

비터비 알고리즘을 이용한 $r=1/3$, $K=9$ 컨벌루션 복부호기의 설계

정희원 송문규*, 원희선**, 박주연*

Design of $r=1/3$, $K=9$ Convolutional Codec Using Viterbi Algorithm

Moon Kyou Song*, Hee Sun Won**, Ju Yeon Park* *Regular Members*

요약

채널의 영향으로 수신 데이터에서 발생한 에러를 정정할 수 있는 부호율 $r=1/3$ 이고 구속장 $K=9$ 인 컨벌루션 코덱 칩을 간단한 회로에 주안점을 두고 VLSI 설계한다. 복호기에서는 3비트 연성판정을 이용한 비터비 알고리즘이 사용된다. 정보 데이터의 정정과 저장을 위해서는 45단의 레지스터 교환 방식을 채택하였다. 회로의 설계시 VHDL 언어를 이용하였고, 회로의 시뮬레이션과 합성을 위해 Synopsys사의 Design Analyzer와 VHDL 시뮬레이터를 사용하였다. 이 칩은 ENCODER, ALIGN, BMC, ACS, SEL_MIN 및 REG_EXCH 블록으로 구성된다. 회로의 동작은 여러 가지 에러 상황을 가정하여 논리 시뮬레이션을 통해 검증하였고, 합성 후 타이밍 시뮬레이션 결과 325.5Kbps의 정보 데이터까지 부호 및 복호가 가능하였으며, 외부 메모리부를 제외하면 총 6,894 게이트가 소요되었다.

ABSTRACT

In this paper, a VLSI design of the convolutional codec chip of code rate $r=1/3$, and constraint length $K=9$ is presented, which is able to correct errors of the received data when transmitted data is corrupted in channels. The circuit design mainly aimed for simple implementation. In the decoder, Viterbi algorithm with 3-bit soft-decision is employed. For information sequence updating and storage, the register exchange method is employed, where the register length is $5 \times K$ (45 stages). The codec chip is designed using VHDL language and Design Analyzer and VHDL Simulator of Synopsys are used for simulation and synthesis. The chip is composed of ENCODER block, ALIGN block, BMC block, ACS block, SEL_MIN block and REG_EXCH block. The operation of the codec chip is verified though the logic simulations, where several error conditions are assumed. As a result of the timing simulation after synthesis, the decoding speed of 325.5Kbps is achieved, and 6,894 gates is used.

I. 서론

1990년대 이후 전 세계적으로 통신 서비스에 대한 요구가 폭발적으로 커지고 있으며 요구 서비스도 음성뿐만 아니라 오디오, 데이터, 화상, 영상 등

과 같은 멀티미디어 서비스를 시간과 장소의 제약 없이 언제, 어디서나, 누구에게도 제공할 수 있는 방향으로 전개되고 있다. 이에 따라서 열악한 채널 상황에서 데이터를 우수한 통신성능으로 전송하기 위해서 에러정정 부호의 적용이 고려되어 왔으며,

* 원광대학교 전기전자공학부(mksong@wonms.wonkwang.ac.kr)

** 한국방송공사 기술연구소(hswon@tri.kbs.co.kr)

논문번호 : 98439-1009, 접수일자 : 1998년 10월 9일

* 이 논문은 1998년도 원광대학교의 교비지원에 의해서 연구됨.

특히 전력 제한된 상황에서 에러정정 부호의 적용은 필수적이다. 데이터가 고속이고 채널이 열악할수록 보다 강력한 에러정정 기법이 요구된다. 이러한 에러정정을 위해서 현재 여러가지 우수한 기술이 개발되어 있으며 그 중에서 콘벌루션 부호는 여러 분야에서 가장 널리 표준으로 채택되고 있는 부호 중 한 가지이다. 본 논문에서는 $r=1/3$, $K=9$ 의 콘벌루션 부호를 이용하여 오류 정정이 가능한 복부호기 칩을 설계 및 합성하고 기능을 검증하고자 한다. 설계한 회로의 기능을 검증하기 위해 테스트 벤치를 이용하여 논리 시뮬레이션을 수행하고 합성하여 타이밍 시뮬레이션을 수행하고자 한다.

