

효율적인 채널 정보 메모리 관리를 위한 PCM 인코더 설계

Design of Advanced PCM Encoder Architecture for Efficient Channel Information Memory Management

노윤희·김건희·김동영·김복기*·이남식
단암시스템즈 기술연구소

Yun-Hee Ro · Geon-Hee Kim · Dong-Young Kim · Bok-Ki Kim* · Nam-Sik Lee
R&D Center, DANAM Systems, Anyang, 13930, Korea

[요 약]

원격측정 시스템은 비행체의 상태 정보 데이터를 획득하여 지상 수신소로 전송하는 시스템이다. PCM 인코더는 획득한 상태 정보 데이터를 이용하여 프레임 생성하기 위해 채널 정보를 저장할 메모리가 필요하다. 대형 비행체의 경우 각 부위의 센서 및 시스템이 증가함에 따라 많은 데이터를 계측해야 하므로 더 큰 용량의 메모리를 필요로 한다. 그러나 한정된 메모리에서 모든 채널 정보를 저장하기에는 어려움이 있다. 따라서 본 논문에서는 FPGA 내부 ROM의 한정된 메모리에서 중복되는 채널 정보는 메모리에 한번만 할당하고, 마이너 프레임마다 다른 정보를 가지는 서브콤의 채널 정보는 서브콤 배수만큼 메모리에 할당하는 PCM 인코더 구조를 제시하고 구현하였다. 이는 중복으로 할당되는 채널 정보를 최소화하여 효율적으로 메모리를 관리할 수 있다. 또한 다양한 계측 주기의 채널을 구성한 시뮬레이션을 통해 제시한 PCM 인코더의 동작을 검증하였다.

[Abstract]

Telemetry system is a system that transmits status information data acquired from the aircraft to the ground station. PCM encoder needs memory to store channel information in order to generate a frame format using the acquired data. Generally, telemetry systems in large aircraft require much larger memory for the increased acquisition channel information due to the increased sensors and subsystems. However, they have difficulty to store all channel information in limited memory. In this paper, we suggests and implements an advanced PCM encoder that can efficiently manage memory by minimizing duplicated channel information. This novel PCM encoder allocates duplicated channel information to memory only once. And, sub commutation channels having different information for each minor frame are allocated to the memory by multiples of sub commutation channels. Finally, the suggested PCM encoder was proved by simulation that composed channels of various measurement cycles.

Key word : Telemetry system, Sub commutation, ROM, FPGA, VBA.

<https://doi.org/10.12673/jant.2020.24.4.305>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 15 July 2020; Revised 23 July 2020

Accepted (Publication) 20 August 2020 (30 August 2020)

*Corresponding Author: Bok-Ki Kim

Tel: +82-31-538-6008

E-mail: bokki@danam.co.kr

I. 서론

원격측정 시스템은 비행 시험 중인 비행체의 상태 정보를 실시간으로 분석하기 위해 지상의 수신소로 전송하는 시스템으로 국내에서는 1980년대부터 PCM (pulse code modulation) / FM (frequency modulation) 방식으로 개발되어왔다. 비행체 내부의 상태를 센서 및 주변 장치들을 통해 계측하고, PCM 엔코더에서 데이터를 취득하여 IRIG-106(inter-range instrumentation group-106)의 메이저 프레임 형식으로 구성한다. 이렇게 구성된 프레임은 송신기를 통해 지상 수신소로 전송되고, 점검 프로그램 등을 통해 데이터를 실시간으로 분석한다. 원격측정 시스템에서 획득한 데이터는 비행체의 성능을 점검하는 데이터이므로 비행 중에 문제가 발생하면 원인을 분석할 때 필요하다.

PCM 엔코더는 획득한 데이터들을 분석이 가능한 프레임 형태로 구성하고 전송하는 역할을 수행하므로 원격측정 시스템에서 중요한 부분을 담당한다. 일반적으로 PCM 엔코더는 계측 채널에 대한 정보를 EEPROM (electrically erasable programmable read-only memory)을 이용하여 저장하거나 FPGA (field programmable gate array) 내부의 ROM(read-only memory)을 이용하여 채널 정보를 저장하는 구조이다[1]-[3]. 대형 비행체의 경우, 각 부위의 센서 및 시스템에서 상태 정보를 획득하는 분산 계측 장치들이 추가되면서 많은 데이터들이 계측된다. 때문에 최근 원격측정 시스템의 정밀도 향상과 더불어 PCM 엔코더에서 프레임을 구성하는 채널의 수가 점차 증가하고 있다. 하지만 일반적인 엔코더에서는 계측하는 채널이 많아지거나 계측 주기가 다른 채널 정보를 구성하는데 유연성이 떨어진다. 현재 다수의 마이너 프레임으로 메이저 프레임을 구성할 경우 각각의 마이너 프레임에 대한 채널 정보를 모두 저장하기 때문에 PCM 엔코더는 메모리를 비효율적으로 사용한다.

이에 대해 PCM 엔코더의 프레임을 구성하는 채널 정보를 UART (universal asynchronous receiver-transmitter)를 통해 실시간으로 PC에서 제어하고 flash 메모리를 통해 저장 및 업데이트가 가능한 엔코더들이 개발되어 왔다[4]. 하지만 메이저 프레임을 구성할 때 다수의 마이너 프레임 채널 정보를 효율적으로 저장하는 방법과 마이너 프레임마다 계측 주기가 다른 채널 정보를 가진 서브콤 채널에 대한 구현이 고려되어 있지 않다.

본 논문에서는 프로그램이 가능한 ROM을 이용하여 다수의 마이너 프레임으로 메이저 프레임을 구성할 때 마이너 프레임의 중복된 채널을 최소화하여 효율적으로 메모리에서 사용이 가능하도록 채널 정보를 구성하고, 서로 다른 계측 주기를 갖는 서브콤 채널 구현이 가능한 구조를 제안한다.

