

선박용 화재탐지장치의 통신 Error를 감소시키기 위한 Turbo 복호기 개발에 관한 연구

정병홍, 최상학, 오종환, 김경석

A Study on a Development of Turbo Decoder for reducing communication error of fire detection system for Marine Vessels

Abstract

In this study, adapted Turbo Coding Algorithm for reducing communication error of fire detection system for marine vessels, especially image transmission. and proposed decoding speed increasing method of Turbo Coding Algorithm. The results are as follows :

- 1) Confirmed that a Turbo Code is so useful methods for reducing communication error in lots of noise environments.
- 2) Proposed technology in this study of speed increasing method of Turbo Coding Algorithm proved 2 times speed up effect than normal Turbo Code and communication error reducing as well in the board made by VHDL software & chips of ALTER Company.

1. 서론

본 연구는 선박용 화재탐지장치의 개발에 있어, 현재 선박에 설치되어 있는 화재탐지장치가 탐지기 및 탐지센서 등의 잦은 오동작으로 경보가 발령되어도 당직자가 이를 무시해버리거나 화재사실을 직접 현장까지 가서 확인하는 과정을 거치는 등으로 결국 초기소화의 기회를 잃고 누장대응으로 막대한 손실을 초래하는 단점을 보완하기 위하여 해당 탐지구역의 탐지센서가 작동하면 화재현장의 영상이미지를 화재탐지장치에서 원격으로 모니터링할 수 있도록 하였다. 이렇게 영상을 모니터링함에 있어 별도의 통신선을 추가 설치하지 않고 전원선에 화재에 관한 Data 및 Image 신호를 동시에 전송하는 방식을 채택하였다.

일반적으로 데이터통신에 있어서 통신 Error를 줄이기 위하여 순환중복검사(CRC ; Cyclic Redundancy Check) 및 Channel Coding의 방법을 주로 사용하고 있다. 그러나 선박과 같이 잡음이 많은 환경에서는 통신 Error에 대책을 별도로 강구하는 것이 바람직하다.

잡음에 대한 해결방안의 하나로 1993년 Berrou씨가 Turbo-Code에 관한 논문을 발표한 이후 1~2년간에 걸쳐 Turbo-Code의 성능이 입증되었고, 현재는 그 적용분야에 대한 연구가 집중되고 있으며 시스템 엔지니어링 차원에서 Turbo-Code 설계 기술이 축적/고도화되고 있는 단계이다. 지금은 무선통신, 전력선통신 등 각종 통신방식에 Turbo coding Algorithm이 주로 적용되고 있다. 또한, 최근에 Issue가 되고 있는 IMT2000 시스템의 무선전송기술규격(RSPS series)이 1999년 말에 완료되어 IMT2000 시스템에 Turbo-Code를 적용하도록 확정되었다.

그러나 현재의 Turbo-Code 응용 및 적용범위는 기존의 Viterbi 복호기와는 달리 처리할 데이터량 연산작용이 매우 복잡해 처리속도가 저속이어서 저속 서비스에만 적용될 예정이며, 고속 서비스로의 적용은 불가능한 실정이다. Turbo-Code의 고속화가 해결되면 이의 적용 범위가 IMT2000 뿐만 아니라 위성통신 및 타 이동 무선통신에서의 적용이 확대되고, 음성, 데이터, 영상 등 무선 멀티미디어 서비스 개발과 함께 점차 확대될 전망이다. Turbo-Code는 2000년 후, 이동 통신 시스템의 오류제어기법방식으로, Viterbi decoder를 대체할 차세대 오류제어기법으로서 향후 IMT2000을 포함한 차세대 이동 단말기 산업을 고려할 때 현시점에서의 고속화 설계 기술개발 및 지적 재산권 경쟁은 절실히 요구되는 분야로 전망된다.

따라서 본 연구에서는 화재탐지장치의 통신 Error를 줄이기 위하여 Turbo Coding Algorithm을 적용하고, 기존의 Turbo Coding Algorithm이 복호시간이 오래 걸리는 단점을 개선하여 시스템의 실시간 처리를 가능하게 하였다.

