모를 감소를 예측할 수 있다.

ABSTRACT

The performance of communication systems should be tested against a set of requirements. To this end, noise generation systems are used to generate noise signals with specified characteristics. In recent study, noise generation system using DCT outperforms the conventional noise generation system when a noise model requires complicated PSD(Power Spectral Density) specifications. In this paper, low area/power structures of non-DCT block in DCT-based noise generation system are proposed. Simulation results show that the low area structure results in area reduction by 61~64% and the low power structure achieves power reduction by 88~89% except DCT blocks.

1. 서론

모뎀을 비롯한 통신장비는 선로잡음이나 기타 잡음의 영향을 고려하여 시스템을 측정하고 평가해야 한다. 그러나 실제의 잡음 환경을 구현하고 측정하기란 매우 어렵고, 비용 또한 무시할 수 없는 경우가 많다. 따라서 통신 시스템을 측정할 때

는 신호에 영향을 주는 요소들을 인위적으로 만들어 실제 전송되는 신호를 가정하게 되는데, 이때 요구되는 것이 잡음 발생기이다.

고속모뎀의 테스트에 사용하는 VDSL의 선로잡음은 AWGN, 누화, 라디오 잡음, 임펄스 잡음, 배경 잡음 등이 있다. 일반적으로 잡음 발생을 위해 Yule-Walker equation을 사용하여 디지털 필터를

1) 정회원 : 서남대학교 컴퓨터 정보통신과 교수

구현할 수 있다 [1, 2]. 그러나 일반적으로 천이 대역폭이 좁으며 요구되는 주파수특성이 주파수에 대해 빠르게 변화할 경우 이를 만족시키는 디지털필터를 구현하기는 어려우므로 이 방법을 이용하여 VDSL에 적용하기 위한 잡음발생기의 필터 계수를 구하면 VDSL의 잡음 스펙과 오차가 심하게 발생되어 잡음 신호로서 부적합한 경우가 있게 된다. 이러한 문제점을 해결하기 위해서 Yule-Walker equation을 이용하는 대신 중심 극한 정리(Central Limit Theorem)와 DCT(Discrete Cosine Transform)를 이용하여 잡음신호를 발생시키는 알고리즘이 최근에 제안되었다 [3].

본 논문에서는 참고문헌[3]에 제안된 잡음 발생기의 면적을 최소화시킬 수 있는 구조와 전력소모를 감소시킬 수 있는 구조를 제안한다. 면적을 고려한 구조에서는 DCT 블록에서 발생되어 Mux에 입력되는 중복입력 신호를 이용하고, Adder Tree에서 수행되는 덧셈의 순서를 변화시킴으로써 DCT 블록을 제외한 나머지 회로의 면적이 감소되어진다. 또한, 전력을 고려한 구조는 기존의 잡음발생기에서 요구되었던 빠른 내부 클럭을 제거하여 회로의 스위칭 동작을 감소시켜 전력소모를 감소시켰다.

2절에서는 기존의 잡음 발생기에 대하여 간략

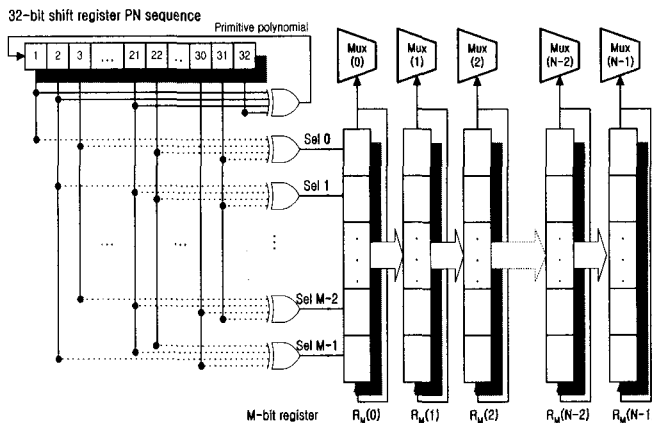
하게 설명하고, 3절에서는 면적과 전력소모를 고려한 구조들에 대하여 기술한다. 그리고 IV절에서는 Synopsys CAD tool을 이용한 구조별 면적 및 전력소모에 대한 실험 결과를 검토하며 V절에서 결론을 맺는다.

