

EPRML 읽기 채널용 면적 효율적인 저전력 폴딩 비터비 검출기의 구현

정회원 기훈재*, 김성남**, 안현주***, 김수원****

Area Efficient and Low Power Folding Viterbi Detector for EPRML Read Channels Application

Hoon-Jae Ki*, Sung-Nam Kim**, Hyun-Joo An***, Soo-Won Kim**** *Regular Members*

요 약

본 논문에서는 비터비 검출기의 복잡도와 전력소모를 감소시킬 수 있는 폴딩 비터비 검출기를 제안하였다. 제안된 폴딩 비터비 검출기는 상태 전이도가 대칭적인 것을 이용하여 상태는 서로 반전된 값을 갖는 것끼리 묶여지며, 확률거리의 경우 서로 부호가 반대인 값끼리 묶여진다. 제안된 폴딩 비터비 검출기를 EPRML 읽기 채널에 적용할 경우 확률거리 계산에 필요한 두 개의 가산기를 하나의 가감산기로 대체하여 기존의 GVA 알고리즘에 비해 하드웨어 복잡도를 37.4 % 감소시킬 수 있었다. 또한 불필요한 전력소모의 원인이 되는 글리치 발생을 신호 재배치와 병렬 구조와 같은 상위 수준의 저전력 기법을 적용하여 억제한 결과 12.7 %의 전력소모 감소를 나타내었다.

ABSTRACT

This paper proposes a new method, FVD (Folding Viterbi Detector), that has state transition diagram folded with inverted states. Most trellis coded signals have a symmetrical transition diagram, which means if it is folded exactly in half, branch metrics are folded together with those with inverted sign ones. Since one adder/subtractor substitutes for two adders where each branch requires one, the number of adders can be reduced. The FVD is applied to the EPRML read channels and compared with the GVA (Generalized Viterbi Algorithm) detector which requires eight adders. The FVD requires five adder/subtractors so the total gate count is reduced by 37.4 %. And by applying the signal reordering and parallel architecture due to the the unwanted glitches, the 12.7 % of the power can be saved in the Viterbi detector compared to the conventional architecture.

I. 서론

디지털 정보 시스템이 고속, 대용량화함에 따라 빠른 접근 속도와 높은 기록 밀도를 갖는 정보 저장 매체가 요구되고 있다. 디지털 자기 기록 시스템은 높은 기록 밀도의 특성을 가지고 있으며, 많은 정보를 적은 비용으로 저장할 수 있다. 특히, 자기 매체의 품질 향상 및 자기 헤드와 같은 기계적인

기술의 발전과 더불어 신호 처리 기술의 진보로 인해 자기 기록 시스템의 특성은 비약적으로 향상되고 있다. 그러나 자기 매체에 좀 더 많은 정보를 기록하기 위해서는 자기 매체나 헤드 기술보다 기록 밀도의 개선 효과가 큰 신호처리 방식의 개선은 필수적이다.

대표적인 디지털 자기 기록 시스템인 디스크 드라이브 시스템의 신호 처리 방식은 그동안 많은 발

* 하이닉스 반도체 System IC SBU LDI 개발 2팀(hoonjae.ki@hynix.com),

** 한국전자통신연구원 컴퓨터 시스템 연구부 병렬시스템 연구실(ksn@etri.re.kr),

*** 하이닉스 반도체 System IC SBU IT설계팀(anhj@sysic.hynix.com),

**** 고려대학교 전자공학과 (ksw@asic.korea.ac.kr)

논문번호 : 00433-1107, 접수일자 : 2000년 11월 8일

* 이 연구는 반도체 설계 교육 센터(IDEC)로부터의 부분적인 지원을 받아 이루어졌음.

전이 있었다. 주로 사용되었던 피크 검출 (peak detection) 방식은 신호의 피크 값만을 검출하는 방식으로 기록 밀도를 높이면 헤드에서 읽은 신호가 서로 부호간 간섭을 일으켜 SNR 및 진폭의 감소로 인하여 데이터를 검출하지 못하므로 기록밀도를 높이는 데 한계를 보이고 있다. 이러한 기록밀도를 향상시키기 위해 제안된 PR (Partial Response) 방식은 기록밀도를 높일 때 생기는 부호간 간섭을 일정 정도 허용하여 신호를 처리하는 방식으로 디스크 드라이브 시스템은 PR-IV 계열을 주로 사용한다. 지연연산자 D를 이용하여 $(1-D)(1+D)^n$ 형태의 PR-IV 계열 신호를 표현하면 n이 1, 2, 3일 경우 각각 PR-IV, EPR-IV (Extended PR-IV), EEPR-IV 신호라 한다. 차수가 높아짐에 따라 기록밀도를 높일 수 있으며 허용되는 부호간 간섭이 커지므로 신호를 검출하는 검출 시스템의 복잡도는 증가한다. 근래에는 고밀도의 디스크 드라이브 시스템이 출시되면서 대부분의 시스템이 EPR-IV 신호 방식을 채용하고 있다^[1].

