

논문 2008-45SD-12-11

QPP 주기성을 이용한 저전력 QPP 인터리버 주소발생기 설계

(A Low Power QPP Interleaver Address Generator Design Using The Periodicity of QPP)

이 원 호*, 임 중 석**

(Won-Ho Lee and Chong-Suck Rim)

요 약

QPP 인터리버는 고속 병렬 터보 디코더에서 메모리 경합 없는 인터리빙 기능을 제공할 수 있어 주목을 받고 있다. 본 논문에서는 QPP 인터리버의 주소 생성 함수 $f(x) = (f_1x + f_2x^2) \% K$ 의 이차항 $f_2x^2 \% K$ 가 아주 작은 주기를 갖는다는 것을 보이고, 이러한 주기성을 이용하여 설계한 저전력 주소 생성기를 소개한다. 소개한 주소 생성 방법에서는 처음 반주기 동안의 $f_2x^2 \% K$ 값들을 메모리에 저장하고 이 값들은 읽어 $f(x)$ 를 계산함으로써 $f_2x^2 \% K$ 값들의 계산 없이 주소를 생성한다. 이렇게 설계한 주소 생성기는 일반적인 방법에 의한 설계에 비하여 평균 5.54%(가변 K 인 경우)와 27.38%(고정 K 인 경우)의 전력 소모 절감 효과를 갖는다.

Abstract

The QPP interleaver has been gaining attention since it provides contention-free interleaving functionality for high speed parallel turbo decoders. In this paper we first show that the quadratic term $f_2x^2 \% K$ of $f(x) = (f_1x + f_2x^2) \% K$, the address generating function, is periodic. We then introduce a low-power address generator which utilizes this periodic characteristic. This generator follows the conventional method to generate the interleaving addresses and also to save the quadratic term values during the first half of the first period. The saved values are then reused for generating further interleaved addresses, resulting in reduced number of logical operations. Power consumption is reduced by 27.38% in the design with fixed- K and 5.54% in the design with unfixed- K on average for various values of K , when compared with the traditional designs.

Keywords : QPP, Interleaver, Low Power Design, Quadratic Polynomial Permutation, Address Generator.

I. 서 론

오류정정부호로 개발된 터보 코드^[1]는 새년의 이론적 한계에 근접하는 우수한 성능을 보여 여러 통신 시스템에 널리 적용되고 있다. 터보 코드는 복호 방법으로 반복 복호 방식을 이용하기 때문에 기본적으로 긴 복호 시간과 낮은 처리율을 가진다. 따라서 높은 처리율을 요구하는 환경에서는 터보 복호기를 병렬로 구성할 필

요가 있으며, 터보 코드의 효율적인 병렬 복호 방법에 대한 많은 연구들이 이루어져 오고 있다.

이와 같이 터보 복호기를 병렬로 구성할 시에 인터리버의 선택에 따라 복호 구간 동안 메모리 충돌 문제가 발생할 수 있다^[2]. 복호 구간 동안 병렬로 구성된 MAP 복호기들로부터 계산된 부가 정보 값들은 서로 다른 저장 영역으로 쓰기 동작이 이루어져야 한다. 3GPP 표준으로 사용되는 Rel6 PIL^[3]이나 S-random 인터리버 등과 같은 인터리버 경우, MAP 복호기들로부터 계산된 부가 정보 값들이 동시에 같은 저장 영역으로 쓰기를 시도하는 경우가 발생하게 된다. 이와 같은 메모리 충돌 문제를 해결하기 위하여 주소발생 시작시간의 지연이나 추가적인 버퍼의 사용 등의 해결 방안들

* 정회원, (주)코닉시스템 통신기술연구소
(DL Team, KORNIC Systems Co., Ltd)

** 평생회원, 서강대학교 컴퓨터공학과
(Department of Computer Science and Engineering,
Sogang University)

접수일자: 2008년6월19일 수정완료일: 2008년12월1일

이 사용되고 있다^[4~5].

효율적인 고속 병렬 터보 복호기 구현을 위하여 메모리 충돌이 일어나지 않는 인터리버에 대한 연구^[6~7]들이 이루어져 왔고, 어떠한 병렬 복호 차수에도 메모리 충돌이 일어나지 않는 2차 다항식 순열(Quadratic Polynomial Permutation) 인터리버^[8~10]가 소개되었다. QPP 인터리버는 안정적인 성능^[11]과 단순한 구현 방법, 메모리 충돌이 없는 특성으로 인하여 3GPP LTE 표준으로 제안되었다.

