ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
j.inst.Korean.electr.electron.eng.Vol.24,No.3,838~844,September 2020
논문번호 20-03-21
http://dx.doi.org/10.7471/ikeee.2020.24.3.838
162

블루투스 저전력 시스템을 위한 저복잡도 결합 비터비 검출 및 복호 알고리즘의 하드웨어 설계 및 구현

Hardware Design and Implementation of Joint Viterbi Detection and Decoding Algorithm for Bluetooth Low Energy Systems

박 철 현*, 정 용 철*, 정 윤 호*★

Chul-hyun Park*, Yongchul Jung*, Yunho Jung*★

Abstract

In this paper, we propose an efficient Viterbi processor using Joint Viterbi detection and decoding (JVDD) algorithm for a for bluetooth low energy (BLE) system. Since the convolutional coded Gaussian minimum-shift keying (GMSK) signal is specified in the BLE 5.0 standard, two Viterbi processors are needed for detection and decoding. However, the proposed JVDD scheme uses only one Viterbi processor by modifying the branch metric with inter-symbol interference information from GMSK modulation; therefore, the hardware complexity can be significantly reduced without performance degradation. Low-latency and low-complexity hardware architecture for the proposed JVDD algorithm was proposed, which makes Viterbi decoding completed within one clock cycle. Viterbi Processor RTL synthesis results on a GF55nm process show that the gate count is 12K and the memory unit and the initial latency is reduced by 33% compared to the modified state exchange (MSE).

요 약

본 논문에서는 검출과 복호가 결합된 효율적인 비터비 알고리즘 (joint Viterbi detection and decoding (JVDD))의 저복잡도 하드웨어 설계 및 구현 결과를 제시한다. 길쌈부호화된 GMSK 신호가 BLE 5.0 표준으로 채택 되어있으므로 검출과 복호를 위해 두개의 비터비 프로세서가 필요하다. 그러나, 제안된 JVDD 알고리즘은 GMSK에 의해서 유발된 심볼간의 간섭정보 (ISI : inter-symbol interference)가 반영된 가지 메트릭 (branch metric)을 사용하여 단지 하나의 비터비 만을 사용하여도 검출과 복호 수행이 가능하며, 성능 저하 없이 복잡도 감소가 가능하다. JVDD 알고리즘을 적용한 BLE 비터비 복호기의 하드웨어 구현을 위해 효율적인 구조 설계가 수행되었다. 제안된 구조는 1 클럭 사이클 동안 복호를 완료할 수 있기 때문에 저지연 및 저면적 구현이 가능하다. 제안된 비터비 복호기는 Verilog-HDL을 이용하여 RTL 설계되었고, GF 55nm 공정을 활용하여 논리합성 및 구현되었다. 합성결과 12K 게이트 수를 포함하였으며 메모리 유닛 및 초기 지연시간은 MSE (modified state exchange) 대비 33% 감소 가능함을 확인하였다.

Key words : BLE, ISI, JVDD, TraceBack, Viterbi

* School of Electronics and Information Engineering, Korea Aerospace University
★ Corresponding author
E-mail : yjung@kau.ac.kr, Tel : +82-2-300-0133
※ Acknowledgment
This work was supported by the Technology Innovation Program, 10080619, funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) and CAD toolswere supported by IDEC.
Manuscript received Sep. 7, 2020; revised Sep. 22, 2020; accepted Sep. 23, 2020.

Ⅰ. 서론

BLE 5.0은 전송속도 증가와 전송가능 거리의 증가, 그리고 적은 전력소모를 위해서 LE(low energy)-Uncoded 프로토콜에 2Mbps가 추가되었고, GMSK (Gaussian minimum-shift keying)와 길쌈부호가 결합된 LE-Coded 프로토콜에서 500Kbps와 125Kbps를 지원한다[1]. BLE 5.0은 이러한 특징으로 인해서 IoT(internet of thing) 응용에서 각광받고 있는 무선 통신 기술이다. GMSK가 적용된 신호는 스펙트럼 효율을 향상 시키면서 포락선이 일정하게 유지되는 특성이 있기 때문에 전송단에서 저전력 비선형 앰프를 사용할 수 있다. 이런 특성으로 인하여 전체적으로 전력소비가 감소할 수 있다. 또한, GMSK 신호에 길쌈부호기를 사용한다면 수신감도를 증가시킬 수 있어서 전송거리가 크게 향상되는 효과가 있다. 그러나, GMSK는 연속된 심볼에 걸쳐서 ISI(inter-symbol interference)를 유발하는 특성을 가지므로, 최적의 검출을 위한 방법은 최대 유사도 시퀀스 추정(MLSE)이 적용된 비터비 알고리즘을 사용하는 것이다[2].