이러한 PR-IV 계열의 신호 처리 방식으로 기록된 신호를 재생하기 위해서는 MLSD (Maximum Likelihood Sequence Detection) 방식을 사용한다. PR-IV 신호 처리방식과 MLSD 검출 방식이 결합된 것을 PRML (Partial Response Maximum Likelihood) 읽기 채널 시스템이라고 한다. MLSD 방식의 대표적인 방법인 비터비 알고리즘은 모든 가능한 경로에 대해 확률거리를 계산하여 가장 가능성 있는 경로만을 남겨 신호를 검출하는 방식으로, 대부분의 트렐리스 코드로 인코딩된 신호 방식에서 검출 방법으로 사용하고 있다^[2, 3].

일반적으로 EPRML 시스템의 경우 8 상태의 상태 천이도를 이용하여 신호를 검출한다. 그러나, 각 상태마다 상태값을 저장하고 이에 따른 확률 거리를 계산하기 위하여 큰 하드웨어가 요구되는 단점이 있다. 이러한 하드웨어 복잡도를 감소시키기 위해 약간의 기록밀도를 감소시키면서 상태 수를 줄이는 방식이 많이 채택되어왔으며 근래에는 기록밀도를 낮추지 않으면서 상태 천이도의 유사성을 이용하여 하드웨어의 복잡도를 감소시키는 방향으로 연구가 진행중이다^[4].

본 논문에서는 비터비 검출기의 복잡도를 줄이기 위해 폴딩 비터비 검출기 (Folding Viterbi Detector, FVD)를 제안하였다. 제안된 검출기는 확률거리를 저장하는 레지스터와 확률거리 계산에 필요한 하드웨어를 감소시켰으며 저전력 기법을 채용하여

전력을 감소시켰다.

II. EPRML 비터비 검출기⁽³⁾

EPRML 읽기 채널에서는 비터비 알고리즘을 이용하여 각 상태로 확장되기 위한 확률거리를 계산하고 가장 가능성 있는 경로만 남겨 데이터를 검출한다. 디지털 데이터 열을 $\{ak\}$, 비터비 검출기 입력 샘플 신호를 $\{yk\}$, 각 상태로 확장될 때의 확률 거리를 E라 하면 식 (1)과 같이 확률거리를 구할 수 있다.

$$E = \sum_{k=0}^{\infty} [y_k - (a_k + a_{k-1} - a_{k-2} - a_{k-3})] \quad (1)$$

여기서 비교할 값 $(a_k + a_{k-1} - a_{k-2} - a_{k-3})$ 를 c_k 라 정의하고 누적되는 상수항 y^2_k 를 제거하면 식 (2)와 같이 표현할 수 있다.

$$E = \sum_{k=0}^{\infty} y^2_k = \sum_{k=0}^{\infty} [(y_k - c_k)^2 - y^2_k] \quad (2)$$

발생할 수 있는 모든 경로에 대하여 식 (2)를 이용하여 확률거리를 구하고 그 중에서 가장 작은 신호열을 얻어내 데이터열을 검출한다. EPR-IV의 경우 표 1과 같이 다섯 가지의 비교값에 대해 확률 거리를 구할 수 있다. EPR-IV 신호는 식 (3)과 같이 연속된 3개의 자화 정보 기록을 추적해야하므로 총 8개의 상태가 존재한다. 즉, EPR-IV에서는 '000', '001', '010', '011', '110', '101', '110', '111'과 같은 8개의 상태를 가진다.

표 1. EPR-IV 입력신호의 확률거리

c_k	$E_k = (y_k - c_k)^2$	$E_k y^2_k$
0	y^2_k	
-1	$y^2_k + 2y_k + 1$	$2y_k + 1$
+1	$y^2_k - 2y_k + 1$	$-2y_k + 1$
-2	$y^2_k + 4y_k + 4$	$4y_k + 4$
+2	$y^2_k - 4y_k + 4$	$-4y_k + 4$

$$(1-D)(1+D)^2 = 1 + D - D^2 - D^3 \quad (3)$$

식 (3)을 이용하면 그림 1과 같은 상태 천이도와 트렐리스를 나타낼 수 있다. EPR-IV 신호는 각 상

태에서 2개의 경로가 존재하므로 8상태 2진 트렐리스 형태이다. 비터비 검출기는 확률거리를 연산하는 ACS (Add, Compare and Select) 회로와 트렐리스와 동일한 구조를 가지고 각 상태값을 저장하는 PM (Path Memory) 회로로 구성되어진다. 즉, ACS 회로에서 확률거리를 연산한 뒤 판단 결과를 이용하여 PM 회로의 상태를 확장시켜 신호를 검출한다.