II. 콘벌루션 부호와 비터비 알고리즘

1. 콘벌루션 부호

콘벌루션 부호는 1955년 Elias에 의해 처음으로 소개되었는데, 정보계열을 k 비트씩 블록으로 분리하여 부호화가 수행되지만 과거의 정보 블록이 현재의 정보 블록에 영향을 미치는 메모리 기능을 가지고 있다^{[1],[2]}. 정보 블록은 M 단의 지연소자에 입력되고, 생성계열에 의해 선택된 각 지연소자의 출력을 모듈로-2 연산을 이용하여 부호화한 n 개의 출력을 전송로에 내보낸다. 이 부호를 (n, k) 콘벌루션 부호라 부른다. 한 개의 정보 블록이 직접 영향을 줄 수 있는 블록 수는 $M+1$ 개이다. 이를 구속장(constraint length)이라 하고, K 로 표시한다. 식 (1)은 k 가 1인 경우 구속장 K 와 지연소자의 수 M 의 관계식으로 지연소자의 수는 구속장 K 보다 1이 적다.

$$K = M + 1 \tag{1}$$

부호율(code rate) r 은 n 과 k 의 비로 식 (2)와 같이 표현할 수 있다.

$$r = k/n \tag{2}$$

일반적으로 k 비트의 정보 블록과 n 비트의 부호어는 $k < n$ 의 관계가 있다. 주어진 K 와 r 에 대해 최대의 최소자유거리(minimum free distance)를 갖는 콘벌루션 부호의 생성계열은 이미 알려져 있다^{[1],[3]}.

2. 비터비 알고리즘

비터비 알고리즘은 1967년 Viterbi에 의해 콘벌루

션 부호를 복호하기 위한 방법으로 고안되었다. 비터비 알고리즘은 격자도에서 수신어 r 과 가능한 부호어와 비교하여 해밍(hamming) 거리가 가장 작은 경로를 탐색하는 복호법이다^[4]. 즉, 가능성이 적은 경로를 포기하고 최우 경로를 탐색하는 데에 중점을 둔다. 비터비 복호기는 2^M 개의 생존경로를 기억하기 위한 기억소자(memory)와 생존경로와 수신계열과의 해밍 거리를 계산하여 기억하는 2^M 개의 기억소자, 그리고 생존 여부를 판정하는 비교기 등으로 구성된다^[1].

비터비 알고리즘의 기본적인 연산을 수행하는 ACS(Add-Compare-Select)의 동작은 다음과 같다^{[1],[3],[4]}. Add는 현재 시간의 지로 평가량과 이전 상태, 즉 현재 시간이 t 라면 $t-1$ 시간에서의 경로 평가량과 더하는 동작을 수행한다. Compare는 주어진 상태에 들어오는 경로 평가량을 비교하는 동작을 수행한다. Select는 두 개의 경로 평가량 중에서 가장 작은 것을 선택하고 다른 하나는 제거한다. 만약 두 개의 경로 평가량을 비교하여 서로 같다면 어느 것을 선택하여도 무방하다. 이러한 동작의 블록도는 그림 1에 보였다. 여기서 r_t 는 시간 t 에서 수신신호를 나타내며 $S_{j,t}$ 는 시간 t 에서 상태 S_j 이고 $V(S_{j,t})$ 는 $S_{j,t}$ 까지의 경로 평가량이며 $M_{i,j}(r_t)$ 는 수신신호 r_t 에 대한 S_i 에서 S_j 로의 지로 평가량을 나타낸다.

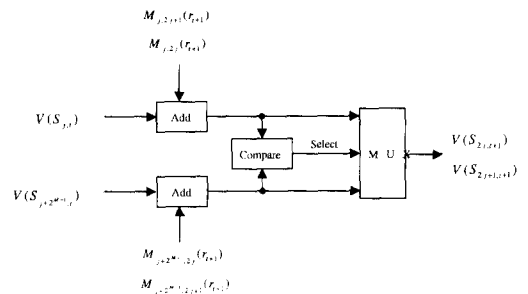


그림 1. $r=1/n$ 콘벌루션 부호에 대한 ACS

III. 콘벌루션 부호기 및 복호기의 설계

1. 부호기

정보계열을 부호화하는 장치로 부호율 $r=1/3$, 구속장 $K=9$ 의 콘벌루션 부호기를 이용한다. 주어진 부호율과 구속장에 대하여 최대의 최소자유거리를 갖는 콘벌루션 부호의 생성계열은 다음과 같다

[1],[3]

$$g_0^{(1)}=(111001001)=7118$$

$$g_1^{(2)}=(110110011)=6638$$

$$g_2^{(3)}=(101101111)=5578$$

위의 생성계열을 적용한 부호기의 구조는 그림 2와 같고, 모듈로 2 연산에 의해 부호화된 3개의 출력을 생성한다.