2. 통신 Error 정정을 위한 적용기술의 이론 해석

다음 Fig. 1은 본 연구에서 채택한 선박에서의 전력선통신을 이용한 영상모니터링시스템의 구성예를 나타낸 것이다.

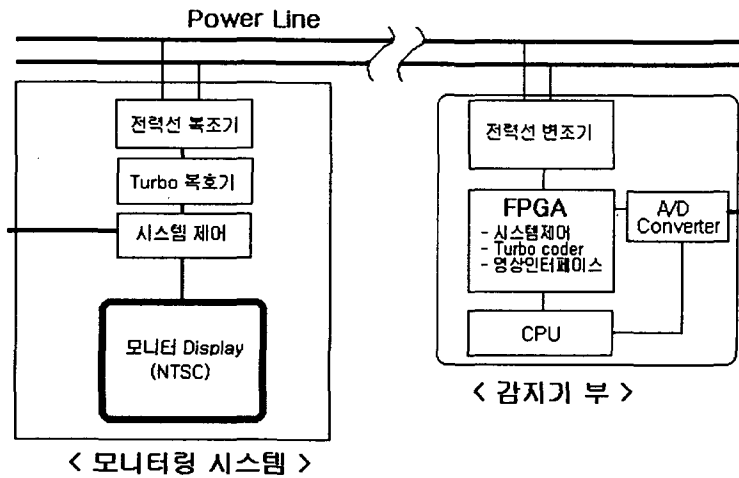


Fig. 1 선박에서의 전력선통신을 이용한 영상모니터링

2.1 CRC 기술

CRC(Cyclic Redundancy Check)는 순환중복검사 방식으로 송신측에서는FCS(Frame Check Sequence) n-bit를 구한 후 이를 원래의 Data k-bit에 추가하여 (n+k)-bit를 송신한다.

- T = 전송될 (k+n)bit Frame (n<k인 경우)
- M = k-bit message, T의 첫 k개 bit
- F = n-bit FCS, T의 마지막 n-bit
- P = (n+1)bit의 Pattern, 젯수(divisor)

여기에서 정보 T를 송신한 후 수신단에서 "T"를 수신하여 "P"로 나누어 떨어지도록 하여야 한다.

따라서, 송신시 보내야 하는 Data는 다음과 같다.

- T= (Message의 n-bit Left Shift) + F(Frame Check Sequence)
- F는 Message를 n-bit Left Shift한 후 이를 P로 나눈 나머지이다.
- 수신측에서는 전송된 Data를 P로 나누어 나머지가 없어야 한다.
- Pattern P는 다음과 같다.

CRC-12 = 110000000111

CRC-16 = 11000000000000101

CRC-CCITT = 100010000001000001

CRC-32 = 100000100110000010001110110110111

본 연구에서 채택한 시스템에서는 CRC-12를 적용하였다. CRC 검사 결과 나머지가 "0"이 되지 않아 Error가 발생하면 재전송을 요청한다.

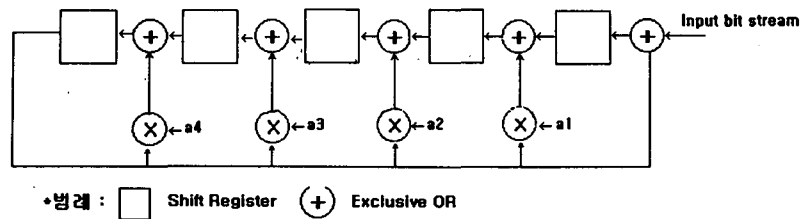
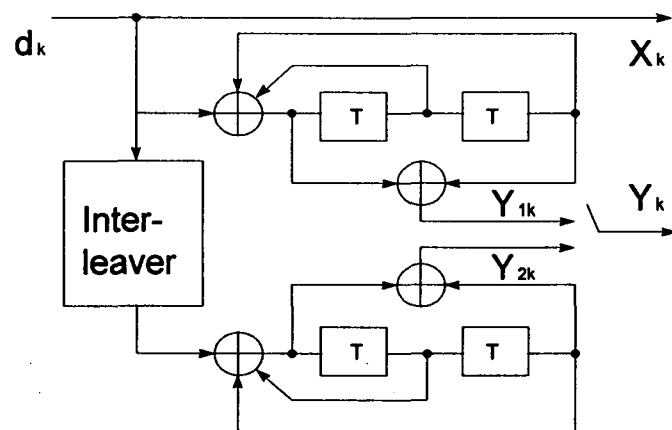


