2011 International Conference on Telecommunication Technology and Applications
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

FPGA Implementation Reed Solomon Encoder for Visual Sensor Networks

Jia Jan Ong, L.-M. Ang and K. P. Seng

Department of Electrical and Electronic
The University of Nottingham Malaysia

Abstract. Reliability of data transmission across the visual sensor network is a major consent for image data captured by image sensor. The available error correction capabilities integrated in the wireless sensor network modules, it is not sufficient to protect a huge amount of transmitted data. With errors often occurred on the transmitted data, the base-station would not be able to receive the accurate information from the sensor nodes. Hence retransmission of data from the sensor nodes to the sink are required which increase power consumption and decrease the lifespan of the sensor nodes. Compare with traditional Reed Solomon encoder, a simple (255,223) Reed Solomon encoder that has low power consumption was implemented onto the visual sensor network. Therefore, the transmission can tolerate more errors and less retransmission is needed.

Keywords: Visual Sensor Networks, Reed Solomon, Minimal Instruction Set Computing.

1. Introduction

In recent years, wireless sensor network (WSN) has gained worldwide interest that resulted in thousand of peer-reviewed publications [1]. These sensor nodes are made up of small sensors and with a limited amount of processing capabilities [2]. These sensor nodes have the ability to gather information from the surrounding, process the sense data locally and transmit the information to the user, which is the base-station. However, most this research has being focused on scalar sensors that collects scalar data such as humidity, position of objects, temperature and pressure [1, 3]. With the emerges of low powered image sensors, these image sensors can replace the available scalar sensors in the WSN to give visual sensor networks (VSN) [4]. In comparison to the scalar sensors, the image sensors produce a large amount of data. It will be a challenge for the low powered sensor nodes, which has low processing capabilities and transmitting bandwidth, to handle a huge amount of data produced by the image sensor [3]. Since a huge amount of data is obtained for each image captured by the image sensors. With the current WSN error correction capabilities at link-layer [1, 3], unreliable data transmission will cost a large amount of image data loss and retransmission of data is required. This will result in huge amount data traffic that significantly lowers the network bandwidth and higher power consumption. Therefore, there is a need to have an error correction scheme that can provide reliable data transmission for the VSN. Since the sensor nodes are mainly powered by limited energy source, like battery [2]. The error correction encoder use in protecting the image data needs to be simple and has low power consumption. This paper proposes the implementation of (255,223) Reed Solomon (RS) minimal instruction set computing processor [5] as an encoder for the VSN. With the use of RS(255,223) coding scheme, the image data transmitted across the network is much more reliable since it can protect the data with 16 errors occur in a codeword. Less retransmission of image data is required thus increasing the bandwidth of the network and lower power consumption. The following section briefly explains the RS coding scheme used for this implementation. Section 3 discussed on the RS(255,223) minimal instruction set computing processor. Section 4 briefly discusses on the wireless sensor networks. Section 5 gives the overview of the implemented RS coding for the VSN. Section 6 provides the implementation results. Section 7 concludes this paper.

2. Reed Solomon

Reed Solomon (RS) coding scheme was introduced by Irving Reed and Gus Solomon in 1960 [6, 7]. With the discovery of Reed Solomon, this coding scheme has being exploited in many applications such as satellite communications, compact disc (CD) players, broadband, wireless communications, digital television, etc [7, 8]. For the implementation, RS(255,223) coding scheme was used that will take 223 information symbols and produce 32 redundant symbols. The redundant symbols will be added to the back of the information symbols to give a codeword with a total of 255 symbols. The RS(255,223) coding scheme can corrects up to 16 symbol errors that occurred in one codeword.

$$g(x) = x^{32}+232x^{31}+29x^{30}+189x^{29}+50x^{28}+142x^{27}+246x^{26}+232x^{25}+15x^{24}+43x^{23}+82x^{22} +164x^{21}+238x^{20}+x^{19}+158x^{18}+13x^{17}+119x^{16}+158x^{15}+224x^{14}+134x^{13}+227x^{12} +210x^{11}+163x^{10}+50x^{9}+107x^{8}+40x^{7}+27x^{6}+104x^{5}+253x^{4}+24x^{3}+239x^{2}+216x+45 \quad (1)$$