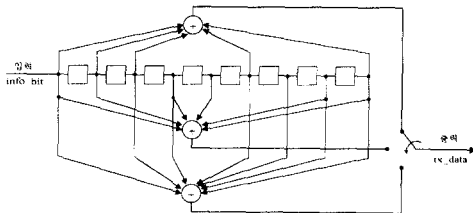


그림 2. $r=1/3, K=9$ 콘벌루션 부호기

부호기의 블록도 및 합성회로는 각각 그림 3과 그림 4에 나타내었다. 부호기는 부호기의 기본 클럭 $eclk$ 과 부호기 구동 신호 enc_ena 에 의해 구동되며 입력 데이터 $info_bit$ 를 부호화하여 부호화 데이터 tx_data 를 출력한다. 입력 데이터는 부호화 데이터율의 $n=3$ 배의 클럭 주기를 갖는다. 부호기에서는 부호기 구동 신호 dec_ena 와 부호화 데이터 tx_data 가 출력된다.

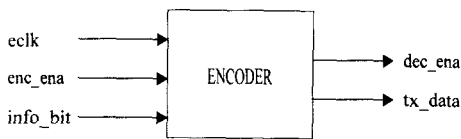


그림 3. 부호기의 구성도

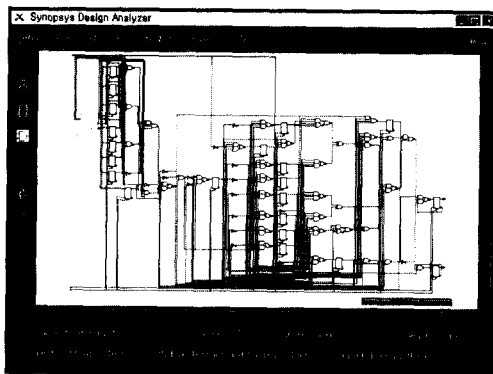


그림 4. 부호기의 합성회로

2. 복호기의 설계

복호기는 부호화 데이터를 수신하여 비터비 알고리즘을 적용해서 복호하는 것으로 전송 과정에서 잡음에 의해 훼손된 데이터를 정정하는 동작을 수행한다. 본 논문에서 사용하는 $r=1/3, K=9$ 인 콘벌루션 부호의 복호를 위한 격자 구조는 그림 5에 보였다. 비터비 알고리즘을 기술하는 전체 격자도는 그림 5의 격자 구조가 반복되는 형태로 표현된다.

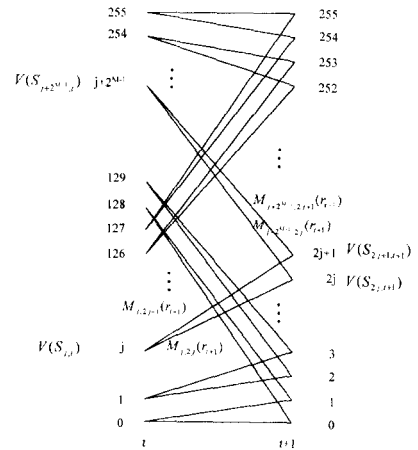


그림 5. $K=9$ 콘벌루션 부호의 격자 구조

비터비 복호기는 그림 6에 보인 바와 같이 ALIGN, BMC(Branch Metric Calculator), ACS(Add · Compare · Select), SEL_MIN(Select Minimum), REG_EXCH(Register Exchange) 등의 블럭으로 구성된다. 이러한 비터비 복호기의 설계는 먼저 C 프로그램을 이용하여 복호 기능이 동작함을 확인한 후 VHDL을 이용하여 기술하였다.

