K Chandra Naik, J. Chinna Babu, Dr.K.Padmapriya, Dr V.R.Anitha / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 2, March -April 2013, pp.224-229

Implementing NAND Flash Controller using Product Reed Solomon code on FPGA chip

K Chandra Naik , J. Chinna Babu, Dr.K.Padmapriya, Dr V.R.Anitha

*(M.Tech VLSI SystemDesign) AITS, Rajampet, KADAPA(Dt),
**Asst.Prof, Dept'of ECE, AITS,Rajampet, KADAPA(Dt),
***Asst Prof in JNTUCE, Anantapur, A.P.
****Prof in ECE, SVEC in Tirupathi,AP.

Abstract

Reed–Solomon (RS) codes are widely used to identify and correct errors in storage systems and transmission and. When RS codes are used for so many memory system and reduces error in data. (255, 223) product Reed-Solomon (RS) for non-volatile NAND flash memory systems. Reed-Solomon codes are the most used in digital data storage systems, but powerful for tool burst errors . To correct multiple random errors and burst errors in order, The composing of product code in to column-wise RS codes and row-wise RS codes may allow to decode multiple errors beyond their error correction capability. The consists of proposed code is two shortened RS codes and a conventional Reed-Solomon code .The non-volatile NAND flash Controller memory systems.

Reed-Solomon codes are the most Powerful used in data storage systems. The proposed coding scheme on a FPGA-based simulator with using an FPGA device. The proposed code can correct 16 symbol errors.

Keywords: Product code, NAND flash controller memory, correction error code ,FPGA; Reedsolomon code.

I. INTRODUCTION

Digital communication system is used to transport an information bearing signal from the source to a user destination via a channel. In this channel Produces some error by using RS coding detects and corrects errors in the communication system. Non-volatile NAND flash controller memory systems are widely used in the mobiles and wireless systems. The main requirement of the NAND flash controller is makes high density and low cost, the operation speed increased and simultaneously creates various types of errors. series of blocks is grouped in to The NAND Flash array , which are the erasable in small .entities in a NAND Flash device .A NAND Flash block is 128KB. Erasing a block sets all bits to 1 .Programming is necessary to change erased bits from 1 to 0. programmed is a byte. Some NOR Flash memory can perform READ-While-WRITE operations. The multi-leveling cell (MLC), even though supplying powerful solutions, the memory storage increases performance of the systems and causes many errors. In MLC, multiple bits are storing per a memory cell by each programming cell with multiple threshold levels.. MLC NAND flash memories, Bose-Chaudhuri-Hocquenghem (BCH) codes are frequently used. The BCH codes provide flexible code length and variable range of correcting the errors capability. propose a product code with using a Reed-Solomon code scheme for NAND flash memories. The correct errors proposed code by the burst error as well as by the multiple random error . The proposed code lower decoding complexity than that of a BCH code with the considerable decoder code rate. Then employ three (255, 247) RS decoders to improve error rate against multiple random errors.

2. PROPOSED PRODUCT CODES

A Reed-Solomon (RS) code is constructed in a Galois field. (GF(2m). A RS code is a block code and can be specified as a cyclic (n, k) RS. The variable 'n' is the size of codeword by the 'k ' is the number of data symbol and the number of parity symbols is '2t'. Each symbol contains 'm' number of bits. The relationship between the size of symbol 'm' and the size of the codeword 'n' is given by $n=2^m-1$

That is 'm' bits in one symbol, there could exist $'2^m-1$ distinct symbols in one codeword. The RS code allows correcting up to t number of symbol Errors where t is given by

$t = (n-k) / 2.$

The codeword is represented as

$c(X) = c0 + c1x + c2x^2 + \ldots + cn\text{-}1X^{n-1}$ (ci € $GF(2^m)$).

$c(X) = c0 + c1X + c2X2 + \ldots + cn\text{-}1X^{n-1}$(ci) $GF(2^m)$).

g(X) is generator polynomial that represents as below.

g(X) = (X – α)(X – α2)(X – α3)…(X – α2t) =
g0 + g1X + g2X2 + … + g2t-1X2t-1 + X2t (gi α GF(2m)).