본 논문의 제 II장에서는 효율적인 채널 정보 메모리관리를 위한 ROM 채널 정보 구조를 제안하고, 이를 적용한 PCM 엔코더의 하드웨어적인 구조와 채널 정보를 입력하는 엑셀 VBA(visual basic for application) 기반 프로그램의 소프트웨어 구조를 기술하며, 제 III장의 시뮬레이션을 통해 제안한 구조의 타당성과 성능을 확인하였다.

II. PCM 엔코더 구현을 위한 구조 설계

본 논문에서는 프로그램이 가능한 FPGA 내부의 ROM을 이용하여 채널 정보를 저장하고 저장한 채널 정보를 이용하여 프레임의 구성하는 PCM 엔코더를 구현하기 위한 ROM 채널 워드 구조와 하드웨어 및 소프트웨어 구조를 기술한다.

2-1 ROM 채널 워드 구조

채널 정보는 원격측정 장치에서 계측할 비행 데이터들의 채널 타입, 계측 주기, 데이터들을 획득할 주소 값 등의 채널 속성 정보들이 포함되어있다[5]. 이러한 채널 정보들은 원격측정 장치들의 요구 조건에 따라 다양하게 설정이 가능하여 하드웨어 변경 없이 여러 규격의 프레임을 구성할 수 있다.

일반적인 PCM 엔코더는 메이저 프레임을 구성할 때 마이너 프레임 단위의 중복된 채널 정보를 구성하는 구조이다[6]. 이러한 구조는 중복되는 채널 정보로 인해 메모리 사용에 비효율적이다. 때문에 메이저 프레임을 구성할 때 채널이 중복되는 노말콤, 슈퍼콤의 채널 정보는 메모리에 한번만 할당하고, 마이너 프레임마다 다른 정보를 가지는 서브콤의 채널 정보는 서브콤 배수만큼 메모리에 할당하는 구조의 PCM 엔코더를 제안한다. 이는 한 개의 마이너 프레임에 대한 채널 정보들과 서브콤 배수별 서브콤 채널 정보를 보관할 메모리만 사용하므로 중복으로 할당되는 채널 정보를 최소화하여 메모리를 효율적으로 관리할 수 있다.

본 논문에서 제안한 채널 정보는 FPGA 내부의 ROM에 저장되어 PCM 엔코더의 클럭 주기에 맞춰 한 채널씩 증가된 주소에서 로드되어 프레임을 구성하기 위한 정보로 사용된다. 채널 정보에는 서브콤을 구현할 수 있도록 서브콤 정보와 프레임을 구성할 데이터 형태, 데이터를 읽어올 주소 값 등이 존재하며 구조는 그림 1과 같다.

채널 정보가 구성된 ROM 테이블의 첫 번째 주소에는 메이저 프레임의 길이 정보가 담긴 헤더가 존재한다. 헤더 정보는 메이저 프레임을 구성할 때 필요한 정보로 메이저 프레임의 길이 정보에 따라 서브콤의 최대 배수가 결정된다. 메이저 프레임의 길이는 IRIG-106 표준에 따라 최대 1024까지 설정이 가능하도록 구현하였다.

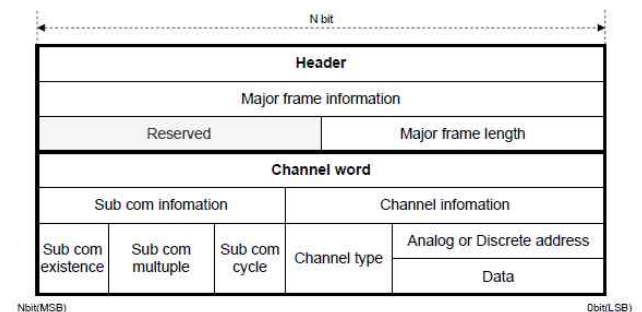


그림 1. ROM 채널 워드 구조
Fig. 1. Structure of ROM channel word

표 1. ROM 채널 정보 예시

Table 1. Example of ROM channel information

Information	Length	Data	Contents
Header	10bit	1~1024	Major frame length
Sub com existence	1bit	0 or 1	Exist or not sub com
Sub com multiple	4bit	0~15	Multiple of sub com
Sub com cycle	1bit	0 or 1	Exist or not cycle of sub com
Channel type	3bit	0b.000	Sync code
		0b.001	Major id(frame count)
		0b.010	Minor id
		0b.011	CRC
		0b.100	Analog data
		0b.101	Discrete data
		0b.110	Serial data
		0b.111	Empty channel
Data or Address	8bit	0~255	Analog MUX address
	5bit	0~31	Discrete MUX address
	16bit	0~65535	Specific Data

두 번째 주소부터는 프레임 구성할 채널 정보가 존재한다. N비트로 구성된 ROM 채널 워드의 상위비트에는 서브콤 정보가 위치하는데 이 정보는 해당 채널이 서브콤인지를 나타낸다. 다음 비트부터 순차적으로 서브콤의 배수 정보, 서브콤의 사이클 존재 여부에 대해 구성된다. 다음은 획득할 데이터에 따라 채널의 종류를 설정하는 비트이다. 마지막으로 주소 또는 데이터 값을 구성하는 비트로 채널 타입 설정 비트에 따라 연동된다. 채널 타입 설정 비트가 아날로그 또는 디스크리트이면 주소가 구성되고, 동기 채널 또는 특정 채널이면 동기코드나 해당 데이터의 값이 구성된다.

본 논문에서 성능 검증을 위하여 25비트의 ROM 워드 구조를 제안하였으며 표 1과 같이 ROM 채널 정보를 구성한다.

2-2 하드웨어 구조

본 논문에서 PCM 엔코더는 FPGA를 이용하여 프레임을 구성하는 로직으로 구현하였으며 PCM 엔코더의 로직 구조도는 그림 2와 같다.

