

영상 전송을 위한 Reed-Solomon Encoder의 FPGA 구현

김동년* · 채우청** · 변건식***

동아대학교

FPGA Implementation of Reed-Solomon Encoder for image transmission

Dong-Nyeon Kim* · Cai Yu Qing** · Kun-sik Byon***

Dong-A University

E-mail : may8114@hanmail.net

요 약

본 논문에서는 디지털 이미지의 무선 전송시 발생하는 에러를 제거 할 수 있는 오류 제어 부호화 기술 중 하나인 RS(Reed-Solomon) coding 중 Encoder의 FPGA 구현에 대한 논문이다. 여러 오류 제어 부호화 기술중 RS coding의 경우 비트가 아닌 심볼로 부호화를 하기 때문에 연집에러에 강한 것으로 알려져 있다.

본 논문에서는 RS coding중 최대 16바이트(Byte)의 에러를 정정할 수 있는 RS(255,223) 부호기를 FPGA 구현을 위해 Matlab을 이용하여 시뮬레이션을 하였다. 이때 사용한 영상은 JPG 파일을 이용하였다. 그리고 FPGA 구현을 위해 Xilinx사의 System Generator를 사용 하였다.

ABSTRACT

This paper is the FPGA Implementation of Reed-Solomon Encoder that is one of Error control Codes. Reed-Solomon codes are block-based error control codes with a wide range of applications in digital communications. RS codes are strong on burst errors because it process signals as symbol.

We simulate this system using Matlab from Mathworks and design it using System Generator from Xilinx. We refer Matlab source in Implementation of Reed-Solomon Error Control Coding for Compressed Images by Simon Anthony Raspa.

Key words

Reed-Solomon code, Encoder, ECC, System Generator

I. 서 론

현재 무선 통신은 인간 사회에서 중요한 부분을 차지하고 있다. 수많은 소비자들은 다양한 통신기기를 이용하여 영상, 비디오 및 음악 등 고용량의 데이터를 무선 전송하고 있다.

영상을 무선 전송 할 때, 채널에서 랜덤 오류와 연집 오류가 발생할 수 있으며, 수신기에서 영상은 왜곡되고 경우에 따라서는 재생 할 수 없게 된다. 따라서 수신기에 무오류의 영상이 나타날 수 있도록 에러를 제거하는 오류 제어 부호화 기술이 필요하다. 이러한 오류 제어 부호화 기술 중 RS(Reed-Solomon) 코드는 다른 선형 블록 부호와는 달리 심볼로 부호화하기 때문에 연집 오류

에 효율적이라고 알려져 있다[1].

본 논문에서 구현한 RS 부호기는 $GF(2^8)$ 상에서 최대 16 바이트의 에러를 정정할 수 있는 RS(255,223) 부호기로 223바이트의 메시지와 32바이트의 패리티를 가지는 부호어 이다. RS 부호기를 설계하기 위해 Matlab 프로그램을 이용하여 시뮬레이션을 하였고, 시뮬레이션을 한 Matlab 프로그램을 기반으로 System Generator를 이용하여 RS 부호기를 구현 하였다.

II. RS Encoder

본 논문에서 구현한 RS 부호기의 목적은 무선

영상을 전송하는 것이기 때문에 데이터를 조직 형태로 부호화하는 것이 유용하다[1]. 즉 데이터를 다음 <그림 1>과 같은 형태로 부호화 한다.

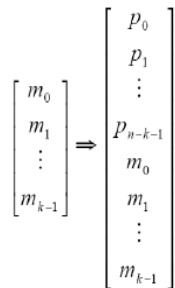


그림 1. 조직 형태의 RS 부호화

여기서 m_i 는 입력되는 원래의 메시지 심볼을 나타내고, p_i 는 각 패리티 심볼을 나타낸다. 조직 형태의 RS 부호화에서 원래의 메시지는 부호화된 블록에 들어있음을 알 수 있다.