2차 다항식 순열(QPP)은 수식 (1)과 같이 정의 된다^[9].

$$f(x) = (f_1x + f_2x^2) \% K \quad (1)$$

수식 (1)에서 K 는 인터리브 길이에 해당하고, x 의 계수 f_1, f_2 를 적절하게 선택함에 의하여 2차 다항식 $f(x)$ 는 2차 다항식 순열이 된다. 다양한 K 값에 대해서 [12]에 제안된 f_1, f_2 값들을 QPP 인터리버에 사용할 경우 기존의 Rel-6 인터리버보다 더 나은 성능을 보인다. 수식 (1)에서 보는 바와 같이 QPP 인터리버를 구현할 시에 $f(x)$ 값을 계산하기 위하여 모듈로 연산을 하게 된다. 곱셈이나 모듈로 연산 없이 $f(x)$ 값을 계산하기 위하여 이전 $f(x)$ 값를 이용하여 다음 $f(x)$ 값을 재귀적으로 계산하는 방법이 고안되었다^[13].

본 논문에서는 $f(x)$ 를 $f(0)$ 부터 인터리브 길이 $f(K-1)$ 까지 순차적으로 계산하였을 때, $f_2x^2 \% K$ 의 값들이 짧은 주기를 가지고 반복됨을 보이고, 또한 한 주기 내에서 그 값들이 대칭 형태를 가짐을 보인다. 그리고 이를 이용하여 [13]에서 보인 주소발생 방법보다 전력 소모 면에서 높은 절약 효과를 보이는 인터리버 설계 방법을 제안한다. 본 논문에서 제안하는 설계에서는 $f(x)$ 값을 순차적으로 계산할 때, 첫 반주기 동안만의 $f_2x^2 \% K$ 값들을 메모리에 저장하고 이 값들을 연속적으로 읽어 $f_1x \% K$ 값과 더하여 $f(x)$ 값을 계산한다. 따라서 $f_2x^2 \% K$ 값을 계산할 필요가 없거나 또는 짧은 반주기만 계산하면 되므로 논리적인 연산을 줄일 수 있다. 이러한 방법을 구현하여 시뮬레이션을 통하여 전력소모를 예측한 결과 일반적인 설계방법에 비하여 K 가 고정된 경우 평균 27.38%, K 가 가변적일 경우 평균 5.54%의 전력을 절약할 수 있다.

본 논문의 구성은 다음과 같다. 이어지는 II장에서는 2차 다항식 순열에서 $f_2x^2 \% K$ 값이 짧은 주기를 가짐

을 보이고, K, f_1, f_2 에 따른 $f_2x^2 \% K$ 의 최소 주기를 구하는 방법을 보인다. 또한 한 주기 내에서 $f_2x^2 \% K$ 값이 대칭적인 형태를 가짐을 보인다. 3절에서는 기존의 일반적인 설계 방법과 본 논문에서 제안하는 $f_2x^2 \% K$ 의 주기적인 성질을 이용한 설계를 보인다. IV장에서는 III장에서 제안한 설계에 대한 전력 소모량 비교 결과를 보이고 V장에서는 결론을 보인다.

II. $f_2x^2 \% K$ 의 주기적 특성

QPP $f(x) = (f_1x + f_2x^2) \% K$ 는 K 의 모듈로 함수이므로 주기함수이며 $f_1x \% K$ 와 $f_2x^2 \% K$ 역시 주기함수이다. 그런데 $f(x)$ 의 주기는 K 이지만 $f_2x^2 \% K$ 의 주기는 K 에 비해 아주 작은 값을 갖는다. 또한 $f_2x^2 \% K$ 는 한 주기 내에서 좌우 대칭인 특성이 있다.