Implementing encoding procedure is dividing Xn-km(X) by g(X), which is written Xn-km(X) = q(X)g(X) + p(X) where, q(X) and p(X) are quotient and remainder polynomials, respectively. Encoding procedure is implemented dividing by Xn-km(X) by g(X), i.e., written as Xn-km(X) = q(X)g(X) + p(X) where, q(X) and p(X) are quotient and remainder polynomials,
Respectively

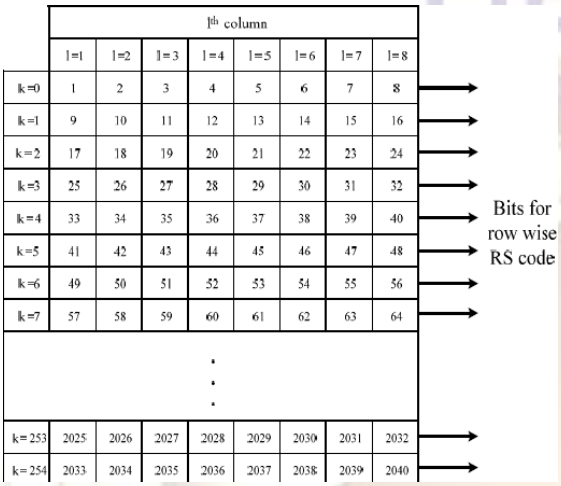


Fig : The block diagram of transferring row-column data

After first encoding, containing of the encoded data extra 7symbols are sent to a (255, 247) conventional Reed Solomon encoder. The RS encoder processes of the conventional encoder receives data in different format, compared to the sRS encoder. Therefore, before second encoding, the 247 data symbols transfer block is must be rearranged. This transfer processing is different from first one. the shortened RS codes are transfers block code after encoder rearranges column wise to a row wise sequential bit block code .

3. NAND FLASH CONTROLLER

The NAND Flash array is grouped into a series of blocks, the smallest erasable entities in a NAND Flash device.A NAND Flash block is 128KB. Erasing a block sets all bits to 1 . Programming is necessary to change erased bits from 1 to 0. The smallest entity that can be programmed is a byte. Some NOR Flash memory can perform READ-While-WRITE operations. The READ and WRITE operations are cannot perform simultaneously in NAND Flash memory; it is possible to accomplish READ/WRITE operations at the system level using a method called shadowing .in personal computer used in this Shadowing process for many years to load the BIOS from the slower ROM into the higher-speed RAM.

3.1. NAND FLASH ARCHITECTURE AND BASIC OPARATION

The NAND Flash device is organized as 2048 blocks, with 64 pages per block Each page is 2112 bytes, consisting of a 64-byte spare area and a 2048-byte data area . The spare area is typically used for ECC, wear-leveling, and other software overhead functions, although it is physically the same as the rest of the page. Many NAND Flash devices are offered with either an 8- or a 16-bit or 32-bit interface.

Host data is connected to the NAND Flash memory via an 8-bit- or 16-bit-wide bidirectional data bus. For 16-bit devices, commands and addresses use the lower 8 bits (7:0). The upper 8 bits of the 16-bit data bus are used only during data-transfer cycles