III. 조직형태의 RS 부호

RS 부호화를 하기 위해서는 우선 생성 다항식을 구하여야 한다. 생성 다항식은 다음 식 (1)과 같이 나타 낼 수 있는데 여기서 생성 다항식의 차수는 패리티의 심볼과 같으며 't'는 RS 부호의 오류정정능력을 나타낸다.

$$g(x) = g_0 + g_1x + g_2x^2 + \dots + g_{2t-1}x^{2t-1} + x^{2t} \quad (1)$$

위의 생성 다항식에서 $g_0, g_1, \dots, g_{2t-1}$ 의 값들은 구현하고자 하는 RS Encoder에서 패리티를 형성하는 블록에 사용된다. 하지만 실제 위의 식과 같이 32차의 식을 전개하는 것은 많은 시간을 요구하므로 Matlab 프로그램을 이용하여 그 값을 구하였다.

조직적인 형태의 RS 부호화를 하기 위해서는 부호화 하고자하는 메시지 다항식 $m(x)$ 에 x^{n-k} 를 곱하고 패리티 다항식 $p(x)$ 를 더한 형태가 된다. 여기서 메시지 다항식 $m(x)$ 에 x^{n-k} 를 곱하는 것은 메시지 다항식을 $n-k$ 만큼 천이 하는 것이다.

$$x^{n-k}m(x) = q(x)g(x) + p(x) \quad (2)$$

여기서 $p(x)$ 는 $p(x) = x^{d-1}m(x) \text{ mod } g(x)$ 로 나타낼 수 있으며 $g(x)$ 는 생성 다항식이다. 그리고 부호화된 다항식 $c(x)$ 는 다음과 같다

$$c(x) = x^{n-k}m(x) + p(x) \quad (3)$$

아래의 <그림 2>는 일반적인 RS Encoder의 구조로 식(1)에서 구해진 다항식의 계수들을 이용하여 패리티 심볼을 만들고 입력 메시지 심볼과 더하여 조직형태의 RS 부호어를 만드는 시스템을 나타낸다.

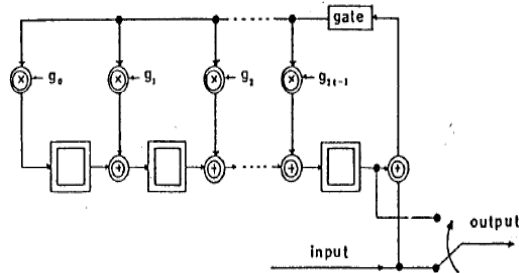


그림 2. 일반적인 RS Encoder

IV. 시뮬레이션

본 논문에서는 Mathworks사의 Matlab 프로그램을 이용하여 구현하고자 하는 RS Encoder를 시뮬레이션 해 보았다. 시뮬레이션에 사용한 영상은 7KB, 118KB 크기의 그림으로서 각각 55896개, 963216개의 데이터를 가지고 있다.

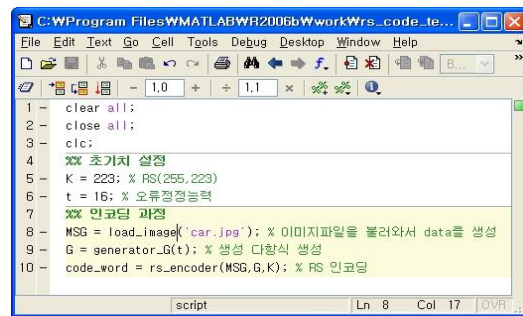


그림 3. RS Encoder Matlab Source

위의 프로그램은 RS(255,223) Encoder과정을 나타내는 것으로 'load_image'함수는 영상을 불러와 data를 생성하여 MSG라는 변수로 저장하는 함수로 위의 프로그램에서 사용한 'car.jpg'는 7KB 크기의 영상이다. 그리고 'generator_G' 함수는 생성 다항식을 만드는 함수로 오류정정능력 t를 입력으로 받는다. 함수 내부에는 Galois Field 승산을 위한 함수와 xor 함수로 구성되어 있다. 본 논문에서 구현한 RS(255,223)의 경우 최대 16바이트의 에러를 정정 할 수 있으며 생성되는 생성다항식

VI. Co-simulation 결과

앞절에서 설계한 System Generator 모델의 검증 및 Target Device에서 동일하게 동작 하는지를 확인하기 위해 Hardware Co-Simulation을 수행하였다. 이때 사용한 Target Device는 Spartan 3 xc3s1000-4fg676이며 Xilinx System clock 주기는 20ns로 하였다.