비터비 알고리즘은 주로 길쌈부호를 복호하는데 많이 적용되는 방식이다. 검출과 복호를 할 때 각각 비터비 방식을 적용하게 되면 최적의 성능을 낼 수가 있다. 그러나, 최적의 성능을 위해 두 개의 비터비를 사용하는 경우에는 복호지연과 하드웨어 복잡도가 증가하게 된다. 이러한 문제를 해결하기 위해 [3-6]에서는 하나의 비터비 프로세서로 검출과 복호를 동시에 수행하여 하드웨어 복잡도를 감소하는 방법을 제시하고 있다. [6]에서 제시하는 JVDD (joint Viterbi detection and decoding)는 검출 과정에서 ISI 채널정보가 적용된 트렐리스 구조를 비터비 알고리즘으로 이용하고 복호 과정에서는 parity 체크를 적용하여 하나의 비터비 프로세서로 검출과 복호를 수행하는 방법이다. 그러나, [6]의 JVDD는 채널정보 트렐리스에서 비터비 알고리즘이 적용 될 때 충분히 많은 수의 생존경로를 이용하기 위하여 하나의 state에서 여러 개의 생존경로를 관리하여 메모리 사용량이 늘어난다. 즉 하드웨어 복잡도가 생존경로 수와 코드워드 길이에 비례해서 증가하게 되어 기존의 비터비 알고리즘에서 필요한 메모리 보다 큰 사이즈를 필요로 한다. [7]은 GMSK 변조에 의해 발생한 ISI 정보를 가지 메트릭 계산에 적용한 효율적인 JVDD방법을 제안하였다. [7]에서는 ISI 정보를 가지 메트릭 계산에 반영하여 새로운 가지 메트릭을 사용함으로써 일반적인 비터비 알고리즘과 같이 각각의 state에서 하나의 생존 경로만 남겨지도록 한다. 이런 이유로 [7]에서 제안된 JVDD는 [6]에 비해서 낮은 복잡도로 구현 가능하게 된다.

BLE 5.0에서는 패킷수신 이후에 응답 패킷을 보내는 시간으로 정의된 TIFS(inter frame space)를 150us로 정의하고 있고 또한 ±2us 오차를 벗어나지 않는 요구조건이 있다. 길쌈부호가 적용된 LE-Coded에서는 마지막 심볼이 수신된 이후 최대 2us내에 비터비 복호과정이 완료되어야 응답 패킷을 보낼 수 있는 조건을 만족할 수 있다. 그러므로, 수신단에서는 복호 지연이 최소화 되어야 한다.

비터비 복호과정에는 일반적으로 다음과 같은 두 가지 방식이 많이 사용된다. RE(register exchange)는 최적의 생존경로를 실시간으로 계산하여 복호지연이 없는 장점이 있기 때문에 마지막 심볼이 수신되면 복호가 끝난다[8]. 하지만, 너무 과도한 레지스터 스위칭 과정에 의해서 전력소모가 커지므로 저전력 시스템에는 적용을 하지 않는다. TB(trace back)은 역추적으로 복호를 하게 되어 복호지연이 존재하지만 전력소비가 적고 메모리 사용이 효율적인 구조로서 가장 많이 사용되는 방식이다[9]. 하지만, 일반적인 TB 구조를 사용하면서 BLE 5.0에서 요구하는 TIFS 조건을 충족하기 위해서는 동작 클럭 주파수를 증가시켜 지연시간을 줄이거나 또는 역추적 길이를 줄여서 복호지연 시간을 만족해야 한다. 그러나, 클럭 주파수를 높이면 전력 소모가 증가하게 되는 문제가 발생하고 역추적 길이를 줄이게 되면 성능 열화가 발생한다. 따라서, 본 논문에서는 [7]에서 제시된 JVDD방식을 적용하여 최적의 성능을 유지하면서 BLE 5.0에서 요구하는 TIFS를 만족하기 위해서 비터비 프로세서의 SMU (survivor management unit)이 개선된 비터비 프로세서의 구조를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 [7]에서 제시하는 JVDD 알고리즘에 대해서 설명하고 3장에서는 BLE 5.0에 적합한 JVDD 비터비 프로세서의 구조와 설계를 설명한다. 마지막으로 4장에서는 본 논문의 결론을 맺는다.