PCM 엔코더의 로직은 프레임을 생성하는 PCM ENC 로직과 채널 정보를 가지고 있는 FPGA 내부 ROM과 시스템에서 사용하는 클럭을 생성하는 PLL (phase locked loop)로 구성된다. PCM ENC 로직은 라이징 클럭에 맞춰 ROM에 저장된 채널 정보를 읽어 프레임을 구성하게 된다. 또한 ROM에서 읽어 들인 채널 정보를 통해 서브콤 여부와 채널 타입을 확인한다. 서브콤이 존재하면 서브콤 처리를 통해 서브콤을 구성한다. 이때 채널 타입이 아날로그, 디스크리트, 시리얼일 경우 획득 로직을 통해 데이터를 획득하고 데이터 타입에 따라 레지스터에 저장된다.

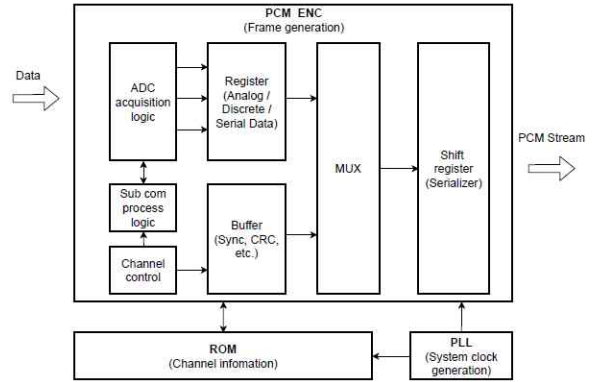


그림 2. PCM 엔코더 로직 구조
Fig. 2. Structure of PCM encoder logic

채널 타입이 sync 및 특정 타입이면 채널 정보의 데이터 값을 버퍼에 저장하고, CRC(cycle redundancy check)이면 데이터들의 CRC를 연산하여 버퍼에 저장한다. 이렇게 저장된 데이터들은 PLL에서 생성한 전송 클럭에 맞춰 MUX (multiplexer)를 통해 한 비트씩 쉬프트되어 PCM 스트림으로 출력된다.

ROM에 저장된 채널 정보를 읽어 들여 프레임 구성하는 순서는 그림 3과 같다.

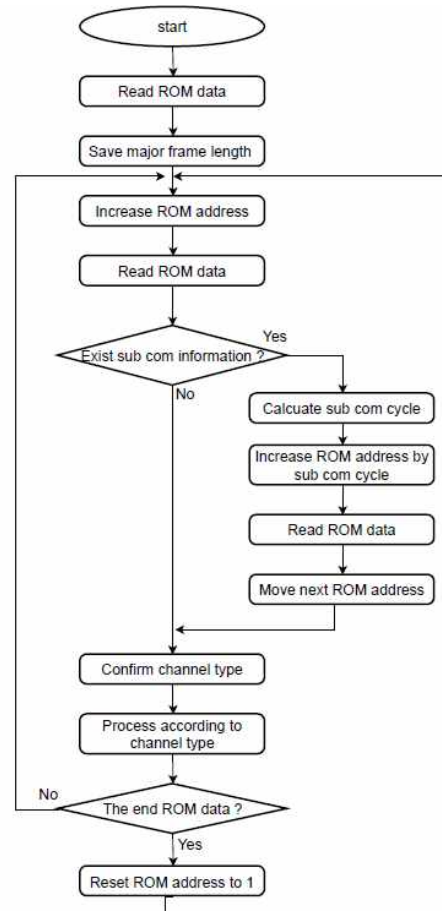


그림 3. 프레임 구성 순서도
Fig. 3. Flow chart of create frame

PCM ENC 로직에서 첫 번째 ROM 주소의 채널 정보를 읽어 온다. 첫 번째 주소에 저장된 헤더에서 메이저 프레임의 크기를 저장한 후 ROM의 주소를 1증가 시킨다. ROM의 채널 정보를 읽은 후 서브콤 정보의 존재 여부를 판단한다. 서브콤 정보가 존재하면 마이너 프레임 카운트와 서브콤 배수를 이용하여 서브콤의 사이클을 계산한다. 계산된 서브콤 사이클만큼 ROM 주소를 증가시킨 후 채널 정보를 ROM에서 읽어오고, 다음 채널 정보를 읽어올 ROM 주소로 이동시킨다. 읽어 들인 채널 정보에서 채널 타입을 확인하고 채널 타입에 따라 프레임을 구성한다. 채널 정보가 채널의 끝을 의미하는 "0x1FFFFFF"이면 ROM 주소를 1로 초기화하고 아니면 다음 ROM 주소의 채널 정보를 읽어오면서 프레임을 구성한다.

2-3 소프트웨어 구조

본 논문에서 FPGA 내부 ROM의 한정된 메모리 내에서 PCM 프레임을 효율적으로 구성하기 위한 엑셀 VBA 기반 프로그램을 그림 4와 같이 구현하였다.

해당 프로그램을 이용하여 사용자가 프레임 정보와 채널 구성 정보를 입력하면 해당 정보로 ROM 테이블을 생성한다. 또한 사용자가 확인할 수 있는 채널 배치표를 자동으로 생성하여 화면에 보여준다. 생성된 ROM 테이블은 ROM에서 읽어 들일 수 있는 MIF 파일 형식으로 저장하도록 구현하였다.

프레임 정보에는 메이저 프레임 크기, 마이너 프레임 크기와 ROM 크기가 있다. 이는 구성할 프레임의 메이저 프레임 크기와 마이너 프레임 크기, ROM 테이블을 저장할 ROM의 크기를 의미한다. 채널 구성 정보에는 채널 이름, MUX 번호, MUX 채널, 채널 번호, 수퍼콤 배수, 서브콤 배수, 서브콤 사이클 번호, 채널 종류, 데이터가 있다. 이러한 채널 구성 정보 중 MUX 번호와 MUX 채널은 아날로그 신호를 획득하기 위한 정보이다. 수퍼콤 배수는 1배~128배, 서브콤 배수는 1~1/16배로 설정이 가능하고 서브콤 배수는 메이저 프레임 크기보다 작거나 같게 설정해야 한다. 서브콤 사이클 번호는 메이저 프레임 구성 시 서브콤 채널을 할당할 마이너 프레임의 순서를 의미하며 0부터 서브콤 배수보다 1 작은 값까지 입력이 가능하다. 채널 종류는 표 1의 채널 타입을 구분하기 위한 정보이다. 프레임 내 서브콤 채널을 구성 시 마이너 아이디 타입의 채널을 반드시 구성해야 한다. 데이터는 고정된 값이거나 아날로그 또는 디스크리트의 주소값을 의미한다.

