

## **IMPLEMENTATION OF LOW POWER VLSI ARCHITECTURE FOR LOSSLESS COMPRESSOR AND DE-COMPRESSOR**

**VINAY S, PRAMOD K. P, R. JAYAGOWRI, S. RANJANA & PRIYANKA V**

Department of Electronics and Communication Engineering, K.S. Institute of Technology, Bangalore, Karnataka, India

### **ABSTRACT**

In this paper we present a compressor and de-compressor which is implemented for low power with GR algorithm using GSM technique. To implement the code compressor and de-compressor unit of the processor we have used the random signal and they are given to the ADC and to the lossless compressor and de-compressor block consists of GR unit. We were able to achieve the total power saving of 37.7% area increased for the compressor by 15.15%. Since the compressor is reducing the number of bits, while processing the area of the subsystems in the processor are reduced which in turn reduces the static power in processor. This compressor and decompressor unit is implemented using FPGA spartan3E and the same we implemented using cadence tool for ASIC flow with 180nm technology.

**KEYWORDS:** Compression Ratio, Data Compressor, Power

### **INTRODUCTION**

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. Thus the development of efficient compression techniques will continue to be a design challenge for future communication systems and advanced multimedia applications. Data is represented as a combination of information and redundancy. Information is the portion of data that must be preserved permanently in its original form in order to correctly interpret the meaning or purpose of the data. Redundancy is that portion of data that can be removed when it is not needed or can be reinserted to interpret the data when needed. Most often, the redundancy is reinserted in order to generate the data in its original form. A technique to reduce the redundancy of data is defined as Data compression. The redundancy in data representation is reduced such a way that it can be subsequently reinserted to recover the original data, which is called decompression of the data. Decompression is the inverse process of compression. Specific coders and decoders can be implemented very differently.

### **COMPRESSION FOR LOW-POWER**

It is well-known that wireless data communication takes up a large share of the total power consumption in most portable wireless devices or systems, with power dissipation proportional to the amount of data transferred. By compressing the data prior to wireless transmission, power can be saved provided that the compression operation itself does not consume too much power.

For short-range, low data bandwidth applications the Bluetooth and Zigbee wireless protocols are recommended but for wide range GSM is used. In order to save energy, a common practice is to turn on the transceiver only when data is available for transmission. Thus, when data bandwidth utilization is low, the transceiver spends most of its time in "sleep" mode, minimizing unnecessary energy consumption. Since power-up and power-down overheads are minimal, duty cycling can result in considerable energy savings. From an operational point of view, data compression can reduce the

energy consumption even further by effecting a reduction in the amount of transmission data and essentially the resulting duty cycle.

## **SURVEY OF DATA COMPRESSION ALGORITHMS**

Data compression applications where storage space reduction and hence CR is the primary, and usually only, figure of merit, compression for low overall system power requires the algorithm's space-time complexity to be considered as well. For example, [8] is exceedingly superior over [9] as far as the compression ratio is concerned, but then requires large buffers and numerous computational iterations for training and accurate error modeling. In a hardware implementation point of view, the former is expected to require significantly more memory and computational elements or iterations, resulting in higher leakage and switching power.

### **Huffman Coding**

The basic idea in Huffman coding is to assign short code words to those input blocks with high probabilities and long code words to those with low probabilities. This concept is similar to that of the Morse code. All of these methods are defined word schemes which determine the mapping from source messages to code words based upon a running estimate of the source message probabilities. The code is adaptive, changing so as to remain optimal for the current estimates. In this way, the adaptive Huffman codes respond to locality.

A Huffman code is designed by merging together the two least probable characters, and repeating this process until there is only one character remaining. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree.