Consider that α to be a primitive element in the Galois Field GF(2^8), the symbols for the RS(255,223) code will have α, α^2,…, α^{32} as all of its roots. By multiplying all these roots together, the generator polynomial, g(x) of the RS encoder is shown in equation (1) [8, 9]. In hardware implementation of the RS encoder, the computational of redundant symbols are performed using a division circuit, which is known as the Linear Feedback Shift Register (LFSR) circuit. From Fig. 1, there will be 223 information symbols clocked into the shift registers. At the same time, the 223 symbols will also be clocked out as part of the codeword. With all the information symbols are clocked in, 32 redundant symbols are produced and stored in the shift registers. By clocking out the 32 redundant symbols, this will form a complete codeword that can be transmitted across the communication channel to the receiver. Since the encoded data will be transmitted to the base-station, which is a computer. The decoding of the received codeword is performed using the available MATLAB RS decoding functions at the base-station.

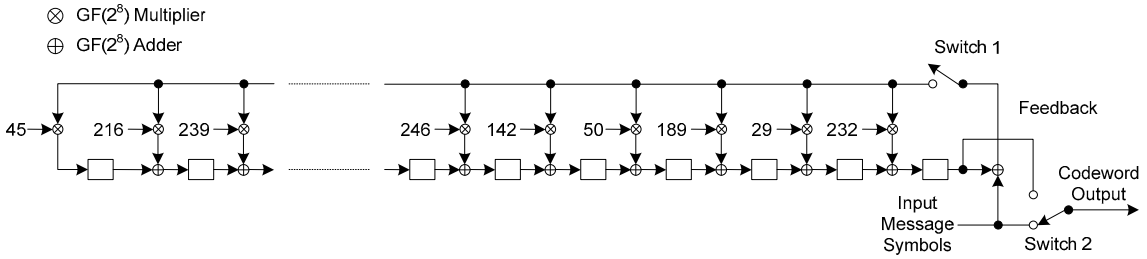

Fig. 1: A RS(255,223) encoder in Linear Feedback Shift Register configuration.

3. Minimal Instruction Set Computing

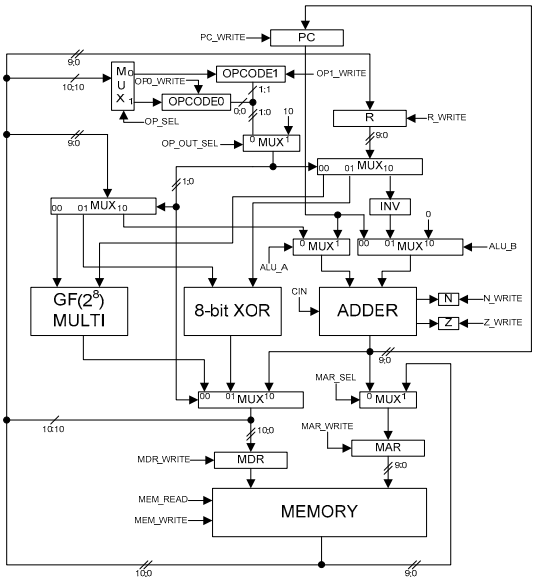

Fig. 2: RS(255,223) MISC processor architecture.

The Ultimate Reduced Instruction Set Computer (URISC) was introduced by Farhad and Behrooz [10]. It is a simple processor, known as the One Instruction Set Computer (OISC) that only operates on a single instruction. This processor is also considered to be the penultimate Reduced Instruction Set Computer (RISC) [11, 12]. There three different paradigms of OISC, this includes the Subtract and Brach if Negative (SBN) [13], MOVE and Half Adder. By considering the SBN OISC processor, the processor will execute the arithmetic operation of subtracting the 2nd operand (data B) with the 1st operand (data A). The result of the arithmetic operation is then stored back to the 2nd operand (B = B – A) thus replacing its original values [10, 12]. By further develop the OISC processor, a minimal instruction set computing (MISC) processor can be developed by introduction a few required instructions that will perform specific task.