Channel Configuration Information											Frame Information		
Channel Information			MUX Information		Communication Information						Major frame Length	Minor frame Length	ROM Size
No.	Channel Name	Channel No.	Channel Type	MUX No.	MUX Channel No.	Super com Multiple	Sub com Multiple	Sub com Cycle No.	Value		1024	1024	
1	subcom1_1	2	A	0	1	1	2	0					
2	subcom1_2	2	S				2	1					
3	subcom1_1	4	A	1	1	1	4	0					
4	subcom2_2	2	S				4	1					
5	subcom2_3	4	D				4	2	0				
6	subcom2_4	4	A	2	1	1	4	3					
7	subcom3_1	6	A	3	1	1	8	0					
8	subcom3_2	6	S				8	1					
9	subcom3_3	6	D				8	2	0				
10	subcom3_4	6	A	4	1	1	8	3					
11	subcom3_5	6	S				8	4					
12	subcom3_6	6	D				8	5	0				
13	subcom3_7	6	A	5	1	1	8	6					
14	subcom3_8	6	S				8	7					
15	subcom4_1	8	A	6	1	1	16	0					
16	subcom4_2	8	S				16	1					

그림 4. 엑셀 VBA 기반 프로그램
Fig. 4. Program based on excel VBA

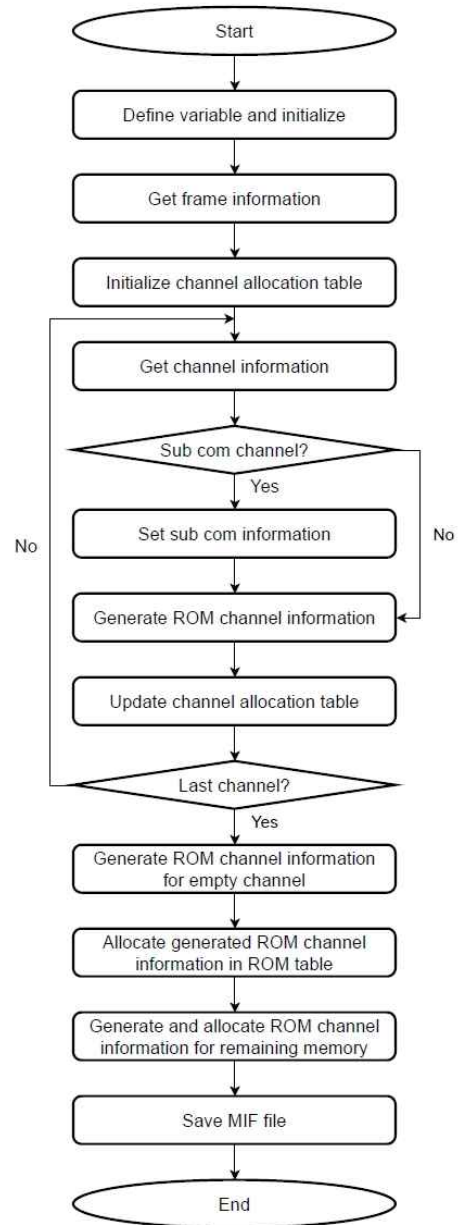


그림 5. ROM 테이블 생성 순서도
Fig. 5. Flow chart of generate ROM table

ROM 테이블을 생성하는 순서도는 그림 5와 같다. 프로그램을 실행하면 ROM 테이블을 구성하기 위한 변수들을 정의하여 초기화하고, 입력된 프레임 정보를 이용하여 채널 배치표를 초기화한다. 다음으로 사용자가 입력한 첫 번째 채널 구성 정보를 가져와 서브콤 배수가 1보다 큰지 비교하여 서브콤 채널인지 확인한다. 서브콤 채널인 경우 서브콤 여부, 서브콤 배수 및 서브콤 사이클 존재 여부를 포함하는 서브콤 정보를 설정한다. 서브콤 채널이 아닌 경우는 서브콤 정보를 설정하는 과정을 생략한다. 이후 ROM 채널 정보를 표 1과 같은 구조로 생성한다. 아날로그 채널 정보는 MUX 번호와 MUX 채널을 통

해 자동으로 계산한 아날로그 주소값으로 생성하고, 디스크리트 채널 정보는 데이터에 입력한 디스크리트 주소값으로 생성한다. 그 외 채널 정보는 고정된 값으로 생성한다. ROM 채널 정보가 생성되면 채널 배치표에 해당 채널을 업데이트한다. 사용자가 입력한 마지막 채널까지 ROM 채널 정보를 생성하고 채널 배치표를 업데이트하는 과정을 반복한다. 반복이 끝나면 마이너 프레임 크기 내에 사용자가 구성하지 않은 채널의 정보는 빈 채널을 의미하는 "0x00733CC" 값으로 생성한다.

이렇게 생성한 ROM 채널 정보들로 ROM 테이블을 생성하는데 첫 번째 주소에는 메이저 프레임의 길이를 포함하는 헤더를 할당하고, 두 번째 주소부터 마이너 프레임의 채널 구성 정보들을 순차적으로 할당한다. ROM 테이블에 마이너 프레임의 채널 정보를 할당할 때 해당 채널이 서브콤일 경우 마이너 프레임의 서브콤 사이클에 대한 서브콤 채널 정보를 연속적으로 할당하게 된다. 서브콤 사이클이 존재하는 경우 표 1의 서브콤 사이클 정보 값이 1인 채널 정보를 할당하고, 서브콤 사이클이 존재하지 않는 경우는 서브콤 사이클 정보 값이 0인 채널 정보를 할당한다.