2. DCT 기반의 잡음발생기

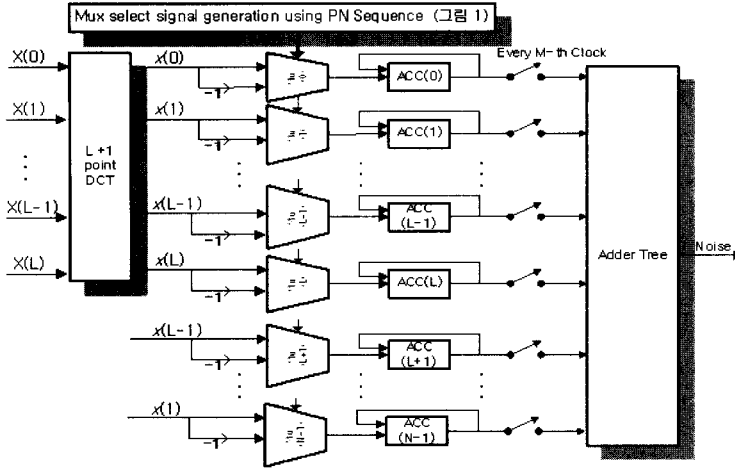
2.1 중심 극한 정리를 이용한 Gaussian 분포를 갖는 독립변수 발생기

일반적으로 통신시스템 측정에 사용되는 잡음 신호는 Gaussian 분포를 가진다 [4]. Uniform 분포를 갖는 신호를 Gaussian 분포를 갖는 신호로 변환하기 위해 Box-Muller equation을 사용할 수 있다 [5]. 그러나 Box-Muller equation은 sin, cos, log 함수를 계산하기 위한 ROM 테이블과 복잡한 곱셈기를 필요로 하기 때문에 하드웨어적으로 많은 면적을 차지하게 된다.

서로 독립인 M개의 독립변수(Random Variables) $X_i, i=1, 2, \dots, M$ 가 주어지고 M이 충분히 클 경우, 이 신호의 합인 $X = X_1 + \dots +$



[그림 1] PN 시퀀스 발생기를 이용한 Mux 선택신호 발생기
[Fig. 1] MUX select signal generation using PN sequence



[그림 2] DCT를 이용한 잡음발생기의 구조
 [Fig. 2] Noise generation block diagram using DCT

X_M 는 중심 극한 정리에 의하여 Gaussian 분포를 갖는 신호를 만들어낸다 [5].

[그림 1]은 uniform 분포를 갖는 M 개의 독립 변수를 발생시키기 위한 회로를 보여주고 있다. 여기에서 M 개의 독립변수는 PN 시퀀스 발생기를 사용하여 발생시킨다. 실선을 입력으로 갖는 XOR 게이트는 32비트 쉬프트 레지스터를 이용하여 원시 다항식(primitive polynomial)을 발생시키기 위한 것이며, 점선을 입력으로 갖는 M 개의 XOR 게이트는 서로 독립적인 uniform 분포를 가지는 신호를 만들어 내기 위한 것이다. 이때 XOR 게이트에 입력되는 신호는 서로 각기 다른 패턴을 가져야 한다.

M 개의 XOR 게이트의 출력은 M 비트 쉬프트 레지스터 $R_M(0)$ 에 저장되고 M -clk 동안 레지스터 내에서 쉬프트 되면서 Mux(0)의 선택신호로 사용된다. 그 후 $R_M(0)$ 의 데이터는 $R_M(1)$ 로 이동되고 XOR 게이트의 새로운 출력이 $R_M(0)$ 에 저장된다. $R_M(1)$ 의 데이터는 M -clk 동안 Mux(1)의 선택 신호로 사용되고 $R_M(2)$ 로 이동한다. 즉, $R_M(i)$ 의 데이터는 M -clk 동안 Mux(i)의 선택 신호로 사용되고 $R_M(i+1)$ 로 이동한다.

2.2 DCT를 이용한 잡음 발생기

DCT를 이용한 잡음발생기의 구조는 그림 2와 같다. $(L+1)$ -point DCT를 이용한 잡음 발생 알고리즘은 다음과 같다 ($L = N/2$).

- ① 주어진 PSD (Power Spectral Density) 스펙트럼으로부터, $(L+1)$ 개의 샘플 $X(0), X(1), \dots, X(L)$ 을 구한다.
- ② $(L+1)$ 개의 샘플 값을 입력으로 하는 $(L+1)$ -point DCT를 취한다.
- ③ 그림 1에서 발생된 N 개의 출력을 그림 2의 Mux 선택 신호의 입력으로 사용한다.
- ④ 그림 2와 같이 N 개의 신호를 더하여 실수 값의 시간영역 신호를 얻을 수 있다.

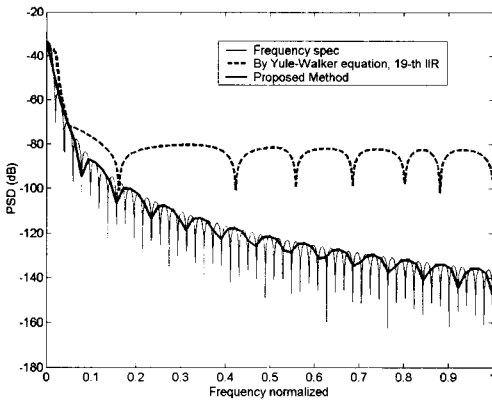
그림 2에서 PSD=1을 갖는 PN 시퀀스 발생기는 32-bit 쉬프트 레지스터가 사용되었다. Mux의 선택신호는 그림1에서 보여준 M 개의 4-input XOR 게이트를 사용하여 얻을 수 있다. Adder Tree 앞에 있는 스위치가 열려있는 동안, Mux 출력 값은 M -clk 동안 ACC(Accumulator) 블록

내에 누적된다. M -clk 후에 스위치가 연결되고 잡음 출력신호는 ACC 블록에 누적된 N 개의 신호를 Adder Tree에서 합하여 얻게 된다.