### **Arithmetic Coding**

Arithmetic coding is a form of entropy encoding used in lossless data compression. Normally, a string of characters such as words hello there is represented using a fixed number of bits per character, as in the ASCII code. When a string is converted to arithmetic encoding, frequently used characters will be stored with fewer bits and not-so-frequently occurring characters will be stored with more bits, resulting in fewer bits used in total. Arithmetic coding differs from other forms of entropy encoding in that rather than separating the input into component symbols and replacing each with a code, arithmetic coding encodes the entire message into a single number. Arithmetic coding is especially suitable for small alphabet (binary sources) with highly skewed probabilities. Arithmetic coding is very popular in the image and video compression applications. There are a few disadvantages of arithmetic coding. One is that the whole codeword must be received to start decoding the symbols, and if there is a corrupt bit in the codeword, the entire message could become corrupt. Another is that there is a limit to the precision of the number which can be encoded, thus limiting the number of symbols to encode within a codeword.

### **Lempel-Ziv Welch Coding**

The Lempel-Ziv algorithm is a variable-to-fixed length code. Basically, there are two versions of the algorithm presented in the literature: the theoretical version and the practical version. Theoretically, both versions perform essentially the same. However, the proof of the asymptotic optimality of the theoretical version is easier. In practice, the practical version is easier to implement and is slightly more efficient.

### **Golomb Rice**

Golomb coding is a lossless data compression method using a family of data compression codes invented by

Solomon W. Golomb in the 1960s. Golomb coding is highly suitable for situations in which the occurrence of small values in the input stream is significantly more likely than large values. Rice coding denotes using a subset of the family of Golomb codes to produce a simpler prefix code. Golomb code has a tunable parameter that can be any positive integer value, Rice codes are those in which the tunable parameter is a power of two. This makes Rice codes convenient for use on a computer, since multiplication and division by 2 can be implemented more efficiently in binary arithmetic.

## ALGORITHM FOR LOW-POWER COMPRESSOR AND DECOMPRESSOR

### Algorithm of Golomb Rice Compressor

- Fix the parameter M to an integer value.
- For N, the number to be encoded, find
- Quotient =  $q = \text{int}[N/M]$
- Remainder =  $r = N \text{ modulo } M$
- Generate Codeword
- **The Code format:** <Quotient Code><Remainder Code>, where
- Quotient Code (in unary coding)
  - Write a q-length string of 1 bits
  - Write a 0 bit
- Remainder Code (in truncated binary encoding)
  - If M is power of 2, code remainder as binary format. So bits are needed. (Rice code)
  - If M is not a power of 2, set
  - If code r as plain binary using b-1 bits.

If code the number in plain binary representation using b bits.

### Algorithm of Golomb Rice Decompressor

Decompression is the reverse of data compression that is restoring back the original data that was compressed for either transmission or storage.

The algorithm for Golomb Rice Decompression is explained below

For M not equal to  $2^k$

- M value is known
- Calculate  $b = \log_2 M$
- Calculate  $x = 2^b - M$

To calculate quotient

- Find out number of 1's before 1st 0 from MSB. The count gives the quotient value (q).