함수 $f_2x^2 \% K$ 의 주기를 T 라고 하자. 그러면 임의의 양의 정수 x 에 대하여 $f_2x^2 \% K = f_2(x+T)^2 \% K$ 를 만족하여야 하며 이는 다음에 보이는 수식 (2)를 모두 만족 하여야 하는 것과 마찬가지 이다.

$$\begin{cases} 2f_2xT \% K = 0 \\ f_2T^2 \% K = 0 \end{cases} \quad (2)$$

수식 (2)를 만족하기 위해서는 $2f_2T$ 와 f_2T^2 모두 K 로 나누어 떨어져야 하며 따라서 수식 (3)을 만족하는 양의 정수 c_1, c_2 가 존재한다.

$$\begin{cases} 2f_2T = c_1K \\ f_2T^2 = c_2K \end{cases} \quad (3a,3b)$$

소수 p_1, p_2, \dots, p_m 을 K 의 소인수(prime factors)라고 하면 K 와 f_2 는 수식 (4)와 (5)와 같이 표시할 수 있다.

$$K = p_1^{d_1} p_2^{d_2} \dots p_m^{d_m} \quad (4)$$

$$f_2 = p_1^{e_1} p_2^{e_2} \dots p_m^{e_m} \cdot z \quad (5)$$

여기서, $d_i \geq 1$ 이고 $e_i \geq 0$ 이며, z 는 1 또는 p_i 와 서로소인 양의 정수이다 ($i = 1, 2, \dots, m$). 수식 (4)와 (5)를 수식 (3)의 첫째 식에 대입하면 주기 T 는 수식 (6)과 같은 형태를 갖는다.

$$T = \frac{c_1}{2z} \prod_{1 \leq i \leq m} p_i^{d_i - e_i} \quad (6)$$

정리 1. 소수 p_1 을 K 의 소인수중 가장 작은 소수라고 가정할 때 함수 $f_2 x^2 \% K$ 의 주기 T 는 수식 (7) 과 같이 계산된다.

$$T = \begin{cases} \frac{1}{2} \prod_{d_i > e_i} p_i^{d_i - e_i} & \text{if } p_1 = 2 \text{ and } d_1 - e_1 \geq 2 \\ \prod_{d_i > e_i} p_i^{d_i - e_i} & \text{if } p_1 \neq 2 \text{ or} \\ & (p_1 = 2 \text{ and } d_1 - e_1 < 2) \end{cases} \quad (7a, 7b)$$

증명. 주기 T 는 양의 정수이어야 하므로 이를 위한 c_1 과 c_2 값을 결정하여야 하며 이들 역시 양의 정수이어야 한다.

Case 1. $p_1 = 2$ 이고 $d_1 - e_1 \geq 2$.

만일 $d_i < e_i$ 인 경우 수식 (6) 에서 $p_i^{d_i - e_i}$ 는 1 보다 작은 수이므로 T 가 양의 정수이기 위해서는 $p_i^{e_i - d_i}$ 가 c_1 의 인수가 되어야 한다. 그리고 z 역시 c_1 의 인수가 되어야 한다. 따라서 c_1 을 수식 (8) 과 같이 설정한다.

$$c_1 = z \cdot \prod_{d_i < e_i} p_i^{e_i - d_i} \quad (8)$$

그러면 수식 (6) 에서 1 보다 작은 모든 인자가 c_1 에 의하여 약분되고, $p_1 = 2$ 이고 $d_1 - e_1 \geq 2$ 이므로 수식 (6) 의 분모에 있는 2 역시 약분될 수 있다. 따라서 이 경우 주기 T 를 수식 (7a) 로 설정한다.

이러한 설정으로 수식 (3b) 를 만족하는 양의 정수 c_2 가 존재하는지 알아보자. 수식 (3b) 로부터 $c_2 = f_2 T^2 / K = c_1 T / 2$ 이므로 수식 (7a) 를 수식 (9) 에 대입하면

$$c_2 = \frac{1}{4} c_1 \prod_{d_i > e_i} p_i^{d_i - e_i} \quad (9)$$

가 된다. 여기서, $p_1 = 2$ 이고 $d_1 - e_1 \geq 2$ 이므로 수식 (9) 의 분모 4 는 약분되며 따라서 c_2 는 양의 정수가 된다.

Case 2. $p_1 \neq 2$ 이거나 $p_1 = 2$ 이고 $d_1 - e_1 < 2$.