3.1. RS(255,223) MISC Processor

The RS(255,223) MISC processor was developed based on the OISC processor by integrating two extra instructions, which are the GF instruction and XOR instruction [5]. Fig. 2 shows the developed RS(255,223) MISC processor architecture consists of three functional blocks, which are the GF(2^8) multiplier, 8-bit XOR and an Adder. From these three functional blocks, there are a total of three instructions available to program the processor, which are the GF, XOR and SBN instructions. The GF instruction will perform the GF(2^8) multiplication between two input data in modulo of primitive polynomial. As for the XOR instruction, the processor will execute the GF(2^8) arithmetic addition between two input data in modulo of 2 with the corresponding bits. Total amount of available memory is 1408 bytes, which consists of both data and program memory that are equally allocated.

3.2. RS(255,223) Encoder Program

Without program codes, the RS(255,223) MISC processor would not be able to encode the input data. Consequently, a RS(255,223) encoder program was written based on the LFSR circuit, as shown in Fig. 1. The program will read the 1st information symbol input to the processor. Then the symbol is added with the 1st shift register from the right to give the feedback. The feedback is then multiplies with the 1st coefficient of the generator polynomial and add with the 2nd shift register value. The result is stored to the 1st shift register. This process is then repeated for the 2nd until 32nd coefficients. By having all the coefficients multiplying with the feedback, this process is repeated again for the 2nd until 223rd information symbols. A total of 68 lines of program instructions were written and programmed into the processor.

4. Wireless Sensor Networks

Wireless sensor network (WSN) consists of many small sensors that are small and with a limit amount of processing capabilities [2]. These sensors measure the environment and process the gathered information before transmitting the important data to the user. Nevertheless, there are many researches had been done for the WSN [2, 14]. However, these research area are performed on using the scalar sensor, that only a limited amount of information is obtained, which might be inadequate for many applications [3]. With the huge leap in the image sensor technology, many products such as toys, laptops and cellular phone has embedded with a low powered image sensor [4]. Thus, the image sensor is now replacing the scalar sensors that sense the temperature, pressure or humidity in the WSN. This gives raise to the visual sensor network, where each of tiny visual sensor nodes consists of an integrated image sensor, embedded processor and a wireless transceiver [4]. Fig. 3 shows the data that are transmitted from one sensor nodes to the sink through a few other sensor nodes. The sink or base-station is made up of a computer and a receiver that will be kept on receiving information from the sensor nodes. Then the computer processes the received data and response to the measured situation at the sensor nodes.

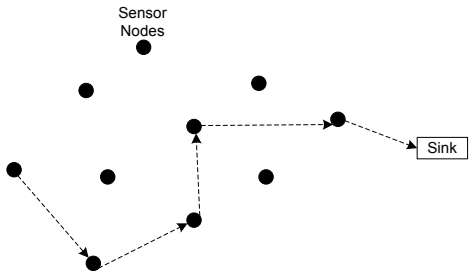

Fig. 3: Data transmission between sensor nodes and sink.

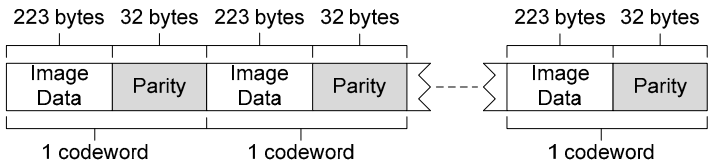

Fig. 4: Transmission of image data splits into codeword through Digi XBee ZB RF modules.

For the implementation, two Digi XBee ZB RF modules were used to transmit the captured image from the sensor node to the base-station. The modules communicate using the ZigBee mesh network that is stated in the IEEE802.15.4 standard. These modules operate at the frequency of 2.4GHz with maximum data throughput of 35kbps. One of the Digi XBee ZB RF modules was used on the visual sensor nodes (image sensor) and another on the base-station (sink), which is the computer. The encoded image data are transmitted across the wireless channel in blocks of codeword. Each blocks of codeword are made up of 223

information symbols and 32 redundant symbols (parity). For each 160x120 image, a total of 87 blocks of codeword are encoded and sent to the base-station. Each frame of image captured by the image sensor, a total of 22185 bytes data are required to be transmitted across the sensor nodes to the sink.