이러한 순서로 마이너 프레임 크기 내에 채널 구성 정보를 할당한 메모리 외의 잔여 메모리에는 프레임의 끝을 의미하는 "0x1FFFFFF" 값을 생성하여 할당한다. 이렇게 생성된 ROM 테이블을 MIF 파일로 저장한다.

III. PCM 엔코더의 동작 검증 시뮬레이션

구현한 PCM 엔코더의 동작 검증을 위해 DE10-Lite Kit를 이용하여 시뮬레이션을 수행하였다. 시뮬레이션 결과는 Intel FPGA사에서 제공하는 시그널 탭을 이용하여 확인하였다.

3-1 시뮬레이션 설정

PCM 엔코더의 동작 검증을 위한 시뮬레이션 설정은 표 2와 같이 ROM 채널 정보를 구성하여 진행하였다.

표 1의 ROM 채널 정보에서 서브콤 배수를 설정하는 비트는 4비트로 최대 1/16배 서브콤 설정이 가능하다. 때문에 마이너 프레임에 1/2 ~ 1/16배 서브콤을 4채널 할당하여 확인하였다. 마이너 프레임에 서브콤 채널 정보가 제대로 할당되어있는지 검증하기 위해 채널 타입을 다양하게 설정하여 마이너 아이디에 따라 출력되는 채널 타입을 확인하였다.

또한 프레임 정보는 최대 1/16배 서브콤이 존재하므로 메이저 프레임의 크기를 16으로 설정하고, PCM 엔코더의 할당된 ROM 크기를 1024 채널 워드(25비트)로 설정하여 ROM 테이블을 구성하였다.

표 2. 시뮬레이션의 채널 정보

Table 2. Channel information for simulation

Contents	type	Information
Major frame length	Header	16
Sub com(x1/2)	Analog	1 Channel
	Serial	
Sub com(x1/4)	Analog	1 Channel
	Serial	
	Discrete	
Sub com(x1/8)	Analog	1 Channel
	Serial	
	Discrete	
	Analog	
	Serial	
	Discrete	
	Analog	
Sub com(x1/16)	Serial	1 Channel
	Discrete	
	Analog	
	Serial	
	Discrete	
	Analog	
	Serial	
	Discrete	
	Analog	
	Serial	
	Discrete	
	Analog	
	Serial	
	Discrete	
Normal com	Minor id	1 Channel
	Analog	1 Channel
	Discrete	1 Channel
	Serial	1 Channel
	Empty	7 Channel
	CRC	1 Channel
	Major id	2 Channel
	Sync code	2 Channel

3-2 시뮬레이션 결과

엑셀 기반의 프로그램에 표2의 채널 정보를 입력하면 그림 6과 같이 채널 배치표가 자동으로 생성되고 그림 7의 ROM에 저장할 MIF 파일이 생성된다.

Channel No.	Channel Name	Channel No.	Channel Name	Channel No.	Channel Name	Channel No.	Channel Name	Channel No.	Channel Name	Channel No.	Channel Name
1	Minor id	3		5		7		9	Analog	11	
2	subcom1_1 subcom1_2	4	subcom2_1 subcom2_2 subcom2_3 subcom2_4	6	subcom3_1 subcom3_2 subcom3_3 subcom3_4 subcom3_5 subcom3_6 subcom3_7 subcom3_8	8	subcom4_1 subcom4_2 subcom4_3 subcom4_4 subcom4_5 subcom4_6 subcom4_7 subcom4_8 subcom4_9 subcom4_10 subcom4_11 subcom4_12 subcom4_13 subcom4_14 subcom4_15 subcom4_16	10	Discrete	12	
3											

그림 6. 자동 채널 배치

Fig. 6. Auto generated channel table

Addr	+0	+1	+2	+3	+4	+5	+6	+7	ASCII
0	0AAA810	0020000	11C0000	11E0000	00733CC	13C0010	13E0000	13D0000	
8	13C0020	00733CC	17C0030	17E0000	17D0000	17C0040	17E0000	17D0000	
16	17C0050	17E0000	00733CC	1FC0060	1FE0000	1FD0000	1FC0070	1FE0000	
24	1FD0000	1FC0080	1FE0000	1FD0000	1FC0090	1FE0000	1FD0000	1FC00A0	
32	1FE0000	1FD0000	1FC00B0	00400F0	0050000	0060000	00733CC	00733CC	
40	00733CC	00733CC	0010001	0010000	0030000	000FE6B	0002840	1FFFFFF	
48	1FFFFFF	1FFFFFF	1FFFFFF	1FFFFFF	1FFFFFF	1FFFFFF	1FFFFFF	1FFFFFF	

그림 7. ROM 테이블 채널 정보
Fig. 7. ROM table with channel information stored

그림 7의 첫 번째 주소에는 메이저 프레임의 크기 정보가 저장된 헤더가 할당되어 있으며 “0x00AAA810”의 하위 10비트를 통해 메이저 프레임 크기가 16임을 확인할 수 있다. 두 번째 주소부터는 사용자가 입력한 채널 정보가 할당되며 프레임 크기 이외의 ROM의 잔여 주소에는 채널의 끝을 의미하는 “0x1FFFFFF”가 할당된 것을 확인할 수 있다. 또한 프레임 크기 내의 할당하지 않은 채널은 빈 채널에 해당하는 채널 타입과 미리 약속된 특정 데이터 값 “0x00733CC”이 할당된다.

그림 7과 같이 ROM 테이블이 저장된 MIF 파일을 PCM 엔코더의 ROM에 저장하여 시뮬레이션을 진행하였으며 시그널 탭으로 확인한 결과는 그림 8과 같다.