II. JVDD 알고리즘

1. JVDD 비터비 복호 알고리즘

수식 (1)에서 $\Delta\Phi(t)$는 수신 심볼의 위상차를 나타낸다. 위상차에는 ISI 정보와 채널 노이즈가 함께 포함되어 있다.

$$\Delta\Phi(t) = \arg(r(t)r(t+T)^{*}) \quad (1)$$

JVDD 알고리즘은 채널정보를 가지 메트릭 계산에 반영하여 하나의 비터비 프로세서로 검출과 복호를 수행하는 방법이다. 아래의 그림 1에 표기된 위상차 검출기에서 심볼 검출을 하지 않고 위상정보만 비터비 프로세서로 전달하여 비터비 프로세서에서 검출과 복호를 수행한다.

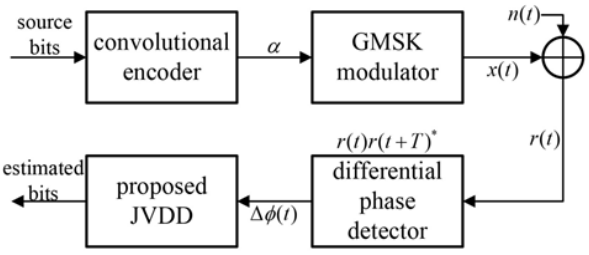

Fig. 1. Block diagram of LE-coded mode in BLE 5.0 system.
그림 1. BLE 5.0 LE-Coded 시스템 모델

아래의 그림 2는 노이즈가 없는 경우에 GMSK가 유발하는 ISI에 의하여 코드워드 '00'가 서로 다른 4가지의 종류로 발생 가능한 $\Delta\Phi(t)^{(k)}_{NF,00}(t)$($k$={0,1,2,3}) 위상차 경로를 표현하고 있다. ISI가 없는 경우에서는 그림 2. (a) 하나의 형태만 존재 하지만, ISI가 발생하는 환경에서는 이전 심볼의 영향을 받게 되므로 이전 코드워드 시퀀스가 어떤 모습인가에 따라서 현재의 코드워드 시퀀스 모양이 달라진다. 그림 2. (b)에서 시작위상이 '1' 근처에서 시작하기 때문에 이전의 코드워드 시퀀스는 '1'로 끝나는 '01'또는 '11'로 유추를 될 수 있다.

JVDD 알고리즘의 특징은 이전에 수신된 코드워드로 위상차 $\Delta\Phi(t)$와 추정된 위상차 코드워드 시퀀스 $\Delta\Phi(t)^{(k)}_{NF,00}(t)$를 수식 (2)에 적용하여 유클리디언 거리계산에 수행한다.

$$ED^{k}_{00} = \left|\Delta\Phi(t) - \Delta\Phi(t)^{(k)}_{NF,00}\right|^{2} \quad (2)$$