5. Implementation (255,223) Reed Solomon Coding in Visual Sensor Network

For this implementation, the Reed Solomon coding scheme integrated onto the visual sensor network consists of one CMOS camera, memory buffer, RS(255,223) MISC processor, two Digi XBee ZB RF modules and a computer. The CMOS camera will capture one frame of image size at 160x120 pixels. The captured image are then stored to the memory buffer. Once the whole image is completely stored to the memory, then the RS(255,223) MISC processor will process the image pixels. For each codeword produced, the RS MISC processor will take in 223 of data and generate one block of codeword that consists of 255 symbols. Once the codeword is produced, it will be sent to the computer (base-station) through the Digi XBee ZB RF modules. A total of 87 blocks of codeword are produced for each frame of image captured by the camera. With each blocks of codeword received by the computer, it will be decoded using the same RS(255,223) coding scheme. The decoding is performed using the available MATLAB RS decoding function. From Fig. 5, it can be seen that the data transfer to the Digi XBee ZB RF module through RS232 communication link. For the base-station, the received data is also transferred to the computer through the use of RS232 communication link.

The hardware used for this implementation is a Celoxica RC10 board, which consists of an OmniVision OV9650 CMOS camera, block RAMs, Xilinx Spartan-3L FPGA and RS232 communication. This implementation imitates the visual sensor node with an encoder that will encode the image data that are needed to be transmitted to the base-station. The encoded image data received by the base-station allows the computer to correct errors that occurred on the image data during transmission. With the capabilities of correcting a certain number of errors, retransmission of data is decreased that would reduced the amount of power consumption used in data transmission. The visual sensor node implemented comprise of the CMOS camera, block RAMs, Xilinx Spartan-3L FPGA and the Digi XBee ZB RF module. As for the base-station, it consists of a Digi XBee ZB RF module and a computer, which will decode the received codeword. Once the codeword is decoded, the image data acquired will be processed to obtain the actual image captured by the image sensor.

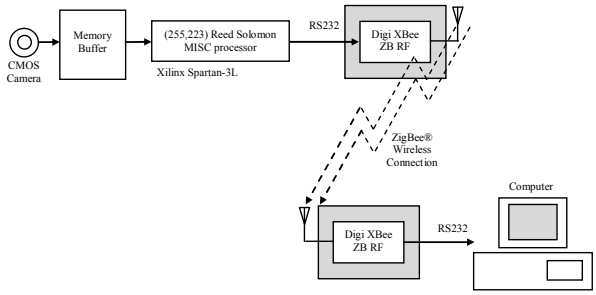

Fig. 5: The implemented Reed Solomon coding scheme on VSN.

6. Implementation Result

By considering the LFSR RS encoder implementations, it requires 32 8-bit XORs, 64 registers, 2 multiplexers and 32 GF multipliers. From Fig. 2, it can be seen that the implementations of the RS MISC processor only requires an XOR, 8 registers, 8 multiplexers and one GF multiplier. Compare with the LFSR RS encoder, as shown in Table 1, the RS MISC processor requires less hardware component to function as a RS encoder. The GF multiplier is the most complex circuit among other components and it requires a lot of hardware for implementing one GF multiplier. For the LFSR RS encoder, the numbers of GF multipliers needed are more than the RS MISC processor thus a large amount of hardware is needed. With less amount of hardware needed for the implementation, the power consumption of the implemented RS MISC processor will be lower in contrast with the LFSR RS encoder. Although the LFSR RS encoder can process the information symbol in parallel, it can encode the data faster compared to the RS MISC processor. Since the visual sensor network will only capture the image for a certain interval of time, the time take to encode one frame of image is not as important as for the real-time video streaming. The important issue for the visual sensor network is to operate the sensor node in low power. This will extend the lifespan of the sensor node with limit energy source. Table 2 shows that 8% of the available flip-flop and 6% of the available LUTs

were used for implementing the RS MISC processor for the VSN. Therefore, the implementation of RS MISC processor onto the Celoxica RC10 board requires a small amount of hardware for the VSN.