그림 8에서 chan_cnt는 ROM의 주소를 의미하며, chan_wd는 해당 ROM 주소의 채널 정보를 의미한다. major_length는 헤더의 메이저 프레임의 크기를 나타내며, view_subcom_multi와 view_subcom_cycle은 각각 서브콤 배수와 서브콤 사이클을 의미한다. buf_mux를 통해 각 채널 정보에 따라 프레임을 구성하기 위한 데이터를 확인할 수 있다. minor_count는 마이너 프레임 카운트로 마이너 아이디의 데이터 값이 되며 메이저 프레임의 크기와 같아지면 0으로 초기화된다.

그림 7의 ROM 테이블 정보에서 0번 주소에 해당하는 헤더 정보의 메이저 프레임 크기와 그림 8의 시뮬레이션의 시그널 탭 결과 값 중의 major_length에 저장된 값이 16으로 일치하는 것을 확인하였다. 또한 헤더 정보가 저장된 0번 주소는 PCM 엔코더 전원이 인가된 후 초기 1번만 읽어오므로 major_length가 16으로 일정하게 출력되는데 이는 PCM 엔코더에서 ROM 테이블의 헤더 정보를 읽는 과정이 정상적으로 동작하는 것을 의미한다.

Name	0	256	512	768	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3
..B_1024K:ENC_U1 PCM_SYNC_P																
⊞ ..24K:ENC_U1 chan_cnt[9..0]		46	47		1				2				3		4	
⊞ ..24K:ENC_U1 chan_wd[24..0]		0002840h				0020000h						11E0000h				
⊞ ..ENC_U1 major_length[31..0]								16								
⊞ ..1 view_subcom_multi[31..0]								16					2			
⊞ ..1 view_subcom_cycle[31..0]								2					1			
⊞ ..24K:ENC_U1 buf_mux[15..0]		D56Eh			FE6Bh				2840h					0003h		
⊞ ..ENC_U1 minor_count[15..0]		2							3							
⊞ ..K:ENC_U1 buf_mn_id[15..0]														3		
⊞ ..24K:ENC_U1 chan_type[2..0]				000b						010b					110b	

그림 8. 커뮤테이션 구조의 시뮬레이션 결과
Fig. 8. Simulation result of commutation structure

ROM의 1번 주소부터는 프레임을 구성하는 채널 정보가 존재한다. PCM 엔코더는 1번 주소부터 라이징 클럭에 맞춰 채널 정보를 읽어 들여 프레임을 구성하게 된다.

그림 8의 chan_cnt가 1일 때 chan_wd는 “0x0020000”이며 chan_type이 0b.010임을 확인할 수 있다. 채널 타입이 0b.010이면 표 1과 같이 마이너 아이디를 의미한다. 시그널 탭 결과를 보면 buf_mux에 마이너 프레임 카운트가 1씩 증가되면서 출력된다. 이를 통해 채널 타입에 따라 데이터가 정상적으로 구성되어 출력됨을 확인하였다.

그림 8에서 ROM의 주소가 1로 초기화되면서 채널 정보 chan_cnt가 1 ~ 47로 반복 출력된다. 또한 마이너 프레임 카운트가 chan_cnt가 1일 때 11에서 12로 1씩 증가된다. 이는 47의 채널 정보가 “0x1FFFFFF”이므로 프레임 구성을 종료하고, 새로운 마이너 프레임을 구성하기 때문이다. 이를 통해 매 주기마다 PCM 프레임 구성을 검증하였다.

다음으로 서브콤 채널 구성의 정상 동작을 확인하였다. 그림 7의 ROM 테이블에서 1/2배 서브콤 1이 할당된 주소는 2번과 3번으로 마이너 프레임 카운트에 따라 buf_mux에 해당 서브콤 채널 데이터가 할당되어야 한다. 마찬가지로 1/4배 서브콤 2는 5~8번 주소, 1/8배 서브콤 3은 10~17번 주소, 1/16배 서브콤 4는 19~34번 주소에 위치하며 해당 주소의 채널 정보를 읽어 들여 프레임을 구성한다. 마이너 프레임 카운트가 15일 때 1/2배 서브콤 1의 사이클은 1이며, 3번 주소의 채널 정보를 읽어야 한다.

그림 9의 결과를 보면 서브콤 배수인 view_subcom_multi가 2일 때 chan_cnt는 3이며, view_subcom_cycle이 1, chan_type이 0b.110으로 시리얼이다. 이를 통해 표 2에서 설정한 1/2배 서브콤에 대한 채널 정보가 일치하는 것을 확인할 수 있다.

Name	2048	3072	4096	5120
024K:ENC_U1 PCM_SYNC_P				
024K:ENC_U1 chan_cnt[9..0]	1	2	3	4
24K:ENC_U1 chan_wd[24..0]	0020000h	11E0000h	00733CC	
:ENC_U1 major_length[31..0]	16			
1 view_subcom_multi[31..0]	16	2		
1 view_subcom_cycle[31..0]	14	1		
24K:ENC_U1 buf_mux[15..0]	2840h	000Fh	AAA	
:ENC_U1 minor_count[15..0]	15			
4K:ENC_U1 buf_mn_id[15..0]	14	15		
24K:ENC_U1 chan_type[2..0]	010b	110b	111b	

그림 9. 1/2배 서브콤1 시그널 탭 결과
Fig. 9. Signal tap result of 1/2 multiple subcom 1

Name	5120	6144	7168	8192
024K:ENC_U1 PCM_SYNC_P				
024K:ENC_U1 chan_cnt[9..0]	4	5	8	9 10 17
24K:ENC_U1 chan_wd[24..0]	13C0020h		00733CCh	
:ENC_U1 major_length[31..0]	16			
1 view_subcom_multi[31..0]	2	4		8
1 view_subcom_cycle[31..0]	1	3		7
24K:ENC_U1 buf_mux[15..0]	AAAAh		33CCh	2C2Ch
:ENC_U1 minor_count[15..0]	15			
4K:ENC_U1 buf_mn_id[15..0]	15			
24K:ENC_U1 chan_type[2..0]	111b	100b	111b	