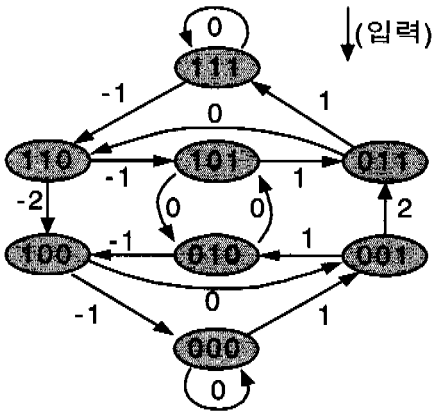
그림 2에 EPR-IV 트렐리스의 일부에 대한 ACS와 PM 회로를 나타내었다. 그림 2(a)에 나타낸 '000' 상태로의 천이는 '000' 상태에서 0을 더하는 경우와 '100' 상태에서 -1을 더하는 경우가 존재한다. 마찬가지로 '001' 상태로의 천이는 '000' 상태에서 '1'을 더하는 경우와 '100' 상태에서 '0'을 더하는 경우가 존재한다. 표 1을 참조하면 계산할 확률 거리는 -1의 경우 $2y_k+1$, +1의 경우 $-2y_k+1$ 이 된다. 그림 2(b)의 ACS 회로는 shifter를 통해서 $2y_k$ 의 계산을 수행하며 각각의 상태를 고려하여 $2y_k+1$

과 $-2y_k+1$ 를 계산한다. 이렇게 계산된 확률 거리는 각각의 비교기에 입력되어 확률거리를 비교하게 된다. 비교기의 출력은 MUX의 선택 신호로 인가되어 두 가지 확률 거리 중에서 작은 값을 선택하여 상태 레지스터인 FF (Flip Flop)에 저장된다.

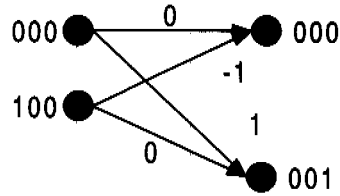
결과적으로 그림 2의 트렐리스 일부를 위해 ACS 회로는 FF, shifter, 증분기, 가산기, 비교기, MUX 등으로 구성되며 확률거리 계산에 필요한 비트 수 만큼의 하드웨어가 필요하게 된다. PM 회로의 경우는 그림 2(c)와 같은 PM 셀을 여러 신호들을 추적할 수 있도록 직렬로 연결하여 확장한다. 또한 이렇게 직렬로 확장된 셀들이 상태 수만큼 필요하게 된다.

III. GVA (Generalized Viterbi Algorithm) 알고리즘^[4]

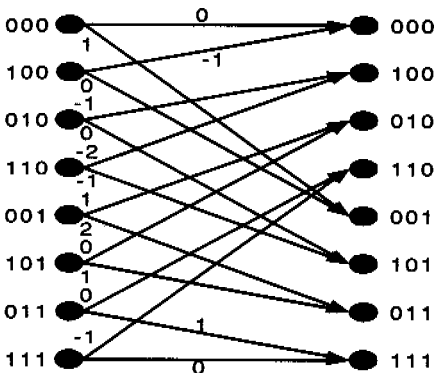
II장에서 설명한 바와 같이 EPR-IV 방식에서 비터비 검출기의 하드웨어 부담이 크기 때문에 하드웨어의 복잡도를 감소시키는 방법들이 연구되고 있다.



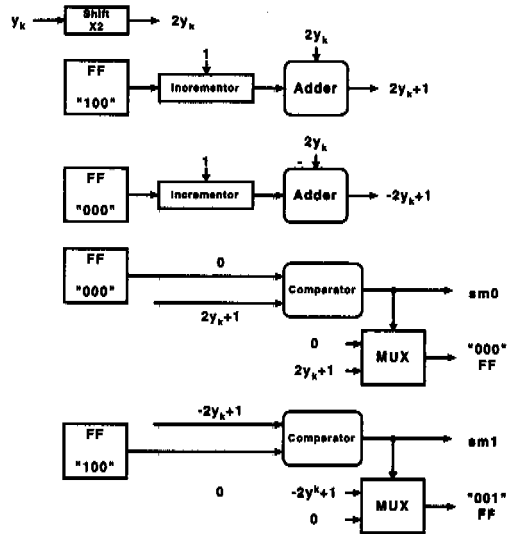
(a) 상태 천이도



(a) 트렐리스 일부

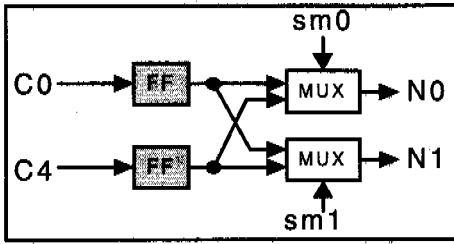


(b) 트렐리스



(b) ACS 회로

그림 1. EPR-IV 상태천이도와 트렐리스



(c) PM 회로

그림 2. EPR-IV 비터비 검출기 구현의 예

그 중 널리 알려진 알고리즘이 GVA이다. 일반적으로 GVA는 L 개의 집합으로 묶어서 표현할 수 있는 트렐리스에서 S 개의 경로를 선택하는 방법이다. 이를 (L, S) GVA로 표현하기도 한다. GVA는 트렐리스 코드 신호가 항상 모든 상태가 동시에 중요하지 않기 때문에 상태 수를 줄여서 표현하는 방식으로 BER 성능이 크게 떨어지지 않기 때문에 일부 상태를 줄여서 사용한다. 일반적으로 EPRML에 사용하는 8 상태를 4 상태로 줄인 (4,1) GVA의 EPR-IV 트렐리스를 그림 3에 나타내었다. EPR-IV 신호 트렐리스 중에서 8 개의 모든 상태가 동시에 중요하지 않기 때문에 가장 이전 상태값을 x (don't care)로 정의하여 두 상태 중 하나만의 확률거리만을 보존하는 방식이다. 상태 수가 절반으로 줄어들기 때문에 확률거리를 저장하는 필요한 ACS와 PM 회로의 FF를 절반으로 줄일 수 있다.