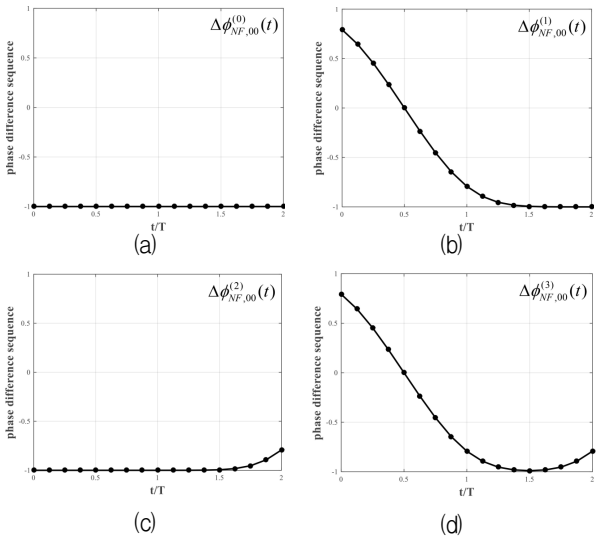

Fig. 2. Four types of phase difference sequences for codeword '00'.
(a) : $\Delta\Phi(t)^{(0)}_{NF,00}(t)$ (b) : $\Delta\Phi(t)^{(1)}_{NF,00}(t)$
(c) : $\Delta\Phi(t)^{(2)}_{NF,00}(t)$ (d) : $\Delta\Phi(t)^{(3)}_{NF,00}(t)$
그림.2. 코드워드 00에 대한 4가지 종류의 위상차 시퀀스
(a) : $\Delta\Phi(t)^{(0)}_{NF,00}(t)$ (b) : $\Delta\Phi(t)^{(1)}_{NF,00}(t)$
(c) : $\Delta\Phi(t)^{(2)}_{NF,00}(t)$ (d) : $\Delta\Phi(t)^{(3)}_{NF,00}(t)$

그림 2에 표시된 4가지 타입을 모두 사용하지 않고 두 가지 타입, k={0,1}만을 가지고 연산을 한다. k={0,1}의 시작 위상 모양은 k={2,3}과 매우 유사하고 k={0,1}만을 사용해도 성능저하는 발생하지 않는다. 이전에 수신된 코드워드는 현재 수신된 코드워드 시퀀스의 초기 위상 시작점과 코드워드 시퀀스 전체 위상 변화에 영향을 주게 된다. SMU에 저장된 이전 시간의 생존경로를 추적함으로써 현재 수신된 코드워드의 위상의 시작점을 유추할 수 있기 때문에 가장 최적의 가지 메트릭을 계산할 수 있다.

그림 3은 가지 메트릭 계산을 위해서 그림 2에서 제시된 최적의 위상차 코드워드 시퀀스를 유추하는 방법을 기술한 예제이다. 가지 메트릭이 "00"인 연산을 위한 예를 들었다. n번째에서 state S0에 대한 가지 메트릭 연산을 위해 이전의 n-1번째에서 결정된 생존경로를 이용하는 과정이다. n-1번째에서 ACS(add compare select) 과정을 통해서 결정된 생존경로 정보는 SMU에 저장되어 있고 n번째에서 이 정보를 활용하여 가지 메트릭 계산에 적용한다.

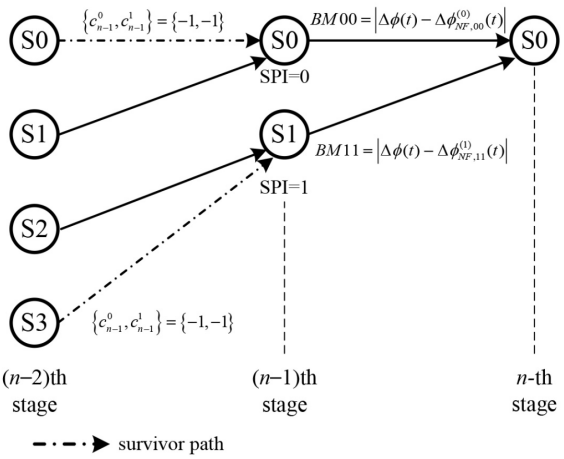

Fig. 3. Branch metric calculation example.
그림 3. 가지 메트릭 연산 예제

SMU에 저장된 정보는 어느 가지가 선택되었는지를 나타내는 생존경로 인덱스로서 그림 3에서 SPI로 표기되었다. n-1번째 S0는 n-2번째 S0 또는 S1 중에서 하나를 선택하게 된다. SPI가 0인 경우에는 상위 가지가 선택된 것이고, SPI가 1이면 하위 가지가 선택된 것이다. 이 경우에는 n번째 S0에서는 S0→S0→S0와 S3→S1→S0 경로 두 가지를 비교하게 된다. n-1번째의 SPI를 확인한 후 이것을 그림 2에 표시된 인덱스 k로 사용하여 가지 메트릭 계산에 바로 적용하게 된다.

2. JVDD 비터비 프로세서 성능평가

성능평가는 위상차 검출기로 검출을 하고 비터비 프로세서로 복호를 수행한 시스템과 JVDD 알고리즘을 적용한 비터비 프로세서 시스템을 비교하였다. 그림 4에서 JVDD 알고리즘이 하나의 비터비 프로세서와 위상차 검출기를 사용한 것과 비교하여 0.6dB의 성능향상이 있음을 보여준다.