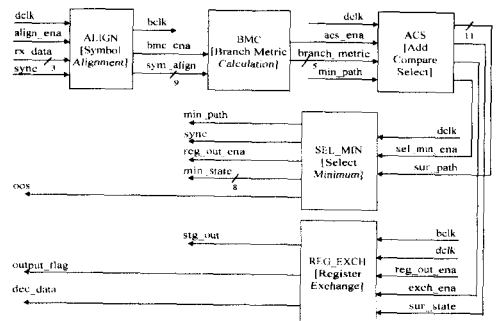


그림 6. 복호기의 구성도

가. ALIGN 블럭

부호화되어 전송된 데이터는 직렬 형태로 수신되어 연성판정 복호를 위해 3비트 연성판정된다. ALIGN 블록에서는 3개의 수신신호에 대한 3비트 연성판정 데이터 rx_data를 병렬로 변환하여 9비트의 sym_align 신호를 출력하는 역할을 수행한다. ALIGN 블록의 입력신호는 복호기 기본 클럭 dclk, ALIGN 블록의 구동 신호 align_ena, 수신신호에 대한 3비트 연성판정 신호 rx_data와 부호 블록 동기신호 sync로 구성된다. 출력 신호는 9비트의 병렬 변환신호 sym_align과 부호 블록 클럭 bclk, BMC블록의 구동 신호 bmc_ena로 구성된다. ALIGN 블록에서 출력되는 부호 블록 클럭 bclk의 주기는 수신 데이터의 주기의 3배이다. sync는 부호 블록의 시작을 맞추기 위한 동기 신호이다.

나. BMC 블록

본 논문에서는 복호기에서 8레벨의 3비트 연성판정(soft-decision) 방식으로 회로를 설계하였다. 이상적인 비터비 복호기는 무한한 정밀도를 가지는 실수의 처리가 가능한 것으로 가정하지만, 디지털 회로로 설계할 때 수신된 신호는 양자화되며, 이로 인해 비터비 복호기의 성능에 다소 손해를 보게 된다. 하지만 8레벨의 양자화로 인한 부호 이득의 손실은 매우 작은 것으로 알려져 있다¹¹⁾. BMC 블록은 지로 평가량(branch metric)을 계산하는 장치로 BMC 구동 신호 bmc_ena가 1로 설정되면 9비트의 입력 신호 sym_align에 대한 비트 평가량(bit metric)을 계산한다. 비트 평가량은 전송채널에 따라 결정되며, 본 논문에서는 DSC (discrete symmetric channel) 채널을 가정하고 회로의 복잡도를 줄이기 위해 전송 심볼 y에 대한 3비트 수신 심볼 r의 조건부 확률 p(r|y)에 대한 비트 평가량을 식 (3)에 의해 표 1과 같이 설정하였다.

$$a[\log_2 b(r|y) + b] \tag{3}$$

여기서 a와 b는 비트 평가량을 정수로 만들기 위한 계수이다.

표 1. 3비트 연성판정의 비트 평가량

	y	0	1	2	3	4	5	6	7
x									
0		0	1	2	3	4	5	6	7
1		7	6	5	4	3	2	1	0

BMC 블록의 입력신호는 복호기의 기본 클럭 신

호 dclk, BMC 구동 신호 bmc_ena와 ALIGN 블록의 출력인 sym_align이다. BMC 블록의 출력신호는 ACS를 구동하기 위한 신호 acs_ena와 지로 평가량 branch_metric 등으로 구성된다.

다. ACS 블록

ACS 블록은 비터비 알고리즘을 구현하는 기본 연산을 수행하는 장치이다. ACS 블록은 dclk, acs_ena, branch_metric과 min_path 등의 입력 신호와 sur_state, sur_path, exch_ena와 sel_min_ena 등의 출력신호를 갖는다. acs_ena는 ACS 블록의 구동 신호이며 branch_metric은 BMC 블록의 출력이다. min_path는 경로 평가량(path metric)을 주기적으로 감소시키기 위한 신호로 SEL_MIN 블록의 연산 결과에 의해 생성되는 신호이다. sur_state 신호는 256개의 생존 경로의 상태를 저장하는 레지스터이고 sur_path는 256개의 생존경로의 누적된 값을 저장하는 레지스터이다. exch_ena는 REG_EXCH 블록의 구동 신호이고 sel_min_ena는 SEL_MIN 블록의 구동 신호이다.