이러한 (4,1) GVA를 이용하여 EPR-IV 신호에 대한 비터비 검출기를 구성할 경우, 두 상태를 구분하여 확률거리를 누적시켜야 하기 때문에 상태를 구분할 수 있는 β 이라는 신호를 정의해야 한다. 즉, 'x00' 상태를 '000'와 '100' 상태로 구분하기 위해 가장 이전 상태값을 β_i 에 저장한다. β_i 가 '1'일 경우 'x00' 상태를 '100' 상태로, '0'일 경우 '000' 상태로 판단하여 서로 다른 확률 거리를 계산, 저장한다.

그림 3의 트렐리스 중 {'x11', 'x01'} 상태에서 {'x11', 'x10'} 상태로 확장되는 트렐리스를 구현한 ACS 회로를 그림 4에 나타내었다. 그림 4(b)에서 shifter를 이용하여 각각 $2y_k$, $4y_k$ 를 계산하고, 'x11'에서 'x11'로의 확률거리 계산은 각각 0과 1이므로 0과 $-2y_k+1$ 을 계산한다. 'x01'에서 'x11'로의 확률거리 계산은 각각 1과 2이므로 $-2y_k+1$ 과 $-4y_k+4$ 를 계산한다. 계산된 확률거리는 β_{x01} 과 β_{x11} 에 의해서 선택되어지고 비교기를 통해서 확률거리를 비교한

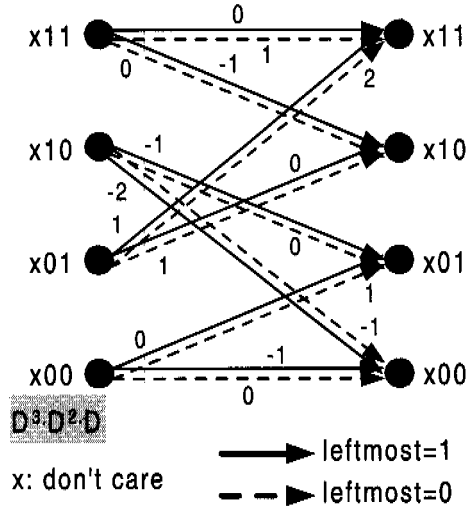
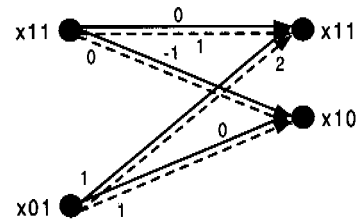
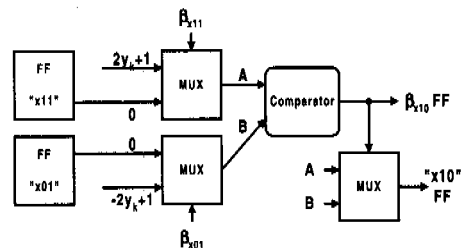
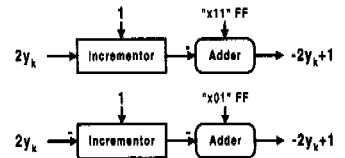


그림 3. (4,1) GVA EPR-IV 트렐리스

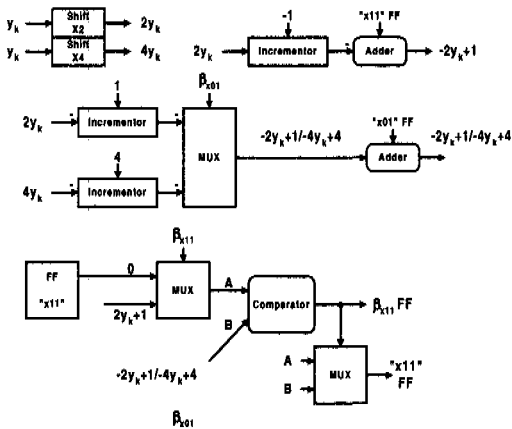
후 MUX를 통해 선택되어진다. 그림 4(c)는 'x10' 상태로의 확률거리를 계산하는 ACS 회로이며 'x11' 상태와 유사하게 계산된다. 비교기의 출력값은 β 레지스터에 저장되며 PM 회로에서 트렐리스를 확장시키기 위한 입력 신호로 사용된다. PM 회로는



(a) 트렐리스 일부



(b) x11 상태



(c) x10 상태

그림 4. (4,1) GVA EPR-IV 비터비 검출기의 ACS 회로

EPR-IV 비터비 검출기와 같은 구조이며 단지 상태 수가 절반으로 감소되었기 때문에 구현에 필요한 PM 셀의 수도 절반으로 감소한다.