Fig. 2 RS-485통신의 Error율을 줄이기 위한 수신측의 순환 중복검사(CRC) 개념도

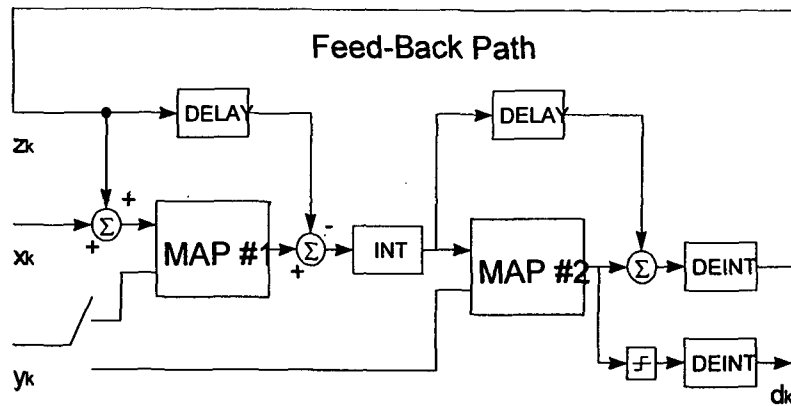
2.2 Turbo Coding Algorithm 개발

Turbo Coding 기술은 Conventional Code를 대처할 Shannon's Limit에 가장 근접한 차세대 오류정정 부호방식으로 1993년 Berrou가 발표한 방법이다.

CRC에 추가로 Error를 완벽하게 정정할 수 있는 Turbo Code Algorithm을 적용함으로써 완벽한 시스템을 구현할 수 있다. Turbo 부호기는 MAP기반의 부호기는 잡음분산평가의 문제점도 있지만, 가장 큰 문제점은 MAP의 복잡성과 많은 연산량으로 인해 처리속도가 저속이고 메모리도 방대하고 설계가 어려우므로 다음과 같은 방법으로 속도를 증가시켰다.



(a) Turbo 부호기



(b) Turbo 복호기

Fig. 3 Turbo Code에 대한 Trellis Diagram

2.2.1 복호속도 개선방안 : 고속 Radix-4방식의 MAP Turbo 복호알고리즘

MAP기반의 Turbo-Code 복호시, Radix-2 방식은 임의의 k시점에서 복호할 때 k-1 시점에서의 순방향 state metric α_{k-1}^m , k+1 시점에서 역방향 state metric β_{k+1}^m 그리고 k 시점에서 branch metric δ_k^m 를 이용하여 k 시점에서 "0" 과 "1"에 대한 Log Likelihood Ratio인 LLR을 구하여 1비트를 복호한다. 이에 반해 radix-4방식은 두상태 이전인 k-2시점에서 α_{k-2}^m , 두상태 이후인 k+2시점에서 β_{k+2}^m 를 구하여 k시점에서 2 비트를 동시에 복호하기 때문에 radix-2 방식보다 속도가 2배 빠르며, 저장되는 메모리도 절반으로 감소할 수 있다. 과거 2단의 수신 비트를 입력받아 한꺼번에 처리하는 Radix-4 방식의 trellis 구조는 Radix-2 방식의 2개 시점을 하나의 시점으로 간주하여 처리하며 Radix-2 방식에서 Radix-4방식으로의 trellis 구조변경은 다음 Fig. 4와 같다.