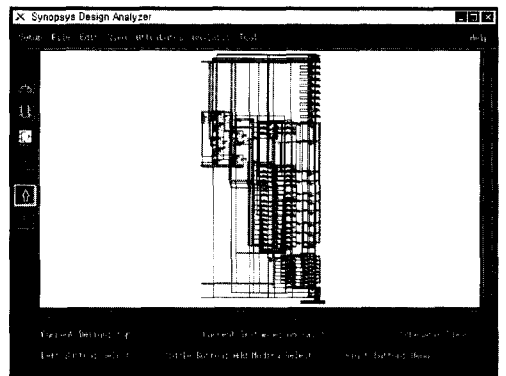


그림 7. ACS 블록의 합성회로

ACS 블록에서는 입력된 지로평가량을 비교하여 생존경로로 선택한 경로 평가량을 출력하고 REG_EXCH 블록과 SEL_MIN 구동 신호를 출력하는 기능을 수행한다. ACS 블록에서는 수신된 부호어에 대해 각 상태에서 결정된 2^M 즉 256개의 생존 경로에 대한 경로 평가량과 이에 해당하는 256개의 복호 정보가 만들어진다. 이때 선택된 생존 경로의 경로 평가량을 저장하는 레지스터의 크기는 한정되어 있으나 시간이 경과할수록 경로 평가량의 값은 계속 증가하기 때문에 이 값을 주기적으로 감소시킬 필요가 있다. 본 논문에서는 3비트 연성판정을 사용하므로 지로평가량의 최대값은 21이며, t=45단 까지 경로평가량은 최악의 경우 945일 것이다. 따라

서 $t=45$ 시간 이후부터 SEL_MIN 블럭의 연산에서 출력되는 최소의 경로 평가량으로 모든 경로 평가량의 값을 감소시킴으로써 경로 평가량의 값을 레지스터에서 저장할 수 있는 최대값 이하로 유지하도록 하였다. 그림 7은 ACS 블럭의 합성회로를 보인 것이다.

라. SEL_MIN 블럭

이 블럭은 SEL_MIN 구동 신호가 1로 설정되면 ACS 블럭에서 선택된 경로 평가량의 크기를 비교하여 가장 작은 값을 선택하는 동작을 수행한다. 입력신호는 기본 클럭 dclk, SEL_MIN 블럭의 구동 신호 sel_min_ena, 생존경로의 값을 저장하는 레지스터 sur_path이고, 출력신호는 경로 평가량을 감소시키기 위한 최소 평가량 min_path와 이에 해당하는 상태 min_state, REG_EXCH 블럭의 출력을 구동하기 위한 구동 신호 reg_out_ena, 부호 블럭 동기를 위한 동기 신호 oos 등으로 구성된다. SEL_MIN 블럭은 ACS 블럭의 출력인 sur_path의 크기를 비교하여 가장 작은 경로를 찾아내는 연산을 수행한다. 256개의 sur_path가 모두 입력된 이후에 가장 작은 경로의 값과 상태를 출력한다. oos는 블럭 동기가 연속해서 5번 맞지 않았을 경우 1을 출력한다. 이 경우 수신된 신호는 새로운 블럭 동기에 맞추어 복호된다.

마. REG_EXCH 블럭

비터비 복호기의 출력에서 복호화된 정보 데이터 시퀀스를 결정하고 저장하는 방법에는 역추적(trace back)과 레지스터 상호교환(register exchange)의 두 가지 방법이 있다. 역추적 방법은 정보 시퀀스를 결정하는 과정에서 레지스터에 저장된 내용이 상호교환될 필요가 없으므로 레지스터 상호교환 방법에 비해 상당히 빠르게 동작할 수 있지만 다소 복잡해지는 단점이 있다¹¹. 본 논문에서는 비교적 저속의 데이터의 응용을 고려하고 회로의 간략화에 주안점을 두어 레지스터 상호교환 방법을 이용하여 설계하였다. 고속의 데이터 응용에 대해서는 ACS를 병렬로 설계하여야 할 것이며 이 경우 역추적 방법이 선호된다.