PN 시퀀스는 '1' 과 '0' 의 논리 값으로 이루어져 있으므로, 잡음을 발생시키기 위해 복잡한 곱셈기를 사용하는 대신 단순한 Mux를 사용하여 곱셈을 수행할 수 있다. 반면에 Box-Muller equation을 사용하여 Gaussian 분포를 갖는 잡음 신호를 얻기 위한 시스템을 구성할 경우에는 복잡한 곱셈기를 사용해야하기 때문에 효율성이 떨어지게 된다.

Yule-Walker equation과 33-point ($N=64$) DCT를 이용한 알고리즘을 사용하여 발생시킨 HDSL과 VDSL 간의 누화잡음 신호의 PSD를 [그림 3]에 보였다.

시뮬레이션 결과를 통하여 DCT를 이용한 알고리즘이 Yule-Walker equation 보다 원하는 PSD 스펙에 근접한 신호를 발생시킴을 알 수 있다.



[그림 3] Yule-Walker equation과 DCT를 이용한 알고리즘에서 얻은 HDSL에서 VDSL로의 누화잡음 신호의 PSD (Sampling frequency = 40 MHz)

[Fig. 3] PSDs of the crosstalk noise signal from HDSL to VDSL obtained by the Yule-Walker equation and the DCT-based algorithm(sampling frequency = 40MHz)

3. 잡음발생기의 저면적, 저전력 구조

3.1 저면적 구조

[그림 2]에서 Mux의 동작특성을 살펴보면, PN 시퀀스를 이용한 Mux 선택 신호가 '0' 인 경우에는 DCT 블록의 출력 $x(j), j=0, 1, \dots, L-1, L$ 가 선택되고 '1' 인 경우에는 $-x(j)$ 가 선택되어 진다. 따라서 Mux 선택 신호 발생기의 M -bit 쉬프트 레지스터에 저장되어 있는 '0' 과 '1' 에 따라 ACC 블록에서 M -clk 동안 덧셈과 뺄셈을 수행하게 된다. 이러한 동작을 수행하기 위해 메인클럭보다 M 배 빠른 클럭신호를 사용하여 쉬프트 레지스터에 저장된 값을 순환 시켜줘야 하며, 이 동작에 따라 Mux의 출력 신호가 결정되고 ACC 블록도 M 배 빠른 클럭에 동기되어 동작되는 것을 알 수 있다.

M -clk 동안, $ACC(0), ACC(1), \dots, ACC(L-1)$ 블록과 $ACC(L), ACC(L+1), \dots, ACC(N-1)$ 블록에 저장되는 신호 값은 다음과 같다.

$$\begin{aligned}
 ACC(0) &= x(0)R_M(0)_0 + x(0)R_M(0)_1 + \dots + x(0)R_M(0)_{M-1} \\
 ACC(1) &= x(1)R_M(1)_0 + x(1)R_M(1)_1 + \dots + x(1)R_M(1)_{M-1} \\
 &\vdots \\
 ACC(L-1) &= x(L-1)R_M(L-1)_0 + x(L-1)R_M(L-1)_1 + \dots + x(L-1)R_M(L-1)_{M-1}
 \end{aligned} \tag{1}$$

$$\begin{aligned}
 ACC(L) &= x(L)R_M(L)_0 + x(L)R_M(L)_1 + \dots + x(L)R_M(L)_{M-1} \\
 ACC(L+1) &= x(L-1)R_M(L+1)_0 + x(L-1)R_M(L+1)_1 + \dots + x(L-1)R_M(L+1)_{M-1} \\
 &\vdots \\
 ACC(N-1) &= x(1)R_M(N-1)_0 + x(1)R_M(N-1)_1 + \dots + x(1)R_M(N-1)_{M-1}
 \end{aligned}$$

따라서 식(1)로부터 M -clk 후, $ACC(i)$ 에서 발생하는 신호는 식(2)와 같이 정의할 수 있다.

$$\begin{aligned}
 ACC(i) &= x(j)R_M(i)_0 + x(j)R_M(i)_1 + \dots + x(j)R_M(i)_{M-2} + x(j)R_M(i)_{M-1} \\
 &= x(j) \sum_{k=0}^{M-1} R_M(i)_k, \quad i = 0, 1, \dots, N-1, \\
 &\begin{cases} \text{if } i \leq L, j = i, \\ \text{if } i > L, j = N-i. \end{cases}
 \end{aligned} \tag{2}$$

여기에서, $x(j)$ 는 DCT 블록의 j 번째 출력, $R_M(i)_k$ 는 $R_M(i)$ 쉬프트 레지스터의 k 번째 레지스터 값이 '0' 인 경우 '1' 이고 '1' 인 경우 '-1' 이 된다.