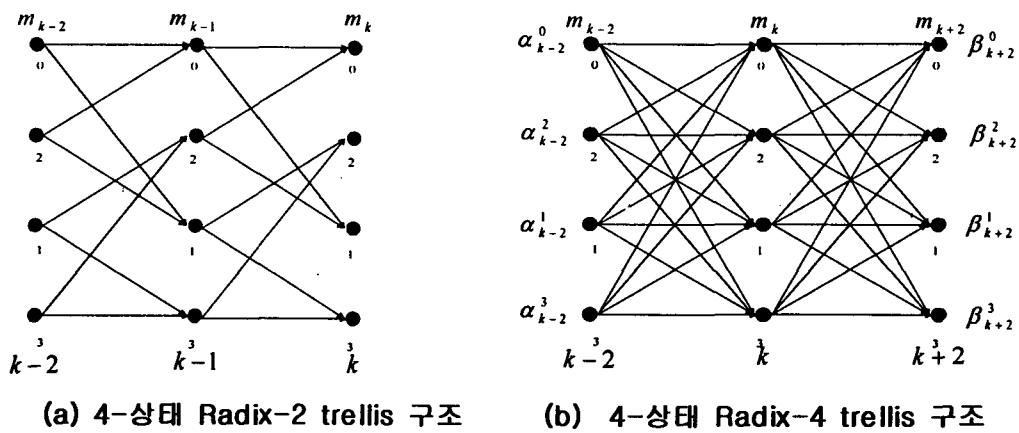


Fig. 4 4-상태 Radix trellis 구조

순방향, 역방향, branch metric을 이용하여 각 상태에서 LLR을 구하기 위한 연산작 용은 Fig. 5와 같이 구한다.

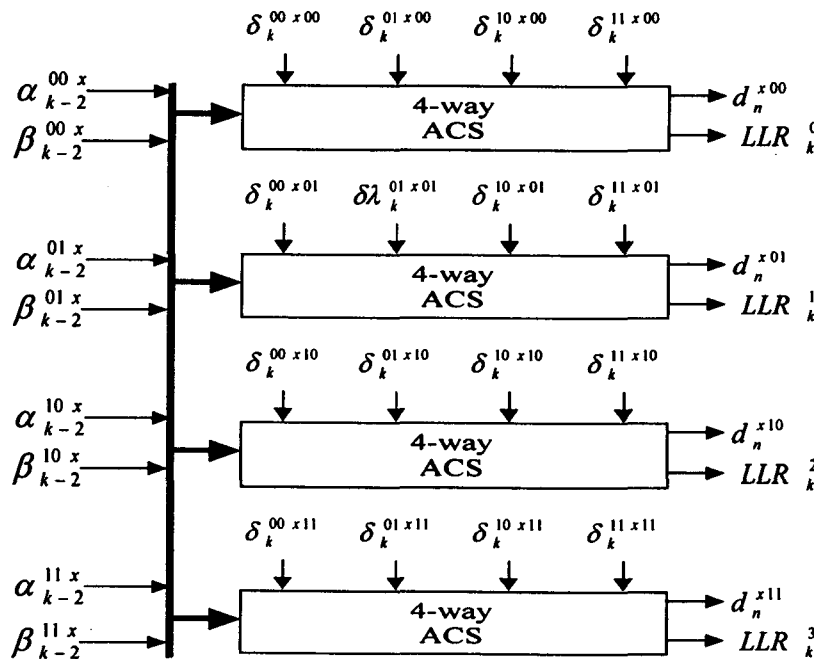


Fig. 5 Radix-4 state metric 구조

δ^{000} 는 branch codeword 0000에 대한 수신 비트의 branch metric을 의미하며, α_{k-2}^{000} 는 k-2 시점에서 상태 000에 대한 순방향 state metric을 의미하며, β_{k+2}^{000} 는 k+2 시점에서 상태

000에 대한 역방향 state metric을 의미하며, LLR_k 는 k 시점에서의 각 상태에서 “0”과 “1”이 될 확률값을 의미하며, d_k^m 은 k시점에서 복호된 비트값이 “00”임을 의미한다.