그림 10. 1/4배 서브콰 2 시그널 탭 결과
Fig. 10. Signal tap result of 1/2 multiple subcom 2

Name	8192	9216	10240	11264
1024K:ENC_U1 PCM_SYNC_P				
024K:ENC_U1 chan_cnt[9..0]	9 10	17	18	19 34
24K:ENC_U1 chan_wd[24..0]	17E0000h		00733CCh	
:ENC_U1 major_length[31..0]	16			
1 view_subcom_multi[31..0]	4	8		16
1 view_subcom_cycle[31..0]	3	7		15
24K:ENC_U1 buf_mux[15..0]	2C2Ch	33CCh	AAAAh	
:ENC_U1 minor_count[15..0]	15			
4K:ENC_U1 buf_mn_id[15..0]	15			
24K:ENC_U1 chan_type[2..0]	111b	110b	111b	

그림 11. 1/8배 서브콰 3 시그널 탭 결과
Fig. 11. Signal tap result of 1/8 multiple subcom 3

Name	10240	11264	12288	13312	14336
024K:ENC_U1 PCM_SYNC_P					
024K:ENC_U1 chan_cnt[9..0]	18	19	34	35	
24K:ENC_U1 chan_wd[24..0]	00733CCh		1FC00Boh	00400F0h	
:ENC_U1 major_length[31..0]	16				
1 view_subcom_multi[31..0]	8	16			
1 view_subcom_cycle[31..0]	7	15			
24K:ENC_U1 buf_mux[15..0]	33CCh	AAAAh	33CCh		
:ENC_U1 minor_count[15..0]	15				
4K:ENC_U1 buf_mn_id[15..0]	15				
24K:ENC_U1 chan_type[2..0]	110b	111b	100b		

그림 12. 1/16배 서브콰 4 시그널 탭 결과
Fig. 12. Signal tap result of 1/16 multiple subcom 4

그림 10은 1/4배 서브콰 2의 시그널 탭 결과로 마이너 프레임 카운트가 15일 때 사이클은 3이며, 8번 주소의 채널 정보를 읽어온다. 마찬가지로 그림 11은 1/8배 서브콰 3의 결과로 사이클은 7이며, 17번 주소의 채널 정보를 가지고 프레임을 구성한다. 1/4배와 1/8배 서브콰의 chan_type은 각각 0b.100과 0b.110으로 표 2의 채널 타입과 일치한다. 마지막으로 그림 12와 같이 1/16배 서브콰 4의 경우 서브콰 사이클이 15이고, 34번주소의 채널 정보로 표 2와 일치하는 데이터를 출력한다.

PCM 엔코더의 서브콰 채널 구성에 대한 마이너 프레임 카운트별 시뮬레이션 시그널 탭 결과는 표 3과 같다.

표 3의 결과를 통해 PCM 엔코더의 서브콰 채널 구성에 대한 동작을 확인할 수 있다.

표 3. 서브콰의 시그널 탭 결과

Table 3. Signal tap result of subcom

Minor count	Subcom	Address	Cycle	Type	MUX buffer
0	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	5	0	0b.100	0x2C2C
	subcom 3	10	0	0b.100	0x2C2C
	subcom 4	19	0	0b.100	0x2C2C
1	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	6	1	0b.110	0xAAAA
	subcom 3	11	1	0b.110	0xAAAA
	subcom 4	20	1	0b.110	0xAAAA
2	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	7	2	0b.101	0xAC2C
	subcom 3	12	2	0b.101	0xAC2C
	subcom 4	21	2	0b.101	0xAC2C
3	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	8	3	0b.100	0x2C2C
	subcom 3	13	3	0b.100	0x2C2C
	subcom 4	22	3	0b.100	0x2C2C
4	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	9	0	0b.100	0x2C2C
	subcom 3	14	4	0b.110	0xAAAA
	subcom 4	23	4	0b.110	0xAAAA
5	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	6	1	0b.110	0xAAAA
	subcom 3	15	5	0b.101	0xAC2C
	subcom 4	24	5	0b.101	0xAC2C
6	subcom 1	2	0	0b.100	0xAAAA
	subcom 2	7	2	0b.101	0xAC2C
	subcom 3	16	6	0b.100	0x2C2C
	subcom 4	25	6	0b.100	0x2C2C
7	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	8	3	0b.100	0x2C2C
	subcom 3	17	7	0b.110	0xAAAA
	subcom 4	26	7	0b.110	0xAAAA
8	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	5	0	0b.100	0x2C2C
	subcom 3	10	0	0b.100	0x2C2C
	subcom 4	27	8	0b.101	0xAC2C
9	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	6	1	0b.110	0xAAAA
	subcom 3	11	1	0b.110	0xAAAA
	subcom 4	28	9	0b.101	0xAC2C
10	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	7	2	0b.101	0xACEC
	subcom 3	12	2	0b.101	0xACEC
	subcom 4	29	10	0b.110	0xAAAA
11	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	8	3	0b.100	0x2C2C
	subcom 3	13	3	0b.100	0x2C2C
	subcom 4	30	11	0b.101	0xAC2C
12	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	5	0	0b.100	0x2C2C
	subcom 3	14	4	0b.110	0xAAAA
	subcom 4	31	12	0b.100	0x2C2C
13	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	6	1	0b.110	0xAAAA
	subcom 3	15	5	0b.101	0xAC2C
	subcom 4	32	13	0b.110	0xAAAA
14	subcom 1	2	0	0b.100	0x2C2C
	subcom 2	7	2	0b.101	0xAC2C
	subcom 3	16	6	0b.100	0x2C2C
	subcom 4	33	14	0b.101	0xAC2C
15	subcom 1	3	1	0b.110	0xAAAA
	subcom 2	8	3	0b.100	0x2C2C
	subcom 3	17	7	0b.110	0xAAAA
	subcom 4	34	15	0b.100	0x2C2C

IV. 결 론

원격측정 시스템은 획득한 비행체의 상태 정보들을 실시간으로 분석이 가능한 프레임 형식으로 구성하여 전송해야한다. 때문에 프레임을 구성하기 위한 채널 정보를 PCM 엔코더에서 저장하고 있어야한다. 최근에는 대형 비행체의 분산 계측 장치들이 추가 되거나 센서들의 계측 데이터가 많아지면서 프레임을 구성하는 채널의 수가 점차 증가하고 있다. 하지만 한정된 메모리에서 수많은 채널 정보를 모두 입력하고 저장하기에는 어려움이 있다. 메모리의 용량부족으로 많은 정보를 저장할 수가 없고, 사용자가 채널 정보를 입력하는 양이 많아질수록 잘못 입력될 가능성이 커진다. 때문에 효율적인 채널 정보 메모리 관리 및 입력 방법이 필요하다.