이 경우 수식 (6) 의 분모에 있는 2 가 c_1 에 의하여 약분되어야 하므로

$$T=2i+1 \quad x : 0 \ 1 \ 2 \ \dots \ i-1 \ i \ T-i \ T-(i-1) \ \dots \ T-2 \ T-1$$

$$T=2i \quad x : 0 \ 1 \ 2 \ \dots \ i-1 \ i \ T-(i-1) \ \dots \ T-2 \ T-1$$

그림 1. $f_2 x^2 \% K$ 의 한 주기 내에서의 대칭 형태
Fig. 1. Symmetric Shape of $f_2 x^2 \% K$ in a period.

$$c_1 = 2z \cdot \prod_{d_i < e_i} p_i^{e_i - d_i} \quad (10)$$

로 설정하고 주기 T 를 수식 (7b) 로 설정한다. 그리고 수식 (7b) 와 (10) 을 $c_2 = c_1 T / 2$ 에 대입하면

$$c_2 = \frac{1}{2} 2z \cdot \prod_{d_i < e_i} p_i^{e_i - d_i} \prod_{d_i > e_i} p_i^{d_i - e_i} \quad (11)$$

가 되어 c_2 는 양의 정수가 된다.

정리 1에서 구한 $f_2 x^2 \% K$ 의 주기는 실제 K 값에 비해 아주 작은 값을 갖는다. 최근 참고문헌 [12]에서는 3GPP용 터보 디코더를 위하여 여러 K 값에 대한 QPP 인터리버의 f_1 , f_2 값들을 제시하였다. 이러한 f_2 값들로 모든 K 에 대해 계산한 $f_2 x^2 \% K$ 의 주기를 그 크기에 따른 분포를 보면 표 1과 같다. 표 1에서 보이는 바와 같이 $f_2 x^2 \% K$ 의 주기는 대부분 16 이하이며 K 값에 대비하여 주기의 크기는 평균 0.79%에 불과하다.

함수 $f_2 x^2 \% K$ 가 K 값에 비해 아주 작은 주기를 갖는다는 사실 외에도 이 함수는 그림 1에서 보인 바와 같이 한 주기 내에서 좌우 대칭 형태를 갖는다. 정리 2에 이러한 특성을 보인다.

정리 2. $f_2 x^2 \% K$ 의 주기를 T 라고 할 때 이 함수는 다음의 등식을 만족한다.

$$f_2((n-1)T+x)^2 \% K = f_2(nT-x)^2 \% K \quad (12)$$

표 1. 모든 K 값에 대한 $f_2 x^2 \% K$ 의 주기 빈도
Table 1. The distribution of periods for the various K .

주기	빈도	주기	빈도
2	45	18	3
4	33	20	2
6	6	24	3
8	50	28	1
10	3	32	7
12	3	54	1
16	30	56	1

여기서 n 은 양의 정수이고 x 는 $1 \leq x \leq T-1$ 를 만족하는 정수이다.

증명. 임의의 양의 정수 a, b, m 에 대해 $a \% m = b \% m$ 이면 $(a-b) \% m = 0$ 임을 이용하여 수식 (12)를 전개하면 다음의 수식 (13)이 성립한다.

$$f_2((1-2n)T^2x^2 + 2(2n-1)Tx) \% K = 0 \quad (13)$$

$(a+b) \% m = (a \% m + b \% m) \% m$ 이므로 수식 (13)은 수식 (14)로 다시 쓸 수 있다.

$$((f_2((1-2n)T^2x^2) \% K + (2(2n-1)Tx) \% K) \% K) \% K = 0 \quad (14)$$

그런데 수식 (3a,b)에 의하여 $2f_2T$ 와 f_2T^2 모두 K 의 배수이므로 수식 (14)의 좌측 항은 모두 0이고 따라서 수식 (12)가 성립한다.

III. 저전력 QPP 인터리버 주소 발생기 설계

QPP를 통하여 인터리버 주소를 연속적으로 발생한 다음 함수 $f_2x^2 \% K$ 값을 계속 계산할 필요가 없다. 즉, 2장의 정리 1과 2에 의하여 $f_2x^2 \% K$ 의 처음 반주기 값만을 계산하여 메모리에 저장하고 이를 주소 생성에 사용하여 전력 소모를 줄일 수 있다. 본 절에서는 먼저 일반적인 QPP 인터리버 주소 발생기 설계에 대하여 간단히 기술하고, 다음 $f_2x^2 \% K$ 의 주기성을 이용한 저전력 주소 생성기의 설계에 대해 기술한다.