잡음발생기의 최종출력은 [그림 2]에서 $ACC(i)$ 에 저장된 N 개의 모든 신호들을 메인클럭 1-clk 마다 식(3)과 같이 Adder Tree 블록에서 동시에 더하여 얻게 된다.

$$Out(t) = \sum_{i=0}^{N-1} ACC(i)_t \quad (3)$$

식(1)과 식(2)를 살펴보면 $x(0)$ 과 $x(L)$ 을 제외한 $(L-1)$ 개의 $x(1), x(2), \dots, x(L-1)$ 은 각각 Mux(1)과 Mux(N-1), Mux(2)와 Mux(N-2), ..., Mux(L-1)과 Mux(L+1)의 입력신호로 사용되어 한 개의 DCT 출력이 동시에 두 개의 Mux 입력으로 인가됨을 알 수 있다.

어느 한 시점에서 Mux에 중복되는 입력을 고려하여 식(2)를 이용하여 식(3)을 전개하면 다음 식(4)와 같이 된다.

$$\begin{aligned} Out = & x(1)[\{R_M(1)_0 + R_M(N-1)_0\} \\ & + \{R_M(1)_1 + R_M(N-1)_1\} + \\ & \dots + \{R_M(1)_{M-1} + R_M(N-1)_{M-1}\}] \\ & + x(2)[\{R_M(2)_0 + R_M(N-2)_0\} \\ & + \{R_M(2)_1 + R_M(N-2)_1\} + \\ & \dots + \{R_M(2)_{M-1} + R_M(N-2)_{M-1}\}] \\ & \vdots \\ & + x(L-1)[\{R_M(L-1)_0 + R_M(L+1)_0\} \\ & + \{R_M(L-1)_1 + R_M(L+1)_1\} + \\ & \dots + \{R_M(L-1)_{M-1} + R_M(L+1)_{M-1}\}] \\ & + x(0)[R_M(0)_0 + R_M(0)_1 + \\ & \dots + R_M(0)_{M-1}] + x(L)[R_M(L)_0 + R_M(L)_1 + \\ & \dots + R_M(L)_{M-1}] \end{aligned} \quad (4)$$

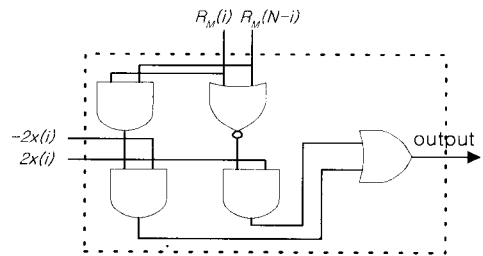
식(4)에서 $x(0)$ 와 $x(L)$ 에 해당되는 식을 제외한 나머지 $x(1) \sim x(L-1)$ 에 해당되는 식을 살펴보면, $\{R_M(i) + R_M(N-i)\}$ 항은 [그림 2]의 Mux (i)와 Mux(N-i)에 대응되며, [그림 1]의 Mux 선

택신호 발생기의 $R_M(i)$ 번째 쉬프트 레지스터와 $R_M(N-i)$ 번째 쉬프트 레지스터로부터 '0' 혹은 '1' 신호를 입력받게 된다. 따라서 Mux(i)와 Mux(N-i)의 선택신호로 인가되는 $R_M(i)$ 와 $R_M(N-i)$ 가 가질 수 있는 선택신호의 조합과 그 결과 값은 <표 1>과 같이 된다.

<표 1> 선택신호에 따른 Mux 출력
(Table 1) Mux output by selection signal

$R_M(i)$	$R_M(N-i)$	Mux(i) 출력
0	0	$2x(i)$
0	1	0
1	0	0
1	1	$-2x(i)$

Mux의 선택신호가 '01' 인 경우에는 $x(i)$ 와 $-x(i)$ 를 더하게 되고, '10' 인 경우에는 $-x(i)$ 와 $x(i)$ 를 더하게 되어 결과 값은 0이 된다. 따라서 이 두 경우에는 ACC 블록에서 덧셈을 할 필요가 없다. 신호가 '00' 일 경우에는 $x(i)$ 를 두 번 더한 결과가 되므로 $2x(i)$ 를 Mux 입력신호로 사용할 수 있다. 마찬가지로 신호가 '11' 인 경우에는 $-2x(i)$ 를 Mux의 입력신호로 사용할 수 있다. $2x(i)$ 와 $-2x(i)$ 는 회로구현 관점에서 살펴보면 $x(i)$ 와 $-x(i)$ 를 왼쪽으로 1비트 이동시킴으로써 간단하게 구현할 수 있다.