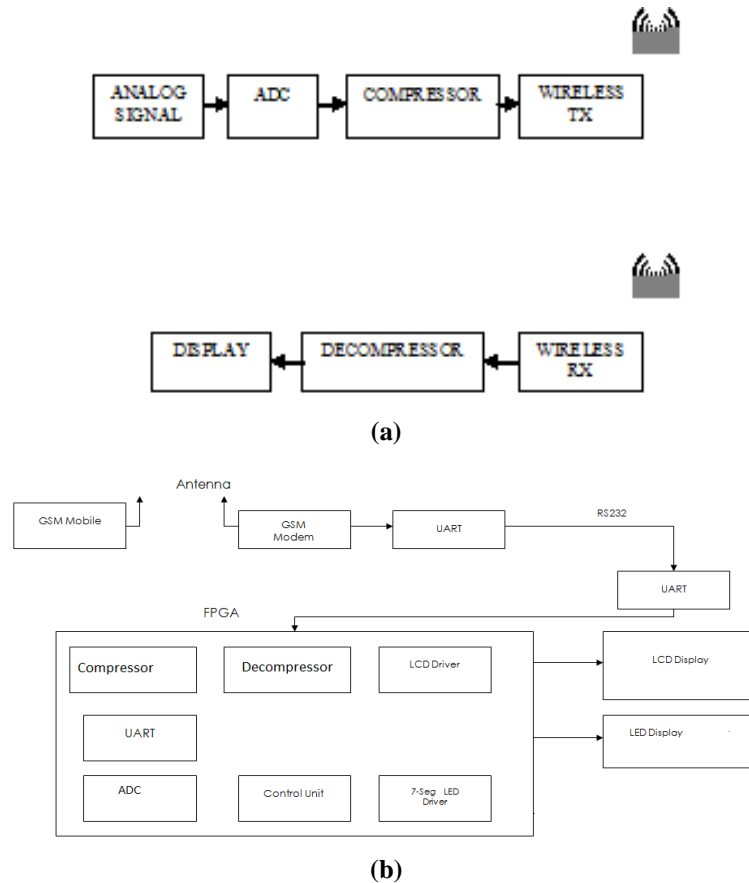
To calculate remainder

- Exclude  $q+1$  MSB bits and read the remaining bits.
  - If that value  $< x$  then  
Rem = value.
  - If value  $\geq x$ , then  
Rem = value -  $x$ .

To Calculate decoded value

- If quotient=0  
Decoded value=rem value
- If quotient  $\neq 0$   
Decoded value= $q * M + \text{rem}$   
For M equal to 2 power k  
Determine Q by counting the number of 1s before the first 0.  
Determine R reading the next K bits as a binary value.  
The decoded value is  $q \times M + \text{rem}$ .

**HARDWARE IMPLEMENTATION**



**Figure 1: (a) General Block Diagram of the Transmitter and Receiver for Compressed Data Communication (b) FPGA Based Transmitter and Receiver Using GSM Based Transmission**

Figure 1 shows the FPGA implementation compressor and decompressor unit used in transmitter and receiver using GSM technique. The ADC converts the analog signal into digital and the synchronizing bit (M) between the encoder and decoder is given through dip switch. The 8 bit digital data from the ADC is fed to the VLSI compressor block. The encoding process takes place according to the algorithm and converts the 8 bit input stream into 6 bit compressed output block. The encoded message is displayed on a LCD screen and the same message is sent as an SMS to the user. The encoded data can be directly sent to the receiver but in order to check the correctness of the system we are using a mobile device.

To cover a long range for data lossless transmission we have chosen GSM technology which operates at 900/1800 MHz and the compressed bits are wirelessly transmitted by sending an SMS through the GSM module.

At the receiving side we receive the data through GSM module which is interfaced with FPGA Spartan-3E technology. The data obtained here is lossless and the synchronizing value, M that was given at the transmitting station is maintained at the receiving station and the value is fed to the golomb rice decode whose output is lossless and same as the ADC output. The sender at the receiver side can send an SMS and this is received by the monitoring station and then forwarded to the decompressor module where decompression of the data takes place and the decompressed data is displayed on the LCD.

The developed algorithm is implemented in hardware and for the lossless compression of analog signals. The input to the system is fed through a potentiometer and a dip switch. This signal is given to an 8 bit ADC (IC 0808) The ADC block converts it into 8 bit digital output. The digital output is displayed on the LED. This digital output is given as an input to the GR compressor block. The compressor compresses the data to 6 bits. The encoded data is transmitted through UART to the GSM module which then sends a SMS. On the other end the mobile user sends the encoded value. This value is again transmitted via GSM to the FPGA. The GR decompressor block decodes the message and displays it on the LCD.