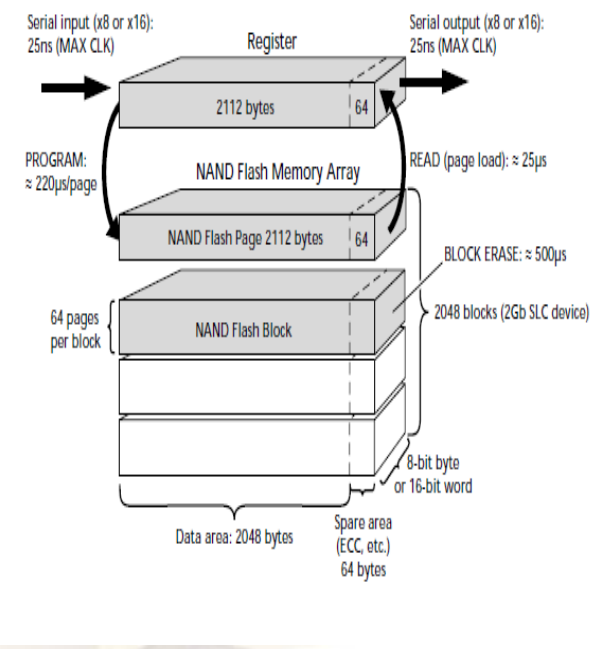


Fig: 2Gb NAND Flash Device Organized as 2048 Blocks

Table : Signal Descriptions

Symbol	Signal	Description
ALE	Address latch enable	When ALE is HIGH, adresses are latched into the NAND Flash address register on the rising edge of the WE#signal.
CE#	Chip enable	If CE is not asserted, the NAND Flash device remains in standby mode and does not respond to any control signals
CLE	Command latch enable	When CLE is HIGH, commands are latched into the NAND Flash command register on the rising edge of the WE# signal.R
R/B#	Ready/busy#	If the NAND Flash device is busy with an ERASE, PROGRAM, or READ peration, the R/B# signal is asserted LOW. The R/B# signal is open drain and requires a pull-up resistor
RE#	Read enable	RE# enables the output data buffers.
WE#	Write enable	WE# is responsible for clocking data, address, or commands into the NAND Flash

Table2: Advantages of NAND and NOR Flashs

	NAND	NOR
Advantages	Fast PROGRAMs	Random access
	Fast ERASEs	Byte PROGRAMs possible
Disadvantages	Slow random access	Slow PROGRAMs
	Byte PROGRAMs difficult	Slow ERASEs
Applications	File (disk) applications	Replacement of EPROM
	Voice, data, video recorder	Execute directly from nonvolatile memory
	Any large sequential data	

4. ENCODER

The encoder is architected using the Linear Feedback Shift Register Design. The coefficients.The $g_i,\ 0 \le i \le 15,$ Berlekamp-Massey algorithm based on equation .i.e

g(x) = x16 + 59x15 + 13x14 + 104x13 + 189x12 + 68x11 + 209x10 + 30x9 + 8x8 + 163x7 + 65x6 + 41x5 + 229x4 + 98x3 + 50x2 + 36x + 59 (12)

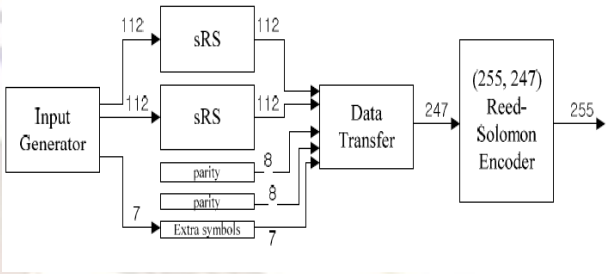


Fig: The block diagram of the proposed product encoder.

The encoder consists of a structure using (255,247) RS codes that have four error correcting capability. That is First part composes two (120, 118) shortened RS codes that column wise is encode input data . #1 shortened RS encoder processes 112 input data symbols from 1st to 112th, and #2 shortened RS encoder processes from 113th to 224th etc. The sRS encoder fills up 135 symbols with symbols '0'. Two RS codes shortened have a format of 255 byte codeword despite of containing 120 symbols of message. For this process,

the input data transferred data are rearranging in row wise sequence into a column wise vector before encoding into shortened RS codes as shown in fig. the restructure each message data for an encoder and decodes in reverse sequence. For a shortened RS code .In encoder the error will be detected, the errors in different form s.

5 . DECODER

A basic diagram for decoding Reed-Solomon codes is shown in the following diagram

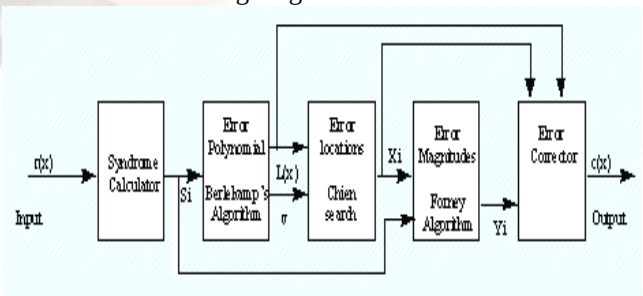


Fig: block diagram of Encoder.

Keys:

r(x) Received codeword
Si Syndromes
L(x) Error locator polynomial
Xi Error locations
Yi Error magnitudes
c(x) Recovered code word
V Number of errors

The output of the received codeword r(x) is the original (transmitted) codeword c(x) plus errors:
r(x) = c(x) + e(x)
To identify the attends of Reed Solomon code the position and magnitude of up to t errors and to correct the errors or erasures.