IV. 풀딩 비터비 검출기 (FVD)

GVA 알고리즘을 이용한 비터비 검출기의 경우, 하드웨어의 복잡도를 감소시키기 위하여 두 상태를 하나의 상태로 묶어서 확률거리 저장에 필요한 FF 수를 줄일 수 있었다. 하지만 신호 검출을 위한 확률거리 계산에 필요한 ACS 회로의 경우, 일반적인 EPR-IV 비터비 검출기의 경우와 같기 때문에 여전히 복잡한 하드웨어 구성을 갖는다.

본 논문에서는 (4,1) GVA 알고리즘을 바탕으로 ACS 회로의 복잡도를 줄이기 위해 풀딩 비터비 검

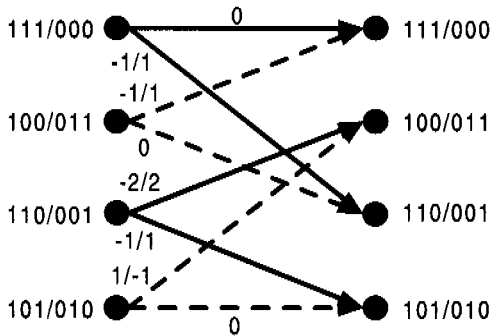
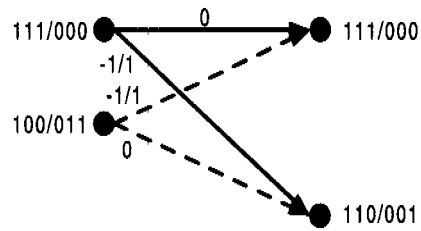


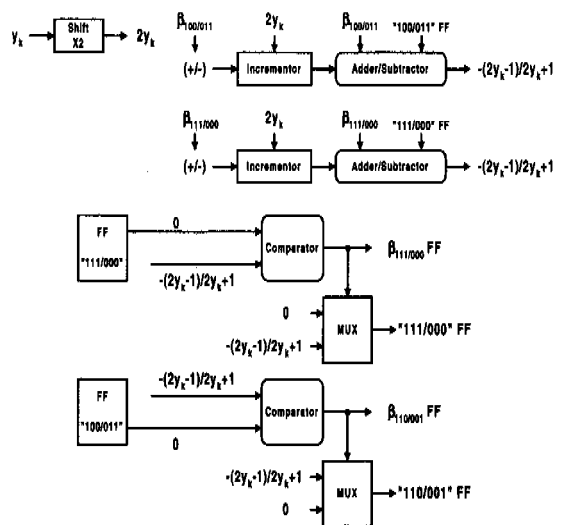
그림 5. FVD의 EPR-IV 트렐리스

출기를 제안하였다. EPR-IV 상태천이도가 대칭적인 구조를 갖는 점에 착안하여 상태천이도를 반으로 접은 형태를 사용한다. GVA와 마찬가지로 트렐리스 코드 신호의 모든 상태가 똑같이 중요하지 않기 때문에 FVD에서도 GVA와 마찬가지로 두 개의 상태를 묶어서 그 중 가장 중요한 상태만을 선택해 신호를 검출할 수 있다. 즉, FVD는 상태 수를 절반으로 줄일 때 GVA에서와는 달리 상태가 서로 반전된 값을 갖는 것끼리 묶어지며, 확률거리의 경우 서로 부호가 반대인 값끼리 묶여진다. 서로 부호가 반대인 확률거리를 묶을 경우, GVA와는 달리 두 개의 가산기들을 하나의 가감산기로 대체할 수 있으므로 하드웨어의 복잡도를 감소시킬 수 있다^[5].

그림 5에 EPR-IV 신호에 대한 FVD의 트렐리스를 나타내었다. GVA에서는 '000' 상태와 '100' 상태를 'x00' 상태로 묶은 반면, FVD에서는 '000'와 '111' 상태를 묶어 '111/000' 상태로 만든다. 이렇



(a) 트렐리스

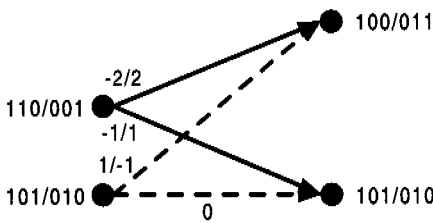


(b) ACS

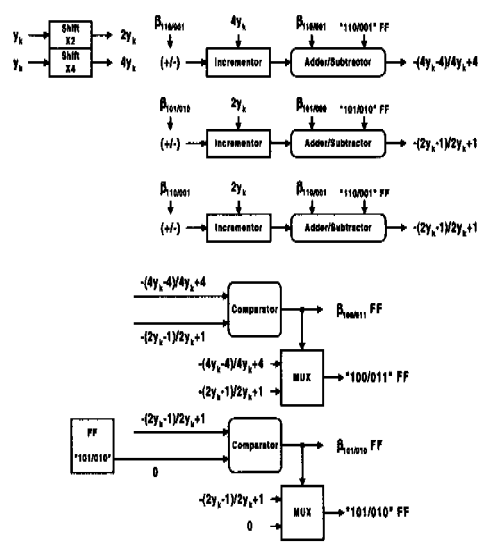
그림 6. '111/000', '100/011' 상태에 대한 ACS 회로