## WIRELESS TRANSMITTER AND RECEIVER

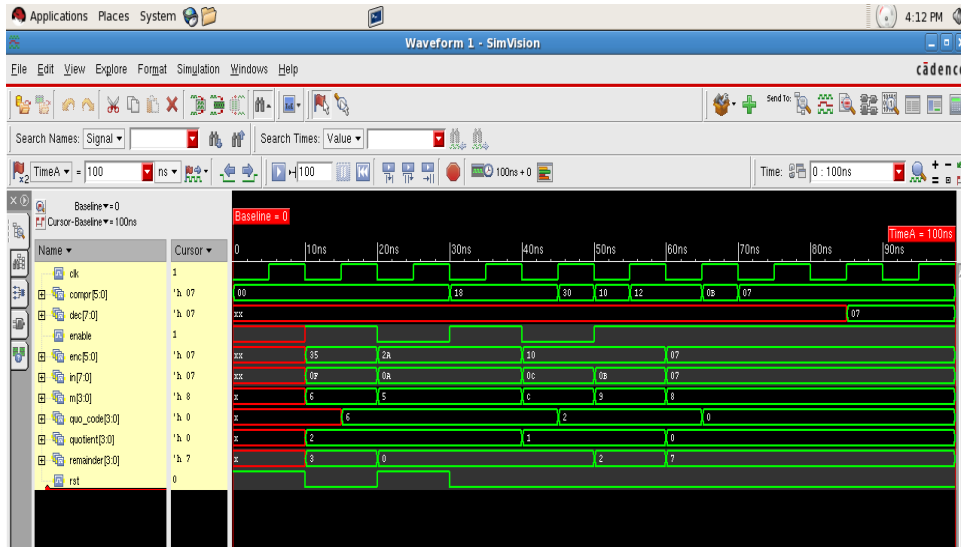
To cover a long range for data lossless transmission we have chosen GSM technology which operates at 900/1800 Mhz and the compressed bits are wirelessly transmitted by sending an SMS through the GSM module.

A GSM modem is a wireless modem that works with GSM wireless networks. A wireless modem is similar to a dial-up modem. The main difference is that a wireless modem transmits data through a wireless network whereas a dial-up modem transmits data through a copper telephone line. Most mobile phones can be used as a wireless modem. To send SMS messages, first place a valid SIM card into a GSM modem, which is then connected to a microcontroller, you can control the GSM modem by sending instructions to it. The instructions used for controlling the GSM modem are called AT commands. GSM modems support a common set of standard AT commands. In addition to this, GSM modems support an extended set of AT commands. One use of the extended AT commands is to control the sending and receiving of SMS messages.

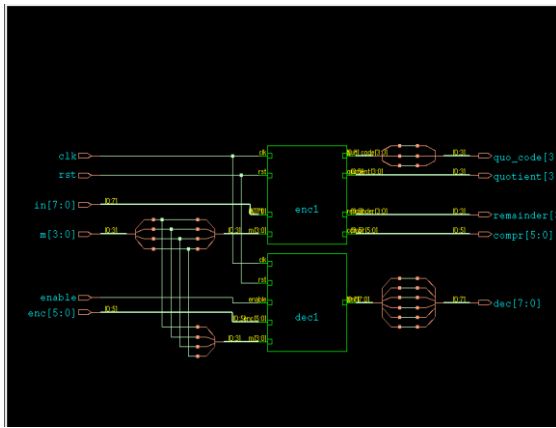
Allows user to verify and edit the message before sending to ensure the command issued is the command requested. Data is transmitted in binary, requires less hardware to manipulate (as opposed to DTMF) Most commonly used in M2M technology.

## RESULTS AND DISCUSSIONS

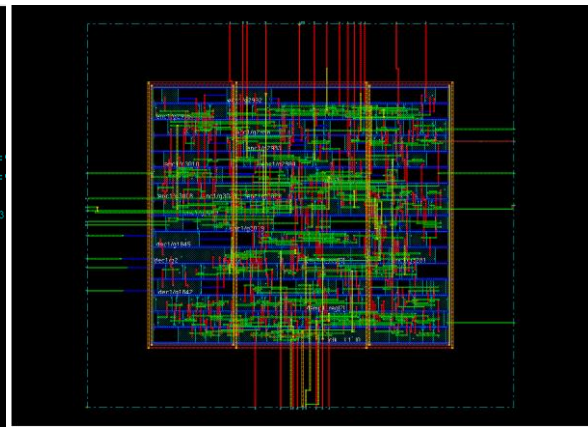
We have Implemented the design using FPGA Spartan 3E and verified the design's real time functionality. The same design we implemented for the ASIC flow using cadence design tools with tsmc 180nm technology. The functionality of the design is verified using the Ncsim for the frequency of 100 MHz. The simulated output