실제 MAP의 구현은 Log 차원에서 수행되므로, 두상태를 하나의 상태로 간주하여 결합하는 radix-4 구조는 branch codeword와 수신된 비트와의 해밍거리를 구하는 viterbi 복호기의 구현 예처럼 단순하지가 않으며, Log차원에서 결합하기 때문에 복잡한 수식을 통하여 증명하고 난 뒤 구현해야 한다.

Log 차원에서의 radix-4 방식 적용시 다음의 항목에 대한 공식을 만들어 기존의 radix-2방식과 성능이 동일함을 시뮬레이션을 통하여 증명하여야 한다.

- Radix-4방식의 새로운 branch metric δ_k^m 계산방법
- Radix-4방식의 새로운 순방향 state metric α_k^m 계산방법
- Radix-4방식의 새로운 역방향 state metric β_k^m 계산방법
- Radix-4방식의 Iteration시 새로운 extrinsic z_k 계산방법
- Radix-4방식의 복호비트를 결정하기 위한 새로운 LLR 계산방법

2.2.2 연산방식 개선방안 : 고속 덧셈/곱셈 알고리즘

Radix-4 구조로의 병행과 더불어 고속화를 위해서는 각 상태 metric과 순방향, 역방향 state metric, LLR을 구하기 위한 고속 덧셈 알고리즘도 병행하여야 한다. 고속 덧셈 알고리즘으로는 Carry Save Adder, Carry Select Adder, Carry Lookahead Adder가 있으며 기존의 Full Adder보다 계산속도가 빠름이 이미 증명되었다.

(1) Carry Save Adder

- 2-operand에 대한 Carry Save Adder

$$S_i = A_i \oplus B_i$$

$$C_i = A_i \cdot B_i$$

- 3-operand에 대한 Carry Save Adder

$$S_i = A_i \oplus B_i \oplus C_i$$

$$C_i = A_i \cdot B_i + C_i \cdot (A_i \oplus B_i)$$

(2) Carry Lookahead Adder

$$G_i = A_i \cdot B_i \text{ (Carry Generation)}$$

$$P_i = A_i \oplus B_i \text{ (Carry Propagation)}$$

$$S_i = (A_i \oplus B_i) \oplus C_{i-1} = P_i \oplus C_{i-1}$$

$$C_i = A_i \cdot B_i + (A_i \oplus B_i) \cdot C_{i-1} = G_i + P_i \cdot C_{i-1}$$

(3) Carry Select Adder

전체 Adder를 몇 개의 Section으로 나눈 후 각 Section별로 Carry가 '0'인 경우와 '1'인 경우를 미리 연산하고, 하위 비트 Section의 Carry에 해당하는 상위 비트의 Section 출력을 선택하여, 하위 비트의 Section과 상위 비트의 Section의 조합으로 값을 출력하는 Adder이다.

2.2.3 복호지연 개선방안 : Center-to-Top 방식

한 클럭에 2비트를 복호하는 radix-4 알고리즘과는 달리 1비트만 복호하면서도 복호지연을 절반으로 감소하는 알고리즘을 제시한다. 여기서 제시한 알고리즘은 Turbo 복호기의 구현시 문제점은 메모리크기 및 복호지연이 가장 큰 문제점인데 메모리 크기에 비례하여 복호지연이 발생하므로 메모리를 절반으로 줄이면서 복호지연 및 복호속도를 2배 향상시킨 알고리즘이다.

다음 Fig. 6은 제시한 고속 turbo 복호기의 복호과정에 대한 그림이다.