참고문헌 [13]에 $f(x) = (f_1x + f_2x^2) \% K$ 를 순환적으로 계산하는 방법을 제시하였다. 즉,

$$\begin{aligned} f(x+1) &= (f_1(x+1) + f_2(x+1)^2) \% K \\ &= (f_1x + f_2x^2 + 2f_2x + f_1 + f_2) \% K \end{aligned} \quad (15)$$

이므로, $g(x) = (2f_2x + f_1 + f_2) \% K$ 라고 정의하면

$$\begin{aligned} f(x+1) &= (f(x) + g(x)) \% K \\ g(x+1) &= (g(x) + 2f_2) \% K \end{aligned} \quad (16)$$

에 의하여 $f(x)$ 를 계산할 수 있다. 여기서, $f(0) = 0$, $g(0) = (f_1 + f_2) \% K$ 이다.

수식 (16)에 의한 구현은 주소 발생 전 구간에 걸쳐 회로의 모든 컴포넌트가 동작하게 된다. 그런데 $f_2x^2 \% K$ 의 주기성을 이용한다면 이의 계산을 미리 해 두거나 반주기만 계산하면 되기 때문에 보다 적은 전력

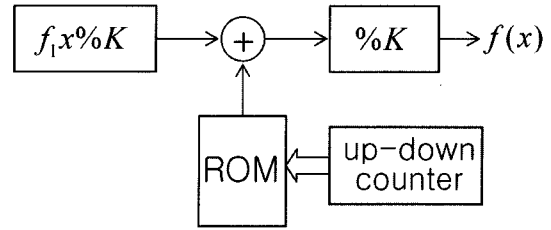


그림 2. K 값이 고정된 경우 주소발생기
Fig. 2. Address Generator for Fixed K .

으로 주소를 생성할 수 있다. 이러한 방식은 K 값이 고정된 경우와 고정되지 않은 두 형태가 있을 수 있다.

K 값이 고정된 경우에는 $f_2x^2 \% K$ 의 반주기 값들을 미리 메모리에 저장하여 $f_1x \% K$ 를 계산하여 이 값과 메모리에서 읽은 $f_2x^2 \% K$ 값을 더하여 주소를 생성할 수 있다. $f_1x \% K$ 값의 계산은 $p(x) = f_1x \% K$ 라고 하면

$$p(x+1) = (p(x) + f_1) \% K \quad (17)$$

에 의하여 순환적으로 계산할 수 있다. 이 경우 메모리는 ROM 또는 이와 동등한 회로를 사용할 수 있다. 그림 2에 이러한 주소발생기의 구성도를 보인다. 메모리는 up-down 카운터를 이용하여 계속 반복적으로 읽으면 충분하다. 그림에서 ' $\%K$ ' 블록은 입력이 K 보다 클 경우 K 를 빼서 출력하는 장치이다.

반면에 입력 패킷의 크기가 가변이라서 K 값이 고정되지 않은 경우에는 처음 반주기 동안은 $f_2x^2 \% K$ 값을 직접 계산하여 주소를 생성하고 동시에 이 값들을 메모리에 저장한다. 반주기 이후부터는 $f_2x^2 \% K$ 값을 생성하는 회로를 정지시키고 대신 메모리에서 이 값을 읽어 주소를 생성한다. $f_2x^2 \% K$ 값은 $p(x) = f_2x^2 \% K$ 이고, $q(x) = (f_2x + f_2) \% K$ 라고 하면

$$\begin{aligned} p(x+1) &= (p(x) + q(x)) \% K \\ q(x+1) &= (q(x) + 2f_2) \% K \end{aligned} \quad (18)$$

에 의하여 순환적으로 계산할 수 있다. 여기서 $p(0) = 0$, $q(0) = f_2$ 이다. 그런데 수식 (18)에 의한 $f_2x^2 \% K$ 값 하나를 위한 계산량은 상대적으로 클 수 있으나 표 1에 보인 바와 같이 $f_2x^2 \% K$ 의 반주기는 전체 주소 발생 기간에 비하여 아주 짧으므로 $f_2x^2 \% K$ 값의 계산량은 전체 계산량에 비하여 아주 작다. 그림 3에 이러한 K 값이 고정되지 않은 경우의 주소발생기 구성도를 보인다.