(a)



(b)



(c)

**Figure 2: Implementation of Compressor and Decompressor in Cadence Tools  
(a) Simulated Output (b) Synthesized Circuit (c) Floor Planned Design**

Of the compressor and Decompressor unit combined as a single unit and shown in figure 2 a. The same design is synthesized using cadence RTL compiler and placed and routed using Virtuoso and they are shown in figure 2b and figure 2c Respectively.

As a conclusion in this paper GR algorithmic code compressor and decompressor unit is implemented using GSM technique. The design is implemented using cadence tool starting from RTL to GDSII. The power and area analysis is done and the results are observed and listed in Table 1. The area increased for compressor unit by 15.51/ and the switching power reduced by 36.31%.

**Table 1: Comparison between Huffman and GR Algorithm in 180 nm Technology**

Techniques	Area(No of Cells)	Net Power(nw)	Switching Power(nw)
Huffman	6524	33165.83	168179.30
GR algorithm	7690	24085.56	123380.02

**REFERENCES**

1. A Low Power Biomedical Signal Processing System-on-Chip Design for Portable Brain-Heart Monitoring Systems. Chih-Chung Fu, Shao-Yen Tseng, and Shih Kang, Department of Electronics Engineering National Chiao Tung University Taiwan (R.O.C.).IEEE 2010

2. Luke Rankine, Nathan Stevenson, Mostefa Mesbah, and Boualem Boas hash, "A Non-stationary Model of Newborn EEG", IEEE Trans. Biomed. Eng., VOL. 54, NO. 1, JANUARY 2007.
3. R. Vigarito, I. Sarela, V. Jousmaki, M. Hamalainen, and E. Oja, "Independent component approach to the analysis of EEG and ME Recordings," IEEE Trans. Biomed. Eng., vol. 47, no. 5, pp. 589-593 May 2000.
4. Kuo-Kai Shyu and Ming-Huan Li, "FPGA Implementation of Fast ICA based on Floating-Point Arithmetic Design for Real Time Blind Source Separation," 2006 International Joint Conference on Neural Networks, Vancouver, BC, Canada July 16-21, 2006. A. Hyvarinen and E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis," Neural Comput., vol. 9, pp.1483-1492, 1997.
5. Seeck, Roger (2008). Binary Essence. Retrieved April 17, 2008, from About Binary Essence Web site: <http://www.binaryessence.com/dct/en000081.htm> Graefe, Author's first name initial G, & Shapiro, L (1991). ACM/IEEE-CS Symp. Data Compression and Database Performance. 1, 1-10.
6. Abdullah, H.; Holland, G.; Cosic, I.; Cvetkovic, D.; "Correlation of sleep EEG frequency bands and heart rate variability," Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, pp.5014-5017, 3-6 Sept. 2009.
7. Fidopiastis, Cali; Hughes, Charles, "Workshop 1: Use of psycho physiological measures in virtual rehabilitation," Virtual Rehabilitation, 2008, pp.xi-xi, 25-27 Aug. 2008
8. Fang, Wai-Chi; Chen, Chiu-Kuo; Chua, Ericson; Fu, Chih-Chung; Tseng, Shao-Yen; Kang, Shih; "A low power biomedical signal processing system-on-chip design for portable brain-heart monitoring systems," Green Circuits and Systems (ICGCS), 2010 International Conference on vol., no., pp.18-23, 21-23 June 2010
9. Yates, D.C.; Rodriguez-Villegas, E.; "A Key Power Trade-off in Wireless EEG Headset Design," Neural Engineering, 2007. CNE '07. 3rd International IEEE/EMBS Conference on, pp.453-456