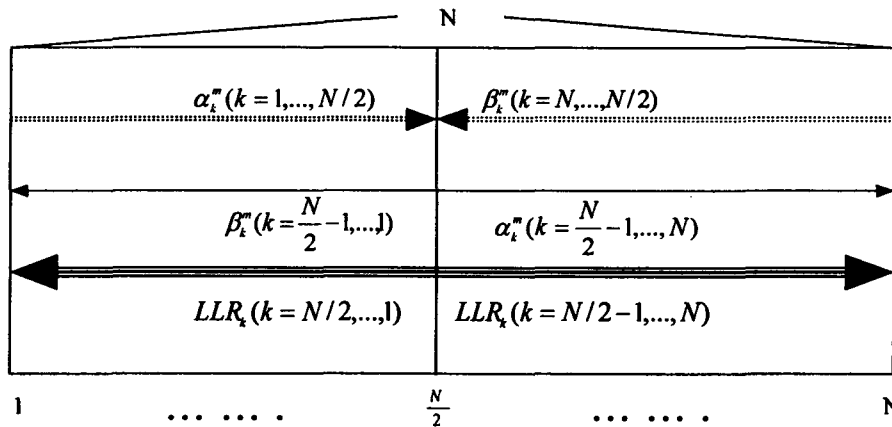


Fig. 6 고속 turbo 복호기의 복호과정

여기서 α 와 β 를 동시에 구한 다음 메모리 절반 위치에 있을 때 복호를 한다. 이렇게 함으로써 복호시점이 역방향 상태 메트릭을 모두 구한 다음 복호를 하는 기존의 방식과는 달리 복호 메모리의 절반 위치에서 복호를 하기 때문에 복호속도가 기존의 방식에 비해 2배 빠르다. 시뮬레이션 결과 기존의 방식과 성능이 일치함을 알 수 있다. 복호시 처리 과정은 다음과 같이 세 단계로 나눌 수 있다.

(1) 단계 1

$i = 0, 1$ 에 대해 순방향 및 역방향 상태 매트릭을 초기화시킨다.

$$\begin{aligned} \alpha_0^i(S_b^i(m)) &= 1 \text{ for } m=0, \\ &= 0 \text{ for } m \neq 0 \\ \beta_N^i(S_b^i(m)) &= 1 \text{ for } m=0, \\ &= 0 \text{ for } m \neq 0 \end{aligned}$$

(2) 단계 2

N symbols을 모두 수신하여 저장한 후 모든 상태 m 에 대해 $N/2$ 까지의 순방향 매트릭 및 N 에서 $N/2$ 까지의 역방향 상태 매트릭을 동시에 구한다.

$$\hat{\alpha}_k^i(m) = \exp\left(\frac{2}{\sigma^2}(x_k i + y_k Y_k(i, m))\right) \prod_{j=0}^{k-1} \hat{\alpha}_{k-1}^j(S_b^j(m)) \quad (k=1, \dots, N/2)$$

$$\hat{\beta}_k^i(m) = \prod_{j=0}^{k-1} \hat{\beta}_{k+1}^j(m) \exp\left(\frac{2}{\sigma^2}(x_{k+1} j + y_{k+1} Y_{k+1}(j, S_b^i(m)))\right) \quad (k=N/2, \dots, N)$$

(3) 단계 3

순방향 및 역방향 상태 매트릭을 구한 후 중간지점부터 LLR을 구하여 양쪽으로 전전하면서 복호한다.

$$L(d_i)_{n,m} = \log \frac{\sum_m \alpha_i^1(m) \beta_i^1(m)}{\sum_m \alpha_i^0(m) \beta_i^0(m)} \quad (k=N/2, \dots, N)$$

$$L(d_i)_{m'} = \log \frac{\sum_m \alpha_i^1(m) \beta_i^1(m)}{\sum_m \alpha_i^0(m) \beta_i^0(m)} \quad (k=(N/2)-1, \dots, 1)$$

3. 결론

Turbo 복호기의 VHDL(Very high speed intergreted circurt Hardware Description Language)코드를 Alter사의 Design compiler를 이용하여 compile하고 합성하였다. Turbo 복호기는 FLEX10K70RC240-3칩으로, TB memory는 외부 RAM을 사용하지 않고 FLEX10 칩안의 내부 EAB(Embedded Array Block)을 사용하여 구현하였다.