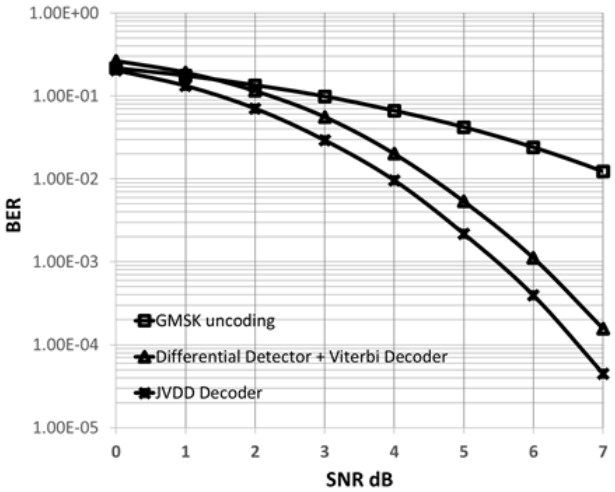

Fig. 4. JVDD Algorithm simulation.
그림 4. JVDD 알고리즘 시뮬레이션

III. 제안된 비터비 프로세서

1. BLE 비터비 프로세서 구조

BLE 5.0은 부호율 1/2, 제한길이 (constraint length) K=4인 길쌈부호기를 사용한다. 생성 다항식은 식(3)과 식(4)으로 표현된다.

$$G_0(X)=1+X+X^2+X^3 \quad (3)$$

$$G_1(X)=1+X^2+X^3 \quad (4)$$

부호기의 초기값은 “0”이 되고 패킷 전송의 마지막에는 부호기를 초기화하기 위해서 연속으로 “0”을 3번 입력하여 부호기를 항상 “0”으로 초기화한다. 그림 5는 JVDD 비터비 프로세서의 하드웨어 구조이다. SMU에 저장된 이전의 SPI를 BMU (branch metric unit) 블럭에 전달하여 ACS 연산을 위한 가장 적합한 위상차 코드워드 시퀀스 후보를 선택한다. 선택된 코드워드 시퀀스를 가지고 ACS 과정을 수행하여 ISI정보가 반영된 가지 메트릭 연산을 하게 된다. ACS 수행과정에서 일반적인 비터비 프로세서와 가장 차이가 나는 것은 ACS를 수행하기 전에 SMU에서 이전의 연산 결과를 확인하는 기능이 필요하며 BMU는 모든 경우에 대한 가지 메트릭을 표현하는 테이블을 가지고 있어야 한다.

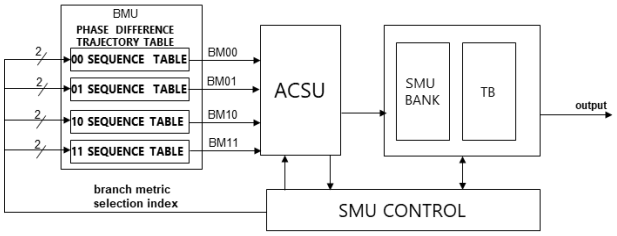

Fig. 5. Block diagram of JVDD Viterbi processor.
그림 5. JVDD 비터비 프로세서 블록 다이어그램

2. 제안하는 SMU구조

TB 방식을 적용하는 일반적인 k-pointer 비터비 프로세서는 메모리 bank의 크기에 비례해서 복호지연 시간이 발생한다[10]. 메모리 bank의 크기는 역추적 길이(trace back depth)보다 최소한 같거나 커야 한다. 대부분 k-pointer 또는 SE(state exchange)

방식을 적용하면서 구현된 경우는 복호지연 시간을 줄이기 위해서 역추적 길이 보다 큰 메모리 bank를 사용하게 된다[11]. 만약 수신되는 패킷의 길이가 bank의 크기와 일치하지 않는 경우 SMU bank를 임의로 다 채운 후에 TB→DC(decoding read)를 하여야 한다. 즉, BLE에서 요구하는 TIFS의 2us 오차를 만족하는 과정에서 일정한 복호시간이 발생하는 것이 아니라 수신되는 패킷의 길이에 따라서 항상 서로 다른 지연시간이 발생할 수 있다.