REG_EXCH 블럭은 ACS의 비교 동작에 의해 선택된 값을 이용하여 상태에 입력되는 정보신호를 결정하는 동작을 수행한다. REG_EXCH 블럭에서 사용하는 레지스터의 길이는 일반적으로 복호 깊이(decoding depth)의 k 배이고, 복호 깊이는 요구되는 BER 성능에 따라 일반적으로 구속장의 5~10배정도로 한다¹¹. 본 논문에서는 레지스터의 길이를 구

속장의 5배인 45단으로 설정하였다. 참고로 BER= 10^{-4} 에서 부호이득 5.0dB를 얻기 위해서는 3비트 연성판정의 경우 복호 깊이를 63이상으로 설정하여야 할 것이다⁵.

REG_EXCH 블럭의 입력신호는 클럭 dclk, REG_EXCH 블럭의 구동 신호 exch_ena, 생존경로의 상태가 저장된 레지스터 sur_state, REG_EXCH 블럭의 출력 구동 신호 reg_out_ena와 가장 작은 경로의 상태값 min_state 등으로 구성되고, 출력신호는 레지스터 교환 방식에 의해 결정된 45비트의 정보 신호를 저장한 레지스터 stg_reg와 reg_out_ena 신호가 1이 되는 순간부터 복호된 정보를 출력하는 dec_out 신호와 복호 정보의 출력이 시작됨을 표시하는 output_flag 등으로 구성된다.

IV. 타이밍 시뮬레이션

본 장에서는 III장에서 설명한 복부호기의 각 블럭에 대한 타이밍 시뮬레이션 결과를 보이고자 한다. 부호기와 복호기 설계를 위해 설계 언어로는 VHDL을 사용하였고, 프로그램의 설계와 합성을 위해 Synopsys사의 Design Analyzer 및 VHDL Simulator를 이용하였다. 설계한 회로의 기능을 검증하기 위해 테스트 벤치를 이용하여 논리 시뮬레이션을 수행하고 합성하여 타이밍 시뮬레이션을 통해 동작을 검증하였다. 부호기와 복호기에 입력되는 기본 클럭 주기는 12ns로 설정하였으며, 이 경우 정보 데이터의 주기는 결국 3,072ns로서 325.5kbps이고 부호화 데이터율은 976.6 kbps가 된다. 타이밍 시뮬레이션 결과 약 145,920ns 이후부터 복호된 정보 신호가 출력되었다. 출력된 정보 데이터는 테스트 벤치에서 입력한 정보 데이터를 정확하게 복호함을 확인할 수 있었다. 설계한 복부호기 회로는 부호기 기본 클럭 eclk, 부호기 구동 신호 enc_ena, 정보 데이터 info_bit, 복호기 기본 클럭 dclk, 복호기 구동 신호 dec_ena, 수신 신호에 대한 연성판정 신호 rx_data 등의 입력 신호와 출력 개시 신호 output_flag, 복호된 정보 데이터 신호 dec_data, 부호 블럭 동기 신호 oos 등의 출력 신호를 갖는다.

1. 부호기

그림 8은 부호기의 출력 파형을 나타낸 것이다. 한 개의 입력 정보 신호가 유지되는 시간인 3,072ns 동안 3개의 부호화된 신호를 출력한다.

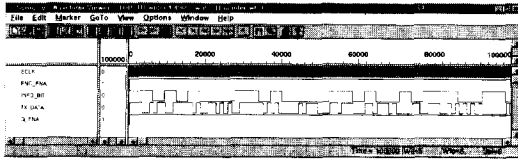


그림 8. 부호기의 시물레이션 파형

2. ALIGN 블럭

그림 9는 ALIGN 블럭의 시물레이션 파형을 나타낸 것이다. 또한 REG_EXCH 블럭에서 사용할 클럭 bclk를 발생하게 된다. 이 클럭의 주기는 수신된 부호 데이터 주기의 3배로서 3,072ns이다. 클럭 bclk의 발생과 함께 bmc_ena도 1로 설정됨을 볼 수 있다.

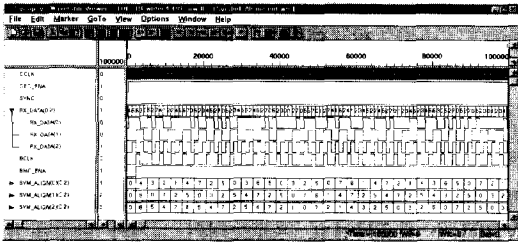


그림 9. ALIGN 블럭의 시물레이션 파형

3. BMC 블럭

BMC 블럭은 입력신호에 대한 256개의 지로 평가량을 계산하고 이 결과는 ACS 블럭으로 입력된다. 지로 평가량을 계산하면 ACS 블럭의 구동 신호인 acs_ena를 1로 설정한다.