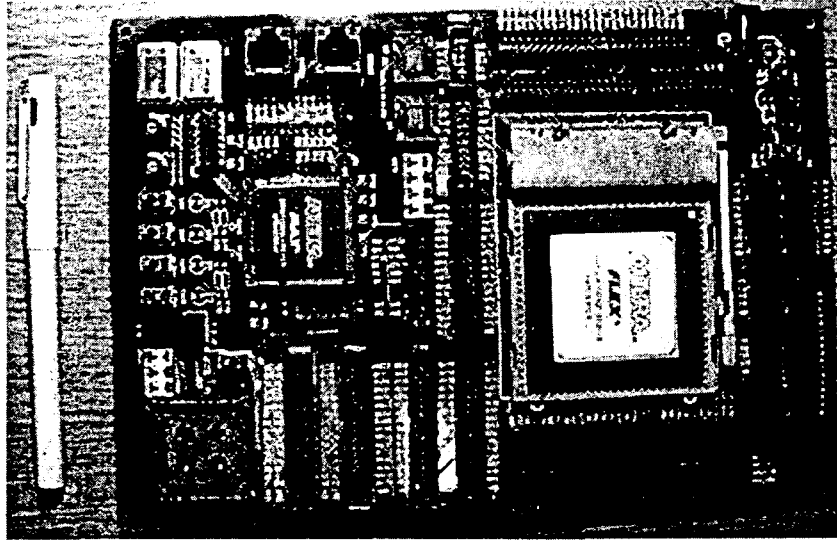


Fig. 7 FPGA로 구현한 Error Correction Board의 제작사진

시뮬레이션에 이용한 입력신호는 임의의 timing error를 부가한 정현파 신호이다. Interpolator의 계수를 결정짓는 cnt는 점점 감소하면서 70으로 수렴함을 알 수 있다. 그리고 oput11과 oput22는 복조 된 신호이다. 신호의 결과치는 2비트로 표현되므로 십진수 3은 이진수 "11"로서 -1을 나타내므로, I채널과 Q채널의 입력값이 동시에 1, 0, 1, 0이므로 정확히 복조됨을 알 수 있다. STR과 CPR을 연동한 회로는 192.8[ns] 내에서 동작한다.

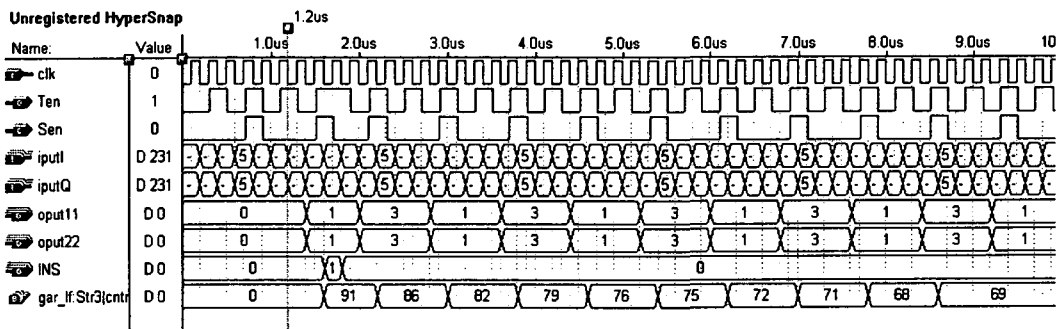


Fig. 8 고속 turbo 복호기의 FPGA로 Simulation한 결과

위 그래프에서 보듯이 기존의 Turbo-Code의 복호속도에 비하여 복호속도는 2배 향상되었고, Error rate는 감소되었다. 따라서 본 연구에서 제안한 Turbo-Code를 이용한 통신 Error 정정 및 복호속도 개선기술은 선박의 화재탐지장치에 적용한 결과 우수한 특성을 얻을 수 있었다. 이상의 결과를 정리하면 다음과 같다.