[그림. 4] 중복입력을 고려한 Mux 회로
[Fig. 4] Mux circuit that consider repetition input

이러한 특성을 이용하여 공통된 $x(i)$ 의 입력을 갖는 두개의 Mux(i)와 Mux(N-i)를 [그림 4]와 같이 선택신호에 따라 $2x(i)$, $-2x(i)$, 0을 출력하는 하나의 Mux 회로로 수정할 수 있다.

따라서 중복입력의 특성을 이용하여 [그림 2]에서 사용되었던 N개의 Mux와 ACC 블록 대신에 (L+1)개만의 Mux와 ACC 블록을 이용하여 똑같은 동작을 수행시킬 수 있다.

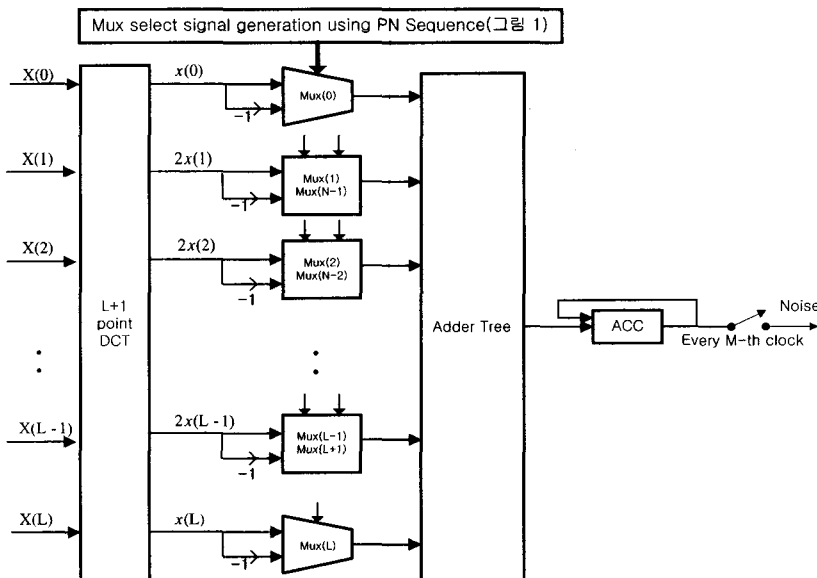
또한, 하드웨어 면적을 좀 더 감소시키기 위해 식(4)의 덧셈 순서를 식(5)와 같이 변환할 수 있다.

Out =

$$\begin{aligned}
 & [x(0)R_M(0)_0 + x(1)\{R_M(1)_0 + R_M(N-1)_0\} \\
 & + x(2)\{R_M(2)_0 + R_M(N-2)_0\} + \\
 & \dots + x(L-1)\{R_M(L-1)_0 + R_M(L+1)_0\} \\
 & + x(L)R_M(L)_0] \\
 & + [x(0)R_M(0)_1 + x(1)\{R_M(1)_1 + R_M(N-1)_1\} \\
 & + x(2)\{R_M(2)_1 + R_M(N-2)_1\} + \\
 & \dots + x(L-1)\{R_M(L-1)_1 + R_M(L+1)_1\} \\
 & + x(L)R_M(L)_1] \\
 & \vdots \\
 & + [x(0)R_M(0)_{M-1} + x(1)\{R_M(1)_{M-1} + R_M(N-1)_{M-1}\} \\
 & + x(2)\{R_M(2)_{M-1} + R_M(N-2)_{M-1}\} + \\
 & \dots + x(L-1)\{R_M(L-1)_{M-1} + R_M(L+1)_{M-1}\} \\
 & + x(L)R_M(L)_{M-1}]
 \end{aligned} \tag{5}$$

즉, 식(4)에서는 메인클럭보다 M배 빠른 M-clk를 사용하여 $x(0) \sim x(L)$ 에 해당되는 (L+1)개의 ACC(0) ~ ACC(L) 출력신호를 개별적으로 동시에 발생시켜 Adder Tree에서 (L+1)개의 모든 ACC 출력신호를 합산하여 하나의 잡음발생 출력을 발생시킨다. 그러나 식(5)에서는 그림 1에서 발생하는 N개의 0번째 레지스터에서 발생하는 Mux 선택신호를 사용하여 (L+1)개의 Mux 출력 값을 발생시켜 ACC 블록에서 덧셈을 수행하고 그 결과 값을 저장시키며, 다음에는 N개의 1번째 레지스터에서 발생하는 Mux 선택신호를 사용하여 (L+1)개의 Mux 출력 값을 발생시켜 ACC 블록에 저장된 값과 덧셈을 수행한 후 다시 결과 값을 저장한다. 이러한 동작을 M-clk 동안 M-1번째 레지스터까지 수행한다. 따라서 식(4)를 이용한 Mux에 연결된 (L+1)개의 ACC 블록을 모두 없애고 Adder Tree 블록 뒤에 한 개의 ACC를 사용하여 잡음 발생기를 구현할 수 있다.