4. ACS 블럭

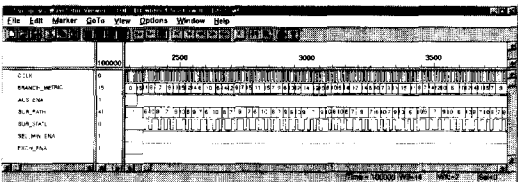


그림 10. ACS 블럭의 시물레이션 파형

그림 10은 ACS의 계산 결과를 출력한 것이다. REG_EXCH와 SEL_MIN의 입력으로 사용할 sur_state와 sur_path를 출력하고, 처음 ACS 블럭이 구동된 후에 exch_ena와 sel_min_ena를 1로 설정한다. ACS 블럭에서는 2개의 클럭이 소요되며, 직렬 방식으로 설계함에 따라 복호기에서 비터비 알고리즘의 적용시 격자도의 모든 128개의 상태에 대해

하나의 ACS 블럭을 128회 사용하게 된다. 따라서 매 ACS 연산에는 기본 클럭 12ns의 256배인 3,072 ns가 소요되며, 여기에서 복호기의 최대 복호율이 결정된다. 따라서 ACS 블럭을 병렬로 설계한다면 회로는 복잡해지나 복호율을 향상시킬 수 있다.

5. SEL_MIN 블럭

SEL_MIN 블럭은 sel_min_ena 신호가 1로 설정 되면 동작을 시작하여 256개의 sur_path 중에서 최소의 경로 평가량 min_path와 그 상태 min_state를 선택하고 이 값을 REG_EXCH 블럭에 입력한다.

6. REG_EXCH 블럭

그림 11은 REG_EXCH 블럭의 출력 파형을 나타낸 것이다. 이 블럭은 ACS 출력인 sur_state 값을 이용하여 45비트 길이의 레지스터를 만든다. SEL_MIN 블럭의 출력인 reg_out_ena 신호가 1이 되면 min_state를 읽어들이 복호한 정보 데이터를 출력한다. 복호된 정보 데이터 신호인 dec_out은 145,920ns에서부터 출력이 시작된다.

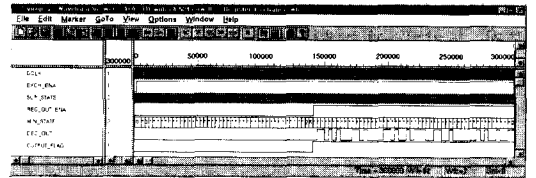


그림 11. REG_EXCH 블럭의 시물레이션 파형

7. 복호기 출력

그림 12는 복호기의 출력 파형을 나타낸 것이다. 입력 정보인 info_bit에 대한 부호화된 정보 tx_data가 있고, 복호기를 이용하여 오류를 정정한 정보인 dec_out 신호의 출력을 나타내었다. 복호된 정보의 출력은 145,920ns에서 최초의 출력이 발생하고 3,072ns 간격으로 복호한 결과값을 출력한다.

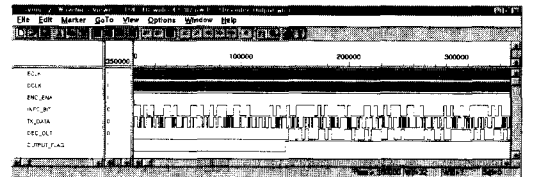


그림 12. 복호기의 시물레이션 파형

V. 결론

본 논문에서 전송로 상에서 발생하는 오류를 검출하여 복구할 수 있는 콘벌루션 부호기와 비터비 알고리즘을 이용한 복호기를 설계하였다. 복호기는 비터비 알고리즘을 이용하여 설계할 것이며 간략한 회로에 주안점을 둔 설계를 시도하였다. 복부호기 칩은 부호율 $r=1/3$, 구속장 $K=9$ 이고, 생성 계열(generator sequence)이 $(711, 663, 557)_8$ 인 콘벌루션 부호를 사용하였다. 또한 3비트 연성관정 비터비 알고리즘을 이용하여 구현하였고, 선택된 정보를 저장하는 방법으로 register exchange 알고리즘을 사용하였다. REG_EXCH 블록에서 사용하는 레지스터의 길이는 45단으로 설정하였다.