5.1 Syndrome Calculation

This is a similar calculation to parity calculation. Reed-Solomon codeword has 2t **syndromes** that depend only on errors . The syndromes can be calculated by substituting the 2t roots of the generator polynomial g(x) into r(x).
(i) Finding the Symbol Error Locations
The symbol error involves solving simultaneous equations with t unknowns. Several fast algorithms are available to do this. These algorithms take advantage of the special matrix structure of Reed-Solomon codes and greatly reduce the computational effort required. In general two steps are involved.
(ii)Find an error locator polynomial
The error locator polynomial can be done using the Berlekamp-Massey algorithm or Euclid's algorithm.this Euclid's algorithm tends to be more widely used in practice because it is easier to implement, however, the Berlekamp-Massey algorithm tends to more efficient hardware and software implementations. This is done using the Chien search algorithm.
(iii)Finding the Symbol Error Values
This involves solving simultaneous equations with t unknowns. A widely-used fast algorithm is the Forney algorithm.
(iv)Finding the Symbol Error Values
The symbol error values involves solving simultaneous equations with t not known's. the Forney algorithm is widely-used for fast algorithm

6. REED SOLOMON CODES

A Reed-Solomon code is a block code and can be as RS(n,k) as shown in Fig. 2. The n is the size of the codeword with the unit of symbols it is variable, the number of data symbols is k and 2t is the number of parity symbols Each symbol contains s number of bits.

The relationship between the size of the codeword is n ,and the symbol size, s, is given by (1). This means that if there are s bits in one symbol, there could exist 2s−1 distinct symbols in one codeword, excluding the one with all zeros.

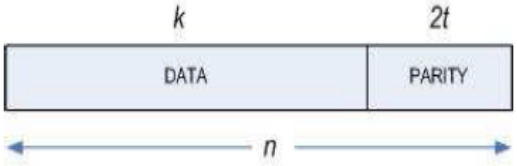


Fig. 2. the structure of a RS codeword

n = 2s − 1
The Reed Solomon code allows correcting up to t number of symbol errors where t is given by

$$t = \frac{n-k}{2}$$

A. Galois Field

The Reed-Solomon code is defined in the Galois field, GF are contains a finite set of numbers where any arithmetic operations and logic on elements of that set will result in an element belonging to the same set. Every element, except zero, can be expressed as a power of a primitive element of the field. For example, a Galois field, GF(8), is built with the primitive polynomial p(z) = z3+z+1 based on the primitive element = z.

TABLE I
GALOIS FIELD, GF(8)

Exponent	Polynomial	Binary
α^0	1	001
α^1	z	010
α^2	z^2	100
α^3	$z+1$	011
α^4	z^2+z	110
α^5	z^2+z+1	111
α^6	z^2+1	101
$\alpha^7=\alpha^0$	1	001
$\alpha^8=\alpha^1$	z	010

7. IMPLEMENTATION OF REED-SOLOMON ENCODER AND DECODER

A number of commercial Many existing systems use "off-the-shelf" integrated circuits that encode and decode Reed-Solomon codes. RS code systems ICs tend to support a certain amount of programmability (for example, RS(255,k) where t = 1 to 16 data symbols). A recent trend is towards VHDL or Verilog designs (**intellectual property core or logic cores s**). These code have a number of advantages over standard ICs. A logic core can be VHDL or Verilog integrated with other components and synthesized to an FPGA (Field Programmable Gate Array) or ASIC (Application Specific Integrated Circuit) – this enables so-called "System on Chip" designs where multiple modules can be combined in a single IC, Depending on logic core

and production volumes can often give significantly small cost system costs than standard ICs. By using logic cores, a designer to avoids the potential need to do a "lifetime buy" of a Reed-Solomon IC.

7.1 Software Implementation

Software implementations in "real-time" required too much computational power for all but the simplest of Reed-Solomon codes. In software is that general purpose processors Galois field arithmetic operations do not support. Example, to implement a Galois field multiply in software requires a test for 0, two log table look-ups, modulo add and anti-log table look-up. careful design together with increases in processor performance means that software implementations can operate at relatively high data rates. The following table gives some example figures on a 166MHz Pentium PC