Mux에 대한 중복입력 특성과 Adder Tree에서의 덧셈의 순서 변환을 적용하여 [그림 5]와 같은 구조를 제안하였다. [그림 5]를 살펴보면, Mux는



[그림 5] 저면적 잡음 발생기의 구조
 [Fig. 5] Noise generation block diagram of low area

모두 $(L+1)$ 개만을 사용하였으며, 식(4)에서의 Mux에 대한 중복입력을 고려하여 Mux(0)와 Mux(L)을 제외한 $(L-1)$ 개의 Mux는 [그림 4]의 수정된 회로를 적용하였다. 또한 식(5)의 ACC 블록에서의 덧셈 순서를 변화시킴으로써 $(L+1)$ 개의 ACC 블록을 Adder Tree 뒷단에 하나의 ACC 블록으로 대체시켰음을 알 수 있다. [그림 2]의 기존의 잡음 발생기 구조와 비교를 해보면, $(L-1)$ 개의 Mux와 $(N-1)$ 개의 ACC 블록을 감소시켰다.

3.2 저전력 구조

식(1)을 살펴보면 $ACC(i)$ 의 결과 값은 $x(j)$ 와 $R_M(i)$ 에 저장된 '0'의 개수와 '1'의 개수의 차와의 곱셈으로 발생됨을 알 수 있다.

즉, M배 빠른 클럭 신호를 이용하여 $ACC(i)$ 출력을 얻는 대신 $R_M(i)$ 에 저장된 '0'의 개수와 '1'의 개수의 차를 계산하여 shift-and-add 동작을 이용한 곱셈을 수행하여 결과 값을 얻을 수 있다. 이러한 동작을 수행하기 위해 [그림 1]의 PN 시퀀스 발생기를 [그림 6]과 같이 수정하였다.

독립변수의 개수를 M, [그림 6]에서 XOR된 독립변수의 값을 $S_0, S_1, \dots, S_{M-2}, S_{M-1}$, 그리고 LFSR에서 XOR되어서 나오는 1의 개수를 R이

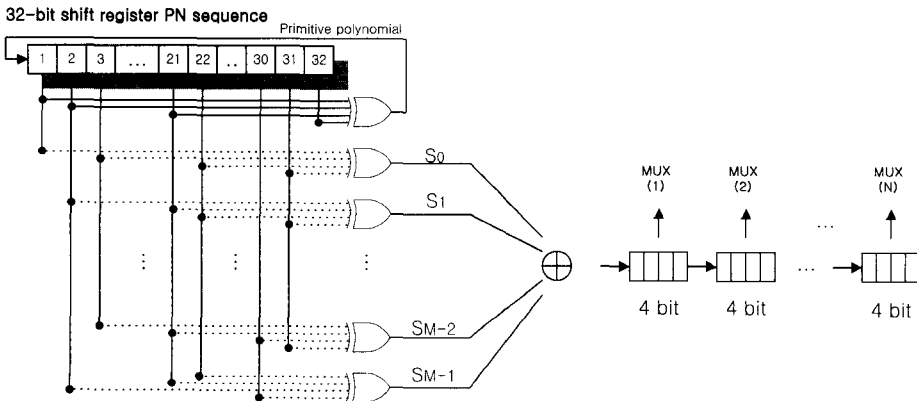
라고 할 때, $R = S_0 + S_1 + \dots + S_{M-2} + S_{M-1}$ 가 된다. 따라서 $\sum_{k=0}^{M-1} R_M(i)_k$ 의 결과 값을 P라 할때 $P = M - 2R$ 이 성립된다. M=10이라 가정하면, P는 다음의 <표 2>와 같이 결정된다.

<표 2> M = 10인 경우 P 값
(Table 2) P value when is M = 10

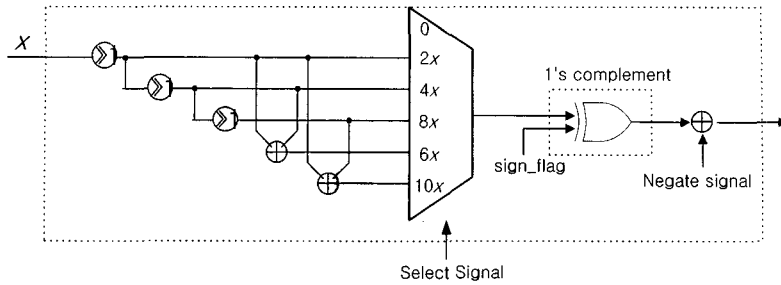
1의 개수 표현 = R	Operation(p)
1010 (10)	-10x
1001 (9)	-8x
1000 (8)	-6x
0111 (7)	-4x
0110 (6)	-2x
0101 (5)	0
0100 (4)	2x
0011 (3)	4x
0010 (2)	6x
0001 (1)	8x
0000 (0)	10x

서로 Symmetric한 특성을 가진다.