계 나머지 상태들에 대해서도 FVD를 적용하면 확률거리가 서로 부호가 반대인 신호들끼리 묶여진다. '111' 상태와 '000' 상태를 구분하기 위하여 GVA와 마찬가지로 β_i 라는 상태를 구분하는 신호를 사용한다. 확률거리 계산에 대해 구체적으로 설명하면 다음과 같다. 그림 5에서 확률거리값이 $-1/1$ 인 경우에 대해 계산할 확률거리는 $-2y_k+1/2y_k+1$ 이다. 여기서 β_i 가 '1'인 경우 $-2y_k+1$, β_i 가 '0'인 경우 $2y_k+1$ 를 계산하는데, $-2y_k+1/2y_k+1$ 은 $-(2y_k-1)/2y_k+1$ 형태로 바꾸어 처리한다. 그 이유는 입력 신호와 누적된 확률거리의 경우 누적된 확률거리 계산에 필요한 비트수가 더 많기 때문에 +1을 직접 확률거리에 가산할 경우 증분기가 처리해야할 비트 수가 증가하기 때문이다. 따라서 입력 샘플 신호 y_k 를 1 비트 이동하여 곱하기 2를 처리하고 -1 또는 1을 더해 $-(2y_k-1)/2y_k+1$ 을 계산한다. 이 값이 계산된 후 선택 신호에 따라 둘 중 하나를 누적시킨 새로운 확률거리를 계산한다. 여기서 감산과 가산이 동시에 이루어지지 않으므로 어느 한 연산만을 수행하므로 가감산기를 이용할 수 있다⁶⁾.

그림 6과 7에 FVD의 ACS 회로를 나타내었다. 그림 6은 '111/000' 상태와 '100/011' 상태로의 확률거리 계산에 대한 회로이다. 그림 6(a)에 나타낸 트렐리스에 대한 확률거리 계산은 0과 $-1/1$ 이 필요하다. $-1/1$ 의 경우, $-(2y_k-1)/2y_k+1$ 에 대한 계산이 필요하며 β 에 따라 증분기와 가감산기의 동작을 선택하게 된다. 즉, β 가 1인 경우 $-(2y_k-1)$ 을 계산하기 위하여 -1을 증가시켜 감산의 동작을 수행하고, β 가 0인 경우 $2y_k+1$ 을 계산하기 위하여 1을 증가시켜 가산의 동작을 수행한다. GVA와 마찬가지로 계산된 확률거리는 각각 비교기와 MUX에 인가되고 작은 확률거리를 갖는 값이 선택되어 레지스터에 저장되며 비교기 출력인 β 는 PM 회로에서 PM 셀을 확장시키는데 사용된다. 그림 7의 경우는 '100/011' 상태와 '101/010' 상태로의 확률거리 계산에 필요한 ACS회로이다.



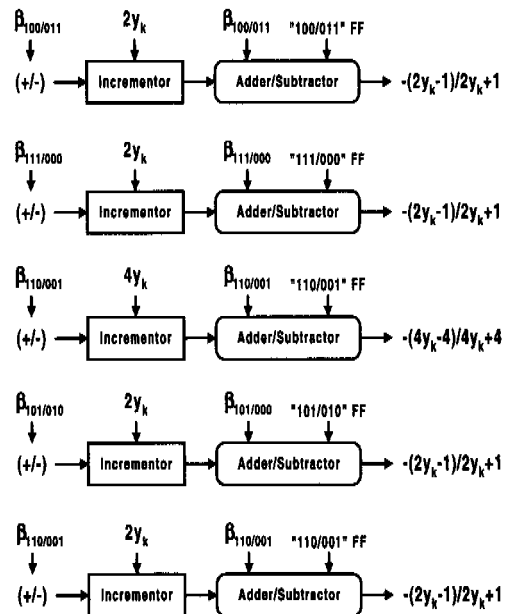
(a) 트렐리스



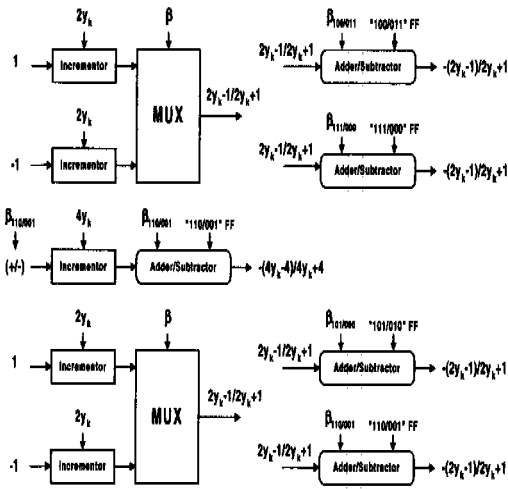
(b) ACS

그림 7. '100/011', '101/010' 상태에 대한 ACS 회로

FVD의 ACS 회로를 구현할 경우, 확률거리를 계산시 1의 증가 또는 감소와 가산 또는 감산의 선택을 β 신호의 값에 따라 결정한다. 이러한 계산 구조는 선택 신호의 천이에 따라 증분기와 가감산기의 출력에서 많은 글리치를 발생시키며 불필요한 전력소모의 원인이 된다⁶⁾. 본 논문에서는 이러한



(a) 신호 재배치와 병렬 구조를 적용하지 않은 경우



(b) 신호 재배치와 병렬구조를 적용한 경우

그림 8. 계산 과정 비교

글리치의 발생을 억제하기 위하여 선택 신호 β 를 재배치하고, β 신호의 값에 따라 미리 증가 또는 감소된 값을 계산해 놓는 병렬 구조를 선택하여 전력소모를 개선하였다. 신호 재배치와 병렬 구조의 유무에 따른 확률거리 계산 과정을 그림 8에 나타내었다.