Component	RS MISC processor	RS LFSR encoder
Flip Flop	284 slice	415 slice
4 input LUTs	500	720
- Logic	458	684
- Route-thru	40	34
- Dual Port Rams	0	0
- Shift registers	2	2
Bonded IOB	46	46
Block RAMs	1	1
BUFGMUXs	3	3
DCMs	1	1

Table. 1: Hardware usage between LFSR RS encoder and RS MISC processor.

Component	Quantity	Total	Usage
Flip Flop	1089 slice	13312	8%
4 input LUTs	1646	26624	6%
- Logic	1441	-	-
- Route-thru	132	-	-
- Dual Port Rams	60	-	-
- Shift registers	13	-	-
Bonded IOB	35	221	15%
Block RAMs	10	32	31%
BUFGMUXs	5	8	62%
DCMs	2	4	50%

Table 2: Hardware usage in implementing RS MISC processor onto RC10 board for VSN.

7. Conclusion

With the implementation of RS MISC processor, the image data transmitted across the VSN can be protected using the RS coding scheme. Since the base-station can correct the errors on the received data, less retransmission is needed from the sensor nodes. Consequently, less amount of energy is needed for the sensor nodes to operate that gives longer lifespan of the device. Besides, low amount of hardware usage for the implementation of RS MISC processor, reduces the power consumption. In future, the codeword can be sent to the base-station in packets instead of continuous sending that does not indicate start and end of a codeword. By compressing the image using SPIHT algorithm, selective error protection will be done on important compressed coefficients that allows reconstruction of image with error occurs on these coefficients.

8. References

[1] Akyildiz, I.F., T. Melodia, and K.R. Chowdury, Wireless multimedia sensor networks: A survey. *IEEE Wireless Communications*. 2007, 14(6): p. 32-39.

[2] Yick, J., B. Mukherjee, and D. Ghosal, Wireless sensor network survey. *Computer Networks*. 2008, **52**(12): p. 2292-2330.

[3] Charfi, Y., N. Wakamiya, and M. Murata, Challenging issues in visual sensor networks. *IEEE Wireless Communications*. 2009, **16**(2): p. 44-49.

[4] Soro, S. and W. Heinzelman, A Survey of Visual Sensor Networks. *Advances in Multimedia*. 2009, p. 1-22.

[5] Jia Jan Ong, L.M. Ang, and K.P. Seng. Implementation of (255,223) Reed Solomon minimal instruction set computing using Handel-C. *3rd IEEE International Conference on Computer Science and Information Technology*. 2010, **5**, pp. 49-54.

[6] Reed, I.S. and G. Solomon, Polynomial Codes Over Certain Finite Fields. *Journal of the Society for Industrial and Applied Mathematics*. 1960, **8**(2): p. 300-304.

[7] Wicker, S. and V. Bhargava. *Reed-Solomon Codes and Their Applications*. IEEE Press, 1999.

[8] Khan, M.A., S. Afzal, and R. Manzoor. Hardware implementation of shortened (48,38) Reed Solomon forward error correcting code. *7th International Multi Topic Conference*. 2003, pp. 90-95.

[9] Lin, S. and D. Costello. *Error Control Coding (2nd Edition).* Prentice Hall, 2004.

[10] Mavaddat, F. and B. Parhami. URISC: the ultimate reduced instruction set computer. *International Journal of Electrical Engineering Education.* 1988, **25**(4): p. 327-334.

[11] Laplante, P.A. and W. Gilreath. Single Instruction Set Architectures for Image Processing. *Proceedings of SPIE*. 2002, **4868**: pp. 20-28.

[12] Gilreath, W.F. and P.A. Laplante. *Computer Architecture.* Springer Science+Business Media, Inc., 2002.

[13] Van Der Poel, W. A simple electronic digital computer. *Applied Scientific Research, Section B*. 1952, **2**(1): p. 367-400.

[14] Akyildiz, I.F., et al. A survey on sensor networks. *IEEE Communications Magazine*. 2002, **40**(8): p. 102-114.