<표 2>에서 음수인 경우와 양수인 경우, 그리고 '1'의 개수와 '0'의 개수가 같을 경우로 나누어서 생각할 수 있다. (여기에서 x는 DCT의 출력)



[그림 6] 수정된 PN 시퀀스 발생기
[Fig. 6] Modified PN sequence generator

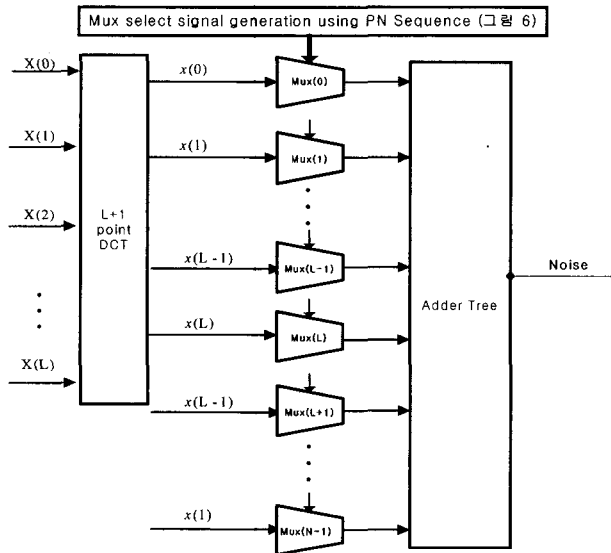


[그림 7] Shift-and-add 곱셈을 이용한 Mux 블록
 [Fig. 7] Mux block using Shift-and-add multiplication

- | | |
|--|---|
| <ul style="list-style-type: none"> i) $R < 5$인 경우 : 양의 값을 갖기 때문에 $2x, 4x, 6x, 8x, 10x$를 각각 계산한다. ii) $R = 5$인 경우 : '1'의 개수와 '0'의 개수가 같기 때문에 '0' 값을 출력한다. iii) $R > 5$인 경우 : 음의 값을 갖기 때문에 $2x, 4x, 6x, 8x, 10x$를 계산한 다음 그 값에 대한 보수를 취해준다. | <ul style="list-style-type: none"> i) $2x$: x의 값을 1-비트 왼쪽으로 이동(shift)시킨다. ii) $4x$: x의 값을 2-비트 왼쪽으로 이동시킨다. iii) $8x$: x의 값을 3-비트 왼쪽으로 이동시킨다. iv) $6x$: $2x$와 $4x$의 값을 덧셈(add)한다. v) $10x$: $2x$와 $8x$의 값을 덧셈한다. |
|--|---|

위의 값들은 곱해지는 하나의 값(DCT의 출력 $x(j)$)이 정해져 있기 때문에 다음과 같이 shift-and-add 곱셈 방법을 이용하여 구현할 수 있다.

[그림 7]은 위에서 설명한 shift-and-add 방식을 이용한 수정된 Mux 블록을 보여준다. <표 2>에서 P의 동작은 0을 중심으로 symmetric한 특징을 가지고 있음을 알 수 있다. 따라서 음의



[그림 8] 저전력 잡음발생기 구조
 [Fig. 8] Noise generation block diagram of low power

값을 곱하는 경우에는 R의 조건에 따라 sign_flag 신호를 발생시켜, 양의 값을 1의 보수로 바꾼 후 negate 신호를 더하여 2의 보수를 발생시킨다.

[그림 6]의 PN 시퀀스 발생기와 [그림 7]의 Mux 블록을 적용하고, M-clk을 사용하는 ACC 블록을 제거한 전력을 고려한 잡음발생기를 [그림 8]에 보여준다. 이 잡음발생기는 PN 시퀀스 발생기에 사용되는 쉬프트 레지스터의 크기 M과 관련된 M배의 빠른 클럭을 사용하지 않고 메인클럭에 동기되어 출력이 발생되므로 회로의 동작속도를 높일 수 있고 전력소모를 줄일 수 있는 장점을 갖고 있음을 알 수 있다.

4. 모의실험 결과 및 검토

〈표 3〉과 〈표 4〉는 기존의 구조[그림 2], 면적을 고려한 잡음발생 구조[그림 5], 그리고 전력을 고려한 잡음발생 구조[그림 8]에서 DCT 블록을 제외한 회로의 면적과 전력을 비교한 것이다. 각각의 회로는 VHDL로 코딩하였으며, Synopsys CAD tool을 이용하여 회로 합성과 전력소모에 대한 계산을 수행하였다. 여기에서 사용된 target library는 Synopsys에서 제공하는 class library이다.