표 2. ACS 회로의 게이트 수 비교

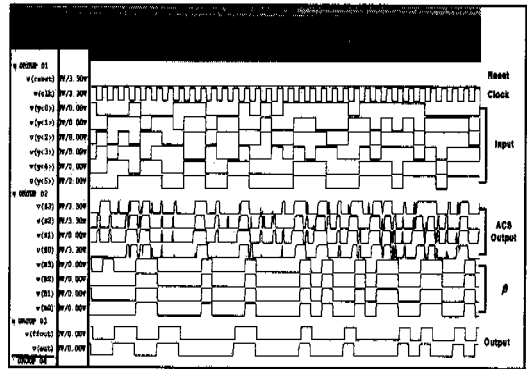
	GVA	FVD
9 비트 증분기 (35 gate)	10	5
	$10 \times 35 = 350$	$5 \times 35 = 175$
9 비트 가산기 (73 gate)	8	5
9 비트 가감산기 (82 gate)	$8 \times 73 = 584$	$5 \times 82 = 410$
Total Gate Count	934	585

V. 모의 실험 및 성능 평가

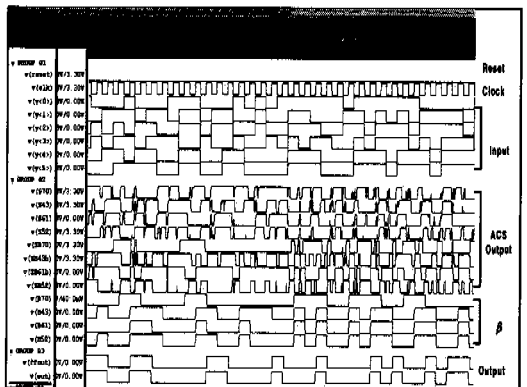
GVA 비터비 검출기와 FVD의 ACS 회로에 대한 비교를 표 2에 정리하였다. 사용된 9비트 증분기의 경우 35 게이트, 9 비트 가산기는 73 게이트, 그리고 9 비트 가감산기는 82 게이트로 구성되며 FVD를 적용할 경우 37.4 %의 복잡도 감소 효과를 나타낸다.

GVA, FVD와 신호 재배치와 병렬 구조를 적용한 저전력 FVD의 전력 소모 비교를 위하여 0.6

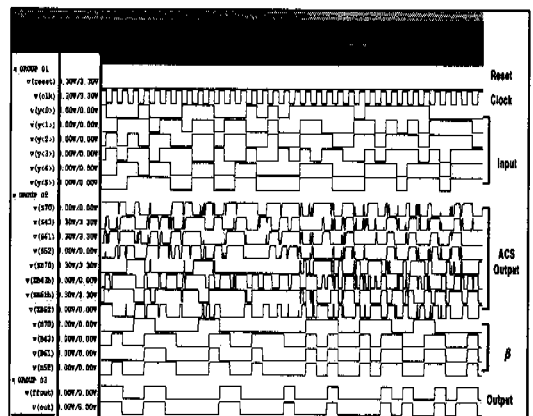
um 1P3M CMOS 공정과 PowerMill을 이용하여 모의실험을 수행한 결과를 그림 9에 나타내었다. 2000개의 입력 샘플을 똑같이 인가하여 전원전압 3.3 V, 100 MHz에서 소비 전력을 조사하여 표 3



(a) GVA



(b) FVD



(c) 저전력 FVD

그림 9. 모의 실험 결과

에 정리하였다. FVD의 경우 GVA에 비해 4.7 %의 전력소모 감소를 나타내었으며 이는 37.4 %의 게이트 감소에 비해 매우 적은 수치임을 알 수 있다. 이러한 FVD의 전력소모 원인은 IV장에서 설명한 바와 같이 증감과 가감산을 선택하는 신호의 빈번한 천이 때문에 발생하는 글리치 때문이다. 선택 신호는 클럭당 평균 47.4 %의 상태 천이가 발생하였으며 이러한 신호의 스위칭 동작으로 인하여 불필요한 전력을 소비하고 있음을 알 수 있다. 이에 비해 신호 재배치와 병렬 구조를 적용한 저전력 FVD의 경우 글리치 감소로 인하여 GVA에 비해 12.7 % 감소된 전력을 소모하였다.