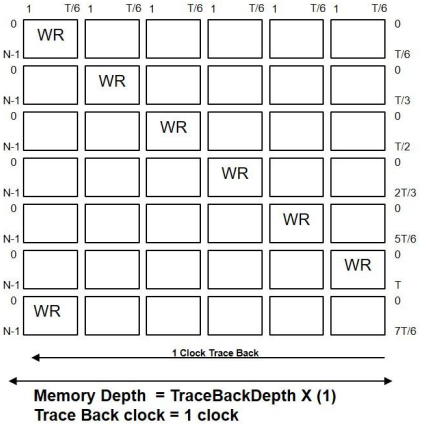

Fig. 6. SMU Operation of Proposed Viterbi Processor.
그림 6. 제안된 비터비 프로세서 SMU 동작 과정

제안하는 SMU 구조에서는 TB와 DC 시간을 최소화하면서 수신되는 패킷의 길이에 최소한의 영향을 받도록 하여 비터비 복호 지연시간을 최대한 작게 하는 것이다. 제안하는 SMU는 메모리 대신 레지스터를 bank에 사용하고 TB 동작에서는 combinational 로직을 사용하여 1 클럭 사이클만에 복호를 한다. *k*-pointer 방식과 차이가 나는 것은 TB와 DC과정이 구별되지 않는 차이점이 있다. *k*-pointer는 WR(Write)→TB→DC의 순서대로 동작을 하게 되지만, 아래의 그림 6에서 보면 제안하는 방식에서는 DC 과정이 TB에 포함되어 있기 때문에 WR→TB로 끝나게 된다. 길쌈부호기의 레지스터에 저장된 입력은 복호하려는 데이터와 동일하고 또한, TB에서 천이되는 state와 동일하다. 그러므로, TB 마지막에 수렴되는 state가 복호하려는 데이터가 된다. 그림 6에서 각 bank의 출력 DEC 신호가 복호된 데이터에 해당된다. *k*-pointer 방식에서는 bank 사이즈와 동일한 크기의 복호 데이터를 출력하지만 제안하는 방식은 TB가 수행 될 때마다 state 단위로 복호를 하게 된다. BLE 5.0 길쌈부호는 제한 길이가 4이기 때문에 3bit 단위로 복호된 데이터를 출력하게 된다. 설계된 구조의 특징은 SE와 동일하게 state 단위로 복호하는 방식을 적용하고 있다.

Table 1. SMU TB MUX control.
표 1. SMU TB MUX 제어 순서

SEQ	SEL0	SEL1	SEL2	SEL3	SEL4	SEL5	DATA
1	DEC	DEC	DEC	DEC	DEC	START	DEC0
2	START	DEC	DEC	DEC	DEC	DEC	DEC1
3	DEC	START	DEC	DEC	DEC	DEC	DEC2
4	DEC	DEC	START	DEC	DEC	DEC	DEC3
5	DEC	DEC	DEC	START	DEC	DEC	DEC4
6	DEC	DEC	DEC	DEC	START	DEC	DEC5

그림 7은 제안하는 SMU 블록구조이다. 각 bank에 3개의 state 천이를 저장하고 있고 SMU는 6개의 bank로 이루어져 있다. 표 1에는 bank 제어 순서가 표시되어 있다. TB가 실행될 때 첫 번째 bank로