〈표 3〉의 각 구조에 따른 면적을 살펴보면, 기존의 구조에 비해 [그림 5]의 구조가 면적에서 61~64% 정도 감소되었다. 이것은 면적을 고려한 구조가 기존의 구조에서 사용된 (L-1)개의 Mux와 (N-1)개의 ACC 블록을 줄임으로써 저면적을 갖게 되었다. 또한 [그림 8]의 구조는 기존의 구조에 비해 약 13% 정도의 면적감소를 보였다. 이것은 기존 구조에 사용되었던 ACC 블록을 제거하고 그 대신 shift-and-add 동작을 위한 쉬프트 레지스터와 덧셈기를 포함하는 Mux 블록을 사용하였기 때문에 많은 면적을 감소시키지는 못하였다. 〈표 3〉의 게이트 수는 2-입력

nand 게이트로 표준화하여 비교하였다.

〈표 3〉 구조별 면적 비교 (Gate 수)
〈Table 3〉 Area comparison in structure (Gate number)

수구분 \ DCT point	16	32	64
그림 2 구조-①	16316	32645	64143
그림 5 구조-②	6316	12207	23143
그림 8 구조-③	14304	28137	55446
②:① 감소율(%)	61.3	62.4	63.9
③:① 감소율(%)	12.3	13.3	13.6

〈표 4〉에 보여준 구조별 전력소모를 살펴보면, [그림 8]의 구조에서는 기존의 구조에서 사용되었던 M-clk을 제거함으로써 약 88% 정도의 전력을 감소시켰다. 또한 [그림 5]의 구조는 41~56% 정도의 전력을 감소시켰다. 이것은 [그림 5]의 구조가 기존의 구조에서 사용된 (L-1)개의 Mux와 (N-1)개의 ACC 블록을 줄임으로써 저면적을 갖게 되었다. 그러나 이 구조는 critical path가 Adder Tree 블록까지 길어지게 되어 전력소모가 면적 감소 비율에 비해 조금 낮은 감소 수치를 보였다.

〈표 4〉 구조별 전력 비교 (uW)
〈Table 4〉 Power comparison in structure(uW)

수구분 \ DCT point	16	32	64
그림 2 구조-①	234,1793	453,7214	869,5370
그림 5 구조-②	139,4519	233,2723	381,5276
그림 8 구조-③	29,1155	52,4499	92,5048
②:① 감소율(%)	40.5	48.6	56.1
③:① 감소율(%)	87.6	88.34	89.4

5. 결 론

본 논문에서는 DCT를 이용한 잡음 발생회로의 면적과 전력을 감소시키기 위한 구조적인 방안을 제안하였다. 면적을 고려한 잡음 발생기는 DCT 블록에서 발생되어 Mux의 입력으로 인가되는 신

호의 중복성과 회로의 마지막 단을 이루는 Adder Tree에서 수행되는 덧셈의 순서를 변화시킴으로써 DCT 블록을 제외한 나머지 회로의 면적을 61~64% 정도 감소시킬 수 있었다. 또한 전력을 고려한 잡음 발생기는 shift-and-add 곱셈 구조를 이용하여 기존의 구조에서 사용한 메인 클럭보다 M배 빠른 M-clk을 제거함으로써 전력소모를 88~89% 정도 감소시킬 수 있음을 보였다.

※ 참고문헌

[1] J. S. Chow, *Finite-length equalization for multi-carrier transmission systems*, Ph.D Thesis, Stanford University, 1992.

[2] J. L. Dixon, J. S. Yeomans, and J. M. Goldthrop, "NEXTNOISE-a programmable noise generating system for testing wire-based loop transmission systems," *Proceedings of the Globecom' 93*, pp.1319-1324, Nov. 1993.

[3] K. Tae, D. Kim, and J. Chung, "Noise Generation System Using DCT," *IEICE Trans. on Communications*, vol. E84-B, no. 9, pp. 2697-2700, Sept. 2001.

[4] J. Cioffi, "Very-high-speed digital subscriber lines system requirements," *Committee T1-Telecommunications*, Nov. 1998.

[5] A. Papoulis, *Probability, Random Variable, and Stochastic Processes*, McGraw-Hill Inc., 1991.

[6] K. K. Parhi, *VLSI Digital Signal Processing System : Design and Implementation*, Willy-Interscience publication, 1999.

[7] H. Kitajima, A symmetric cosine transform, *IEEE Trans. Computer C-29*, 317-323, 1980.

[8] C. S. Burrus, "Digital filter structures described by distributed arithmetic," *IEEE Trans. on Circuits and Systems*, Dec. 1977.

[9] N. Demassieux and F. Jutand, "Orthogonal transforms," in *VLSI Implementations for Image Communications* (P. Pirsch, ed.), pp.217-250, Elsevier, 1993.

이 창 기

1988년 전북대학교 공과대학 전자공학과 졸업(학사)
 1990년 전북대학교 대학원 전자공학과 졸업(공학석사)
 1994년 전북대학교 대학원 전자공학과 졸업(공학박사)
 2000년 ~ 현재 서남대학교 전자계산소 과장
 1995년 ~ 현재 서남대학교 컴퓨터 정보통신학과 교수
 관심 분야 : 반도체 회로 설계, 신호처리