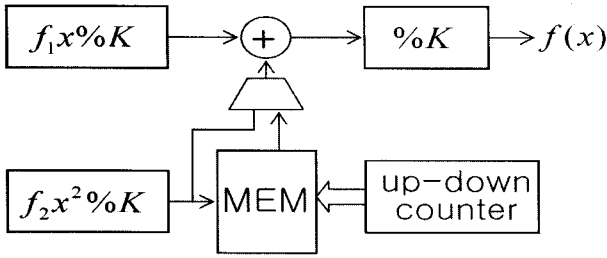


그림 3. 가변 K 값을 위한 주소 발생기
Fig. 3. Address Generator for Unfixed K .

IV. 실험 결과

수식 (16)에 의한 일반적인 구현 방법과 본 논문에서 제시한 두 가지 방법을 각각 VHDL로 구현하고 합성하여 Xilinx의 FPGA에 배치 및 배선을 수행하였다.

이 결과와 입력 패턴을 이용하여 post-layout 시뮬레이션을 통해 생성한 .vcd 파일과 Xilinx의 Xpower를 사용하여 전력 소모량을 비교하였다. 표 2에 11 개의 K 값에 대해 전력소모를 비교한 결과를 보인다. 표에서 보인 바와 같이 본 논문에서 제안한 주소 발생기는 두 경우 모두 수식 (16)에 근거하여 구현한 일반적인 주소 발생기에 비하여 각각 평균적으로 5.54%, 27.38% 전력을 덜 소비함을 알 수 있다. 특히 K 가 고정일 경우, ROM을 사용하여 구현한 설계는 실험한 모든 K 에 대하여 더 적은 전력을 소비함을 알 수 있다. 여기서 K 값은 참고문헌 [12]에서 제시한 값들의 일부를 선택하

표 2. 세 가지 설계들의 전력 소모 비교 (단위 mA)
Table 2. Comparison of power consumption (unit: mA).

K	일반적인 설계 (수식(16))	제안한 설계 (고정 K)		제안한 설계 (가변 K)	
	전력소모	전력소모	감소율	전력소모	감소율
40	10.77	7.43	31.01%	9.90	8.07%
256	11.16	7.35	34.13%	11.56	-3.58%
768	14.47	12.69	12.30%	14.66	-1.31%
1536	15.18	11.31	25.49%	15.37	-1.25%
2368	20.58	13.36	35.08%	16.84	18.17%
3136	15.48	11.81	23.70%	15.48	0.00%
3456	15.58	13.18	15.40%	16.30	-4.62%
4480	19.43	12.79	34.17%	17.04	12.30%
5184	15.20	12.06	20.65%	14.96	1.57%
5248	22.34	13.70	38.67%	17.07	23.58%
6144	17.82	12.37	30.58%	16.38	8.08%
평균 감소율		27.38%		5.54%	

표 3. 세 가지 설계들의 소요 면적 비교
(단위 slice 개수)

Table 3. Comparison of needed area.
(units: no. of slices)

일반적인 설계 (수식 (16))	제안한 설계 (고정 K)	제안한 설계 (가변 K)
144	107	204

표 4. 고정 K 설계시 필요한 ROM의 크기
(width x words)

Table 4. ROM requirement for fixed K .
(width x words)

K	T	ROM 크기
40	2	4 x 2
256	4	8 x 3
768	8	9 x 5
1536	16	11 x 9
2368	8	11 x 5
3136	56	12 x 29
3456	18	12 x 10
4480	8	12 x 5
5184	54	13 x 28
5248	32	13 x 17
6144	32	13 x 17

였다.

표 3에서는 각 설계에서 사용된 슬라이스들의 개수를 보인다. 고정 K 값에 대한 설계 결과는 메모리의 크기가 서로 다르므로 각각의 K 에 대해 사용한 슬라이스의 평균 개수를 보여 준다. 고정된 K 값의 경우 일반적인 설계보다 사용한 슬라이스의 개수가 더 작은 반면, 가변 K 의 경우에는 $f_2x^2\%K$ 생성 로직으로 인하여 일반적인 설계보다 많은 수의 슬라이스가 필요하다.

표 4에서는 각 K 에 대하여 필요한 ROM의 크기를 보여준다. 표에서 보이는 바와 같이 ROM에 저장되는 값의 최대크기는 13비트이고 최대 개수는 29 개 이다. 각 K 에 해당하는 $f_2x^2\%K$ 값의 범위에 맞게 ROM의 크기를 설정하고 각 K 에 해당하는 주기에 맞게 카운터를 설계함으로써 면적 면에서나 전력 소모 면에서 더욱 높은 효과를 기대할 수 있었다.