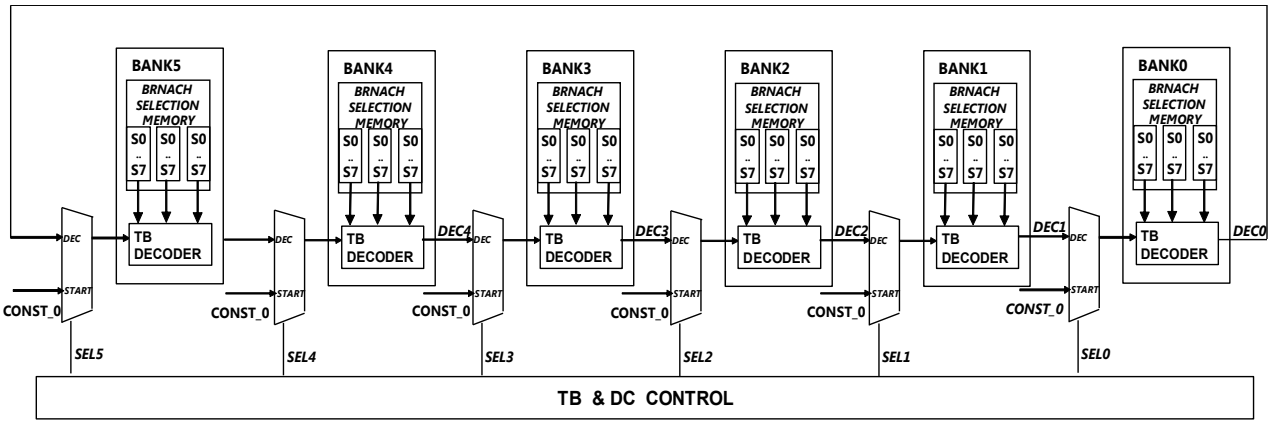

Fig. 7. SMU block diagram of proposed viterbi processor.
그림 7. 안된 비터비 프로세서 SMU 블록 다이어그램

선택된 MUX의 선택 신호는 START를 선택하게 되어 state “0”에서 TB를 시작하고 되고 나머지 bank는 이웃하는 bank의 출력 신호를 받아서 TB를 진행하게 된다.

3. 설계 결과

제안된 SMU 블록의 클럭 주파수는 2MHz이고, 생존경로를 저장하는 메모리 유닛은 144개이다. 본 설계에서는 빠른 복호를 위해서 레지스터를 사용하였다. BLE 5.0처럼 길쌈 부호기의 제한 길이가 크지 않아서 메모리 셀이 적게 사용되는 경우에는 메모리를 직접 사용하는 것보다는 레지스터로 구성하는 것이 사이즈 측면에서 유리하다. 제안된 비터비 프로세서와의 비교를 위해 k-pointer와 MSE (modified state exchange) 구조를 비교 하였다. 제안된 구조와 MSE 구조의 공통점은 각 bank마다 저장되는 생존 경로 수가 동일하다는 것이며, 차이점은 복호지연시간 및 전체 SMU 크기에 차이가 있다. 표 2에서 기술된 latency 항목은 비터비 프로세서의 초기 지연시간을 나타내고 T는 역추적 길이를 나타낸다.

k-pointer와 MSE는 TB를 하는 과정 중에도 WR, DC를 위해서 별도의 bank가 필요하다. 그러나 제안된 SMU 구조는 1 클럭 사이클 동안 TB, DC가 모두 완료되므로 WR 동작 동안에 다른 일을 처리하지 않아서 제한 길이보다 더 큰 bank를 사용하지 않는 장점이 있다. 또한, 표 2에 제시된 바와 같이 제안하는 SMU 구조는 MSE와 비교해서 메모리 유닛을 33% 감소 가능하다. GF 55nm CMOS 공정 기반 논리합성 결과, 제안된 SMU 블록은 4K 게이트로 구성되며, 비터비 복호기는 총합 12K 게이트로 구현 가능함을 확인하였다.

Table 2. Comparison of memory capacity, logic gate count and latency for SMU.

표 2. SMU의 메모리 용량, 논리 게이트 수 및 복호 지연 시간 비교

	3-pointer even [10]	MSE [12]	Proposed SMU
Memory unit	8×9×6(432)	8×3×8(192)	8×3×6(144)
Logic Gate count	5.4K	4.3K	4K
Latency	3T	3T/2	T

Ⅳ. 결론

본 논문에서는 BLE 5.0 비터비 프로세서 설계를 위해 JVDD 알고리즘이 적용된 구조를 적용하며 TIFS 요구조건을 충족하기 위한 비터비 프로세서의 SMU 구조를 제안하였다. 제안된 구조의 초기 지연시간과 메모리 유닛은 MSE와 비교하여 33%가 줄었고, 3-Point Odd 구조와 비교하여 66% 줄었다. 또한, 1 클럭 사이클 동안에 TB와 DC를 완료할 수 있기 때문에 2MHz 동작을 하면서 LE-Coded 500Kbps를 복조할 수 있다. 제안된 JVDD는 GF 55nm CMOS 공정으로 합성한 결과 12K 게이트 수를 포함함을 확인하였다.