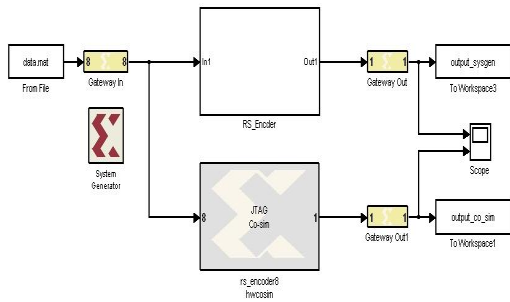


그림 8. Hardware Co-simulation 실험

다음의 <그림 9>과 <그림 10>은 Hardware Co-simulation 결과와 System Generator의 결과를 matlab의 workspace와 Scope의 출력을 비교한 것으로 Scope의 출력은 심볼을 Bit로 변환 하여 확인 하였다. 결과는 그림과 같이 동일한 값을 가지는 것을 알 수 있다.

Time	1	2	3
1017	3		3
1018	3		
1019	2		
1020	4		
1021	3		
1022	223		
1023	225		
1024	154		
1025	29		
1026	171		
1027	211		
1028	13		
1029	2		
1030	203		
1031	205		

그림 9. System Generator와 Hardware Co-simulation 결과비교

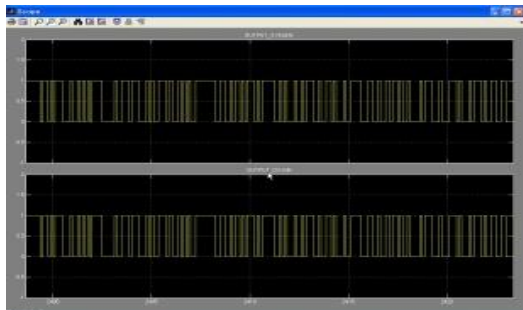


그림 10. System Generator와 Hardware Co-simulation Scope 출력비교

그리고 Xilinx사의 ISE Tool를 이용하여 Resource Estimation한 결과는 아래의 그림과 같다.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2575	7680	33%
Number of Slice Flip Flops	1193	15360	7%
Number of 4 input LUTs	5107	15360	33%
Number of bonded IOBs	10	391	2%
Number of BRAMs	4	24	16%
Number of GCLKs	1	8	12%

그림 11. Resource Estimation 결과

VII. 결론

본 논문은 영상 전송을 위한 RS Encode의 FPGA 구현에 대한 논문으로 RS(255,223) Encoder를 Matlab을 이용하여 시뮬레이션을 하였고, System Generator를 이용하여 System을 설계 후 ISE를 이용하여 FPGA를 구현하였다.

지금까지 설계된 Reed Solomon System의 경우 입력부의 Data를 Matlab상에서 임의로 32개의 '0'을 추가한 Data를 사용하였다. 향후 입력부에 32개의 '0'을 추가하는 부분, 즉 입력 Data를 32만 큼 천이 시키는 블록을 추가하도록 하겠다. 그리고 입력을 영상을 이용하였지만 현재까지는 Encoder만 설계가 되어 결과를 확인 할 수 없다. 향후 Decoder부분까지 완성한 후 잡음을 추가하여 RS Coding의 성능을 분석하도록 하겠다.

참고문헌

- [1] Simon Anthony Raspa "Implementation of Reed-Solomon Error Control Coding for Compressed Image" 2005
- [2] Jorge Castiñeira Moreira, Patrick Guy Farrell "Essentials of Error-Control Coding" ,2006, p144 ~ p184
- [3] www.xilinx.com
- [4] www.mathworks.com
- [5] Xilinx System Generator for DSP user's guide