본 논문에서는 FPGA의 한정된 메모리에서 보다 효율적으로 채널 정보를 저장할 수 있는 방법을 제안하고 시뮬레이션을 통해 동작을 확인하였다. 엑셀 기반의 프로그램을 통해 채널 구성 정보 및 프레임 정보를 입력하면 채널 배치표와 ROM 테이블 정보가 생성된다. 메이저 프레임의 마이너 프레임들 중에 같은 채널 정보가 반복되는 채널들은 ROM 테이블에 채널 정보를 한번만 할당하고, 마이너 프레임마다 채널 정보가 다른 서브콤 채널들은 서브콤 배수만큼 ROM 테이블에 채널 정보를 할당한다. 이러한 ROM 테이블 구조를 위해 서브콤 채널 처리가 가능한 PCM 엔코더를 구현하여 프레임을 구성하였다.

시뮬레이션과 같이 메이저 프레임의 길이를 20으로 설정하고, 제안한 ROM 테이블과 같이 서로 다른 배수의 서브콤 채널에 대해 PCM 엔코더 구조가 정상적으로 동작함을 확인하였다. 시뮬레이션 결과를 통해 제안한 구조로 구현한 PCM 엔코더의 한정된 채널 정보 메모리를 효율적으로 사용이 가능하였다.

다만, ROM에 저장할 채널 정보가 바뀌게 되면 변경된 MIF 파일을 저장하여 다시 컴파일을 해야 하므로 시간이 오래 걸린다. 때문에 추후 PCM 엔코더는 다시 컴파일을 하지 않고 내부

에서 자동으로 생성하는 방안을 제시하거나 외부 메모리를 이용하여 직접 제어가 가능하도록 설계하는 연구가 필요하다.

References

- [1] H. M. Eckstein, "A programmable-signal conditioning pulse code modulated telemetry encoder," in *Proceedings of the International Telemetry Conference*, San Diego: CA, Vol. 17, pp. 1319-1328, Oct. 1981.
- [2] N. C. Poirier and T. P. Wheeler, "Programmable PCM Encoder." in *Proceedings of the International Telemetry Conference*, Las Vegas: NV, Vol. 20, pp. 78-91, Oct. 1984.
- [3] G. H. Kim, M. H. Jin and B. K. Kim, "Design of a simple PCM Encoder architecture based on programmable ROM," *Journal of Advanced Navigation Technology*, Vol. 23, No.2, pp. 186-193, Apr. 2019.
- [4] Y. K. Park and W. J. Yoon, "PCM encoder structure for real-time updating of telemetry system parameters," *Journal of Advanced Navigation Technology*, Vol. 23, No.5, pp. 452-459, Oct. 2019.
- [5] J. S. Kim and M. J. Jang, "Implementation of A 30-Channel PCM Telemetry Encoder," in *Proceedings of the International Telemetry Conference*, San Diego: CA, Vol. 40, pp. 141-150, Oct. 2004.
- [6] D. J. Hermen, and J. O. Horsley, "Application of general purpose computer techniques to the design of a programmable PCM telemetry decommutator," in *Proceeding of the 3rd Space Congress*, Cocoa Beach: FL, pp. 396-408, Mar. 1966.



노 윤 희 (Yun-Hee Ro)

2015년 2월 : 선문대학교 컴퓨터공학과 (공학사)
2014년 12월 ~ 2018년 2월 : (주)비트컴퓨터 ASP솔루션부
2018년 2월 ~ 2018년 11월 : 하이밸류컨설팅(주) 의료사업팀
2018년 12월 ~ 현재 : 단암시스템즈(주) 통신기술연구소 재직 중
※ 관심분야 : 원격측정장치 자료처리



김 건 희 (Geon-Hee Kim)

2014년 2월 : 서울과학기술대학교 전자정보공학과 (공학사)
2015년 4월 ~ 현재 : 단암시스템즈(주) 통신기술연구소 재직 중
※ 관심분야 : 항공 통신 시스템, 원격측정장치



김 동 영 (Dong-Young Kim)

2012년 2월 : 선문대학교 컴퓨터공학과 (공학사)
2012년 2월 ~ 2013년 4월 : 웹솔루션(주) 융합기술연구소
2013년 5월 ~ 2013년 11월 : 다음평가터치(주) 시스템개발팀
2014년 3월 ~ 현재 : 단암시스템즈(주) 통신기술연구소 재직 중
※ 관심분야 : 원격측정장치 자료처리



김 복 기 (Bok-Ki Kim)

1995년 2월 : 서울대학교 수학과 (이학사)
1997년 2월 : 서울대학교 수학과(정수론) (이학석사)
1997년 1월 ~ 2002년4월 : 단암전자통신(주) 연구소
2002년 5월 ~ 현재 : 단암시스템즈(주) 통신기술연구소 재직 중
※ 관심분야 : 무선통신, 채널코딩, 디지털 신호처리 구조



이 남 식 (Nam-Sik Lee)

2001년 2월 : 경희대학교 전파공학과 (공학사)
2001년 2월 ~ 현재 : 단암시스템즈(주) 통신기술연구소
※ 관심분야 : 무선통신 시스템, 암호화, 원격측정장치