ACS 연산에 의해 선택된 경로 평가량은 시간이 경과함에 따라 증가하게 된다. 복호기에서 생존 경로의 평가량을 저장하는 레지스터의 크기는 유한하기 때문에 SEL_MIN 블록에서 결정된 최소 경로 평가량을 이용하여 ACS 연산 시 경로 평가량의 크기를 감소시킴으로써 경로 평가량이 레지스터의 범위를 초과하지 않도록 하였다. 복호기에 연속적으로 입력되는 정보 계열의 블록 동기를 위해 SEL_MIN 블록의 동기 신호를 ALIGN 블록에 인가하여 블록 동기를 맞추는데 사용하도록 함으로써 복호 오류가 발생하는 것을 해결하였다.

설계한 복부호기 칩은 ENCODER, ALIGN (Symbol Alignment), BMC(Branch Metric Calculator), ACS(Add·Compare·Select), SEL_MIN(Select Minimum path & state), REG_EXCH (Register Exchange) 등의 블록으로 구성되어 있다. 설계한 칩의 검증은 위해 회로 내부의 부호기를 이용하였다. 테스트 벤치를 이용하여 시뮬레이션을 수행한 결과 에러를 복구하고 원래의 정보신호를 출력함을 알 수 있었다. 이러한 회로의 설계와 합성을 위해 Synopsys의 Design Analyzer 및 VHDL Simulator를 주로 이용하였고 언어로는 VHDL을 이용하였다. 복부호기 칩을 구현하는 회로의 게이트의 수는 경로 평가량과 register exchange의 계산 결과를 저장하는 메모리인 외부의 RAM 부분을 제외하면 6,894 게이트였다. 설계한 칩의 시뮬레이션 결과 복호기의 정보 데이터율은 최대 325.5kbps이지만, 이는 비터비 알고리즘의 기본이 되는 ACS 블록을 직렬로 설계한 결과이다. ACS 블록을 병렬로 설계한다면 전송률을 최대 128배까지 향상시킬 수 있으며 이 경우 비터비 복호기의 출력단에서 역추적 방법의 적용이 바람직할 것이다.

참 고 문 헌

- [1] Stephen B. Wicker, Error Control Systems For Digital Communication And Storage, Prentice Hall, Inc., 1995.
- [2] The Forward Error Correction Handbook, Fourth Edition, Stanford Telecom, 1996
- [3] Andrew J. Viterbi, CDMA Principles of Spread Spectrum Communication, Addison Wesley, 1995.
- [4] G.D. Forney, JR., "The Viterbi Algorithm", Proceedings of the IEEE, Vol. 61, pp. 268 ~ 278, March 1973.
- [5] 차진중, 현진일, 강인, 김재석, 김경수, "CDMA 이동통신 시스템용 비터비 복호기 ASIC 설계 및 구현", 전자공학회논문지 제33권 A편 제1호 1996. 1.

송 문 규(Moon Kyou Song) 정회원
 1988년 2월 : 고려대학교 전자공학과 졸업
 1990년 2월 : 고려대학교 전자공학과 공학석사
 1994년 2월 : 고려대학교 전자공학과 공학박사
 1997년 10월 ~ 1998년 12월 : 한국전자통신연구원 초빙연구원
 1994년 3월 ~ 현재 : 원광대학교 전기전자공학부 조교수
 e-mail: mksong@wonnms.wonkwang.ac.kr

원 희 선(Hee-Sun Won) 정회원
 1990년 2월 : 연세대학교 전산학과 졸업
 1992년 2월 : 한국과학기술원 전산학과 공학석사
 1992년 3월 ~ 현재 : 한국방송공사 기술연구소 연구원
 e-mail: hswon@tri.kbs.co.kr

박 주 연(Ju-Yeon Park) 준회원
 1996년 2월 : 원광대학교 제어계측공학과공학과 졸업
 1996년 8월 : 원광대학교 제어계측공학과 공학석사