- 1) 잡음환경속에서 통신 Error 정정을 위하여 Turbo code 기술이 적합함을 확인할 수 있었다.
- 2) Turbo coding Algorithm의 단점인 속도개선방안을 제안하였다.
- 3) 이 기술을 이용하여 Alter사의 VHDL 소프트웨어 및 칩을 이용하여 약 12Mbps급의 만족스러운 속도를 얻을 수 있었다.
 - 통신 Error의 감소
 - 복호속도의 2배 향상

본 기술을 바탕으로 선박의 모든 통신방식에 적용할 수 있을 것으로 생각되며, 무선 통신, 전력선통신 등에도 적용할 수 있을 것으로 기대된다.

후 기

본 연구는 산업자원부의 “산업기반기술개발”의 지원으로 수행한 연구결과의 일부이다. 산업자원부 및 산업기술평가원 관계자들에게 감사를 드립니다.

참고문헌

1. L. Bahl, J. Cocke, F. Jelinek and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate”, IEEE Trans. Inf. Theory, vol. IT-20, pp. 284-287, Mar. 1974.
2. J. Hagenauer, E. Offer and L. Papke, “Iterative decoding of binary block and convolutional codes”, IEEE Trans. Inf. Theory, vol. 42, no. 2, pp. 429-445, Mar. 1996.
3. C. Berrou, A. Glavieux and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes”, Proc. 1993 Int. Conf. Commun., pp. 1064-1070.
4. C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes”, IEEE Trans. Commun., vol. 44, no. 10, pp. 1261-1271, Oct. 1996.
5. G. Battail, “A conceptual framework for understanding turbo codes”, IEEE J. Select. Areas Commun., vol. 16, no. 2, pp. 245-254, Feb. 1998.
6. P. Robertson, E. Villebrun and P. Hoeher, “A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain”, Proc. 1999 Int. Conf. Commun., vol. 2, pp. 1009-1013.
7. W. Ryan, “A turbo code tutorial”, Unpublished paper.

8. M. C. Valenti, "An introduction to Turbo codes", Unpublished Report, May 1996.
9. 김현철, 이필중, "한 개의 오류정정능력을 갖는 BCH 부호를 연결한 터보코드의 성능에 관한 연구", 한국통신학회 하계종합학술대회 논문집, 상권, pp 816-819, July 1999.
10. 김 용, 이필중, "거리분포에 대한 모멘트에 따른 터보 코드의 인터리버 특성연구", 통신정보합동학술대회 논문집, 제1권, 4/99, pp.438-442
11. 유남열, 이필중, "터보부호의 반복부호를 위한 효율적인 정지기준", 통신정보합동학술대회 논문집, 제1권, 4/99, pp.46-50
12. Y.M. Choi and P.J. Lee, "An analysis of turbo codes with asymmetric modulation", Electronics Letters, Vol.35, no.1, 1/99, pp.35-36
13. 김 용, 이필중, "IMT2000에 적합한 다중 테일터보코드", 한국통신학회 추계종합학술발표회논문집, 11/98. 624-627
14. 김영대, 김 용, 김현철, 이필중 "Dummy 비트를 이용한 높은 SNR에서 터보코드 성능에 관한 연구", 한국통신학회 추계종합학술발표회 논문집, 11/98, 628-631
15. 최영민, 이필중, "비대칭 변조 터보코드의 해석", 한국통신학회 부호 및 정보이론 워크숍 논문집, 12/97
16. 김 용, 이필중, "FPLMTS에서 음성전송을 위한 turbo code개발" 통신정보합동학술대회 논문집, 제1권, 4/97, pp423-427
17. 박창수, 김 용, 이필중, "전송율 1/2인 Turbo Code에서의 최적의 구성부호", 제3차 부호및 정보 이론헤크숍 논문집, 12/96