V. 결론

본 논문에서는 터보 코드를 위한 2차 다항식 순열 (QPP) 인터리브 주소 발생기를 구현하는데 있어서 전

력 소모량을 줄일 수 있는 설계 방법을 제안하였다. QPP 순열 $f(x) = (f_1x + f_2x^2) \% K$ 에서 $f_2x^2 \% K$ 가 짧은 주기를 갖고 한 주기 내에서 좌우 대칭이라는 사실을 보이고 이를 이용하여 QPP 주소 발생기를 설계하였다. 고정 K 에 대하여 $f_2x^2 \% K$ 의 값을 ROM에 저장하여 주소 생성에 사용하는 방법과 가변 K 에 대하여 처음 반주기 동안 $f_2x^2 \% K$ 의 값을 메모리를 저장하여 주소 생성에 사용하는 방법을 제안하였다. 가변 K 에 대한 설계방법은 면적이 증가하였지만 평균적으로 다양한 K 에 대하여 일반적인 설계방법보다 더 적은 전력을 소모하는 효과를 보인다. 고정 K 에 대한 설계방법은 면적 면에서도 일반적인 설계보다 적은 면적을 차지하고 실험한 모든 K 에 대하여 우수한 전력 소모 절감 효과를 보인다.

참고 문헌

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: turbo-codes," in Proc. of *IEEE ICC'93*, pp. 1064-1070, Geneva, Switzerland, May 1993.
- [2] A. Giuletti, L. van der Perre, and M. Strum, "Parallel turbo coding interleavers: Avoiding collisions in accesses to storage elements," *Elec. Lett.*, Vol. 38, no. 5, pp. 232-234, Feb. 2002.
- [3] 3GPP TS 25.212 v6, "Multiplexing and Channel Coding (FDD) (Release 6)."
- [4] 3GPP TSG RAN WG1-47, "Parallel decoding method for the current 3GPP turbo interleaver," Riga, Latvia, Nov. 2006.
- [5] Z. Wang, Y. Tang, and Y. Wang, "Low hardware complexity parallel turbo decoder architecture," in Proc. *IEEE ISCAS'03*, pp. 53-56, Bangkok, Thailand, May 2003.
- [6] J. Kwak and K. Lee, "Design of dividable interleaver for parallel decoding in turbo codes," *Elec. Lett.*, Vol. 38, no. 22, pp. 1362-1364, Oct. 2002.
- [7] J. Kwak, S. M. Park, S. S. Yoon and K. Lee, "Implementation of a parallel turbo decoder with dividable interleaver," in Proc. of *ISCAS'03*, pp. 25-28, May 2003.
- [8] J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, Vol. 51, no. 1, pp. 101-119, Jan. 2005.
- [9] O. Y. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, Vol. 52, no. 3, pp. 1249-1253, Mar. 2006.
- [10] J. Ryu and O. Y. Takeshita, "On quadratic inverses for quadratic permutation polynomials over integer rings," *IEEE Trans. Inform. Theory*, Vol. 52, no. 3, pp. 1254-1260, Mar. 2006.
- [11] 3GPP TSG RAN WG1-47, "Performance comparison of LTE turbo internal interleaver proposals," Sorrento, Italy, Jan. 2006.
- [12] 3GPP TSG RAN WG1-48, "QPP Interleaver Parameters," St. Louis, USA, Feb. 2007.
- [13] M. Cheng, B. Moision, J. Hamkins, and M. Nakashima, "An Interleaver Implementation for the Serially Concatenated Pulse-Position Modulation Decoder," in Proc. of *ISCAS'06*, pp.4244-4247, Island of Kos, Greece, May 2006.

저자 소개



이원호(정회원)
2002년 서강대학교 컴퓨터공학과
학사 졸업.
2008년 서강대학교 컴퓨터공학과
석사 졸업.
2008년 현재 (주)코닉시스템
통신기술연구소 연구원.

<주관심분야 : 통신 시스템, VLSI설계, ASIC 설계>

임종석(평생회원)
대한전자공학회 논문지
제44권 SD편 제12호 참조