References

[1] *BLUETOOTH SPECIFICATION Version 5.0* | Vol 6, Part A

[2] G. D. Forney, Jr., “Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference,” *IEEE Trans. Inform. Theory*, vol.IT-18, pp.363-378, 1972. DOI: 10.1109/TIT.1972.1054829

[3] K. T. Nimisha and P. Biswagar, “Viterbi algorithm based Bluetooth low energy receiver for IoT,” *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology* (RTEICT), Bangalore, pp.978-981, 2017. DOI: 10.1109/RTEICT.2017.8256744

[4] A. James and K. S. Chan, “Joint Detector Demodulator Decoder (JDDD) over ISI Channels,” *2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW,* pp.1-5, 2017. DOI: 10.1109/VTCSpring.2017.8108440

[5] V. J. Alappat, M. Motani and C. K. Sann, “An Adaptive Joint Viterbi Detector Decoder (AJVDD),” *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS),* Gold Coast, QLD, pp.1-5, 2016. DOI: 10.1109/ICSPCS.2016.7843343

[6] Kheong Sann Chan, S. S. B. Shafiee, E. M. Rachid and Yong Liang Guan, “Optimal Joint Viterbi Detector Decoder (JVDD) over AWGN/

ISI channel," *2014 International Conference on Computing, Networking and Communications (ICNC),* Honolulu, HI, pp.282-286, 2014.
DOI: 10.1109/ICCNC.2014.6785346
[7] C. Park, Y. Jung, J. Kim and Y. Jung, "Joint Viterbi detection and decoding algorithm for bluetooth low energy systems," *in Electronics Letters*, vol. 56, no.6, pp.310-312, 17 3 2020.
DOI: 10.1049/el.2019.2621
[8] D. A. F. Ei-Dib and M. I. Elmasry, "Low-power register-exchange Viterbi decoder for high-speed wireless communications," *2002 IEEE International Symposium on Circuits and Systems*. Proceedings (Cat. No.02CH37353), Phoenix-Scottsdale, AZ, USA, pp.V-V, 2002. DOI: 10.1109/ISCAS.2002.1010809
[9] T. K. Truong, M. -. Shih, I. S. Reed and E. H. Satorius, "A VLSI design for a trace-back Viterbi decoder," in *IEEE Transactions on Communications*, vol.40, no.3, pp.616-624, 1992.
DOI: 10.1109/26.135732
[10] G. Feygin and P. Gulak, "Architectural tradeoffs for survivor sequence memory management in Viterbi decoders," in *IEEE Transactions on Communications*, vol.41, no.3, pp.425-429, 1993.
DOI: 0.1109/26.221067
[11] Chun-Yuan Chu, Yu-Chuan Huang and An-eu Wu, "ower efficient low latency survivor memory architecture for Viterbi decoder," 2008 *IEEE International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, pp.228-231, 2008.
DOI: 10.1109/VDAT.2008.4542454
[12] Yun-Ching Tang, Do-Chen Hu, Weiyi Wei, Wen-Chung Lin and Hongchin Lin, "A memory-efficient architecture for low latency Viterbi decoders," 2009 *International Symposium on VLSI Design, Automation and Test*, Hsinchu, pp.335-338, 2009.
DOI: 10.1109/VDAT.2009.5158163

BIOGRAPHY

Chul-hyun Park (Member)

2001 : BS degree in Electrical Engineering, Dongguk University.
2003 : MS degree in Department of Electrical and Electronic Engineering, Yonsei University.
2003~2007 : Research Engineer, Samsung Electronics.

2016~present : Ph.D degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Yongchul Jung (Member)

2015 : BS degree in School of Electronics and Information Engineering, Korea Aerospace University.
2017 : MS degree in School of Electronics and Information Engineering, Korea Aerospace University.
2017~present : Ph.D degree course in School of Electronics and Information Engineering, Korea Aerospace University.

Yunho Jung (Member)

1998 : BS degree in Department of Electrical and Electronic Engineering, Yonsei University.
2000 : MS degree in Department of Electrical and Electronic Engineering, Yonsei University.

2005 : Ph.D degree in Department of Electrical and Electronic Engineering, Yonsei University.
2005~2007 : Senior Engineer, Samsung Electronics.
2007~2008 : Research professor, Institute of Information Engineering, Yonsei University.
2008~present : Professor, School of Electronics and Information Engineering, Korea Aerospace University