표 3. 전력 소모 비교

	GVA	FVD	저전력 FVD
Supply Voltage	3.3 V		
Technology	0.6 um CMOS Technology		
Power Consumption (mW/MHz)	4.09	3.90	3.57

VI. 결론

본 논문에서는 EPRML 시스템에 사용되는 비터비 검출기의 복잡도와 전력소모를 감소시킬 수 있는 폴딩 비터비 검출기를 제안하였다. 제안된 폴딩 비터비 검출기는 EPRML 읽기 채널의 상태 천이도가 대칭적인 것을 이용하여 상태 레지스터와 신호 검출에 필요한 레지스터의 수를 절반으로 줄일 수 있으며, 확률거리 계산을 공유함으로써 확률거리 계산에 필요한 ACS 회로의 복잡도를 기존의 GVA 알고리즘에 비해 크게 감소시켰다. 또한 신호 재배치와 병렬구조를 적용하여 확률거리 계산시 발생하는 불필요한 글리치를 억제시킴으로써 전력소모를 감소시킬 수 있었다.

제안된 폴딩 비터비 검출기는 확률거리 계산을 공유함으로써 GVA에서 필요한 10 개의 증분기와 8 개의 가산기를 5 개의 증분기와 5 개의 가감산기로 대체할 수 있었다. 이에 따른 하드웨어 복잡도를 비교한 결과, 확률거리 계산에 필요한 하드웨어의 복잡도를 37.4 %만큼 감소시킬 수 있었다. 또한 신호 재배치와 병렬 구조 같은 상위 수준의 저전력 기법을 적용하여 전력감소를 비교한 결과, 3.3 V 전원전압에서 FVD는 GVA에 비해 12.7 %의 전력

소모 감소를 나타내었다. 결론적으로 모의실험과 복잡도 비교를 통하여 제안된 FVD가 복잡도 감소와 저전력 소모에 효율적인 알고리즘임을 증명하였다.

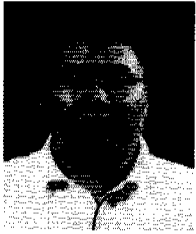
현재, 기록 밀도의 증가에 대한 요구로 인하여 PRML 시스템에서 EPRML 시스템으로 자기 기록 시스템이 급격히 이동하고 있는데 본 논문에서 제안된 폴딩 비터비 검출기가 자기 저장 시스템의 복잡도 감소와 저전력 소모에 큰 기여를 할 것으로 기대된다.

참고 문헌

- [1] H. Kobayashi, and D.T.Tang, "Application of Partial-Response Channel Coding to Magnetic Recording System", *IBM Journal of Research and Development*, pp. 368-375, July 1970
- [2] G. D. Froyen Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequence in the Presence of Intersymbol Interference", *IEEE Transaction of Information Information Theory*, Vol. IT-18, No. 3, pp. 363-378, May 1972
- [3] R. W. Wood and D. A. Peterson, "Viterbi Detection of Class IV Partial Response on Magnetic Recording Channel", *IEEE Transaction on Communication*, Vol. COM-34, No. 5, pp. 3666-3668, Sept. 1987
- [4] T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm", *IEEE Transaction of Information Information Theory*, Vol. IT-33, No. 6, pp. 866-876, Nov. 1987
- [5] H. J. Ki, C. S. Lee, M. H. Song, and S. W. Kim, "Folding Viterbi Detector with Reduced Complexity for EPRML System Applications", *IEE Electronics Letters*, Vol. 36, No. 4, pp. 330-331, Feb. 2000.
- [6] A. P. Chandrakasan, S. Sheng, and R. W. Broderson, "Low-Power CMOS Digital Design", *IEEE J. Solid State Circuits*, Vol. 27, No. 4, pp. 473-483, April 1992

기 훈 재(Hoon-Jae Ki)

정회원



1995년 2월 : 고려대학교
전자공학과 졸업
1997년 2월 : 고려대학교
전자공학과 석사
2001년 2월 : 고려대학교
전자공학과 박사

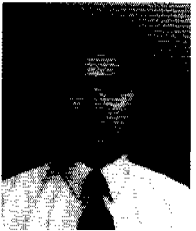
1987년 3월~현재 : 고려대학교 전자공학과 교수

<주관심 분야> 아날로그/디지털 VLSI 설계 기술
개발, 이동통신용 부품 설계, 디지털 신호
처리

2001년 2월~현재 : 하이닉스 반도체 LDI 개발 2팀
<주관심 분야> 저전력 VLSI 설계, PRML 읽기 채널 IC, LCD Driver IC

김 성 남(Sung-Nam Kim)

정회원



1991년 2월 : 고려대학교
전자공학과 졸업
1993년 2월 : 고려대학교
전자공학과 석사
1998년 8월 : 고려대학교
전자공학과 박사

1999년 3월~현재 : ETRI 병렬 시스템 연구팀
<주관심 분야> VLSI 설계, 고성능 연결망 칩셋 설계, 클러스터 컴퓨터 시스템 등

안 현 주(Hyun-Joo An)

정회원



1999년 2월 : 고려대학교
전자공학과 졸업
2001년 2월 : 고려대학교
전자공학과 석사
2001년 2월~현재 : 하이닉스
반도체 IT 설계팀

<주관심 분야> Image Sensor IC, 디지털 신호처리

김 수 원(Soo-Won Kim)

정회원



1974년 2월 : 고려대학교
전자공학과 졸업
1983년 2월 : TEXAS A&M
전자공학과 석사
1976년 2월 : TEXAS A&M
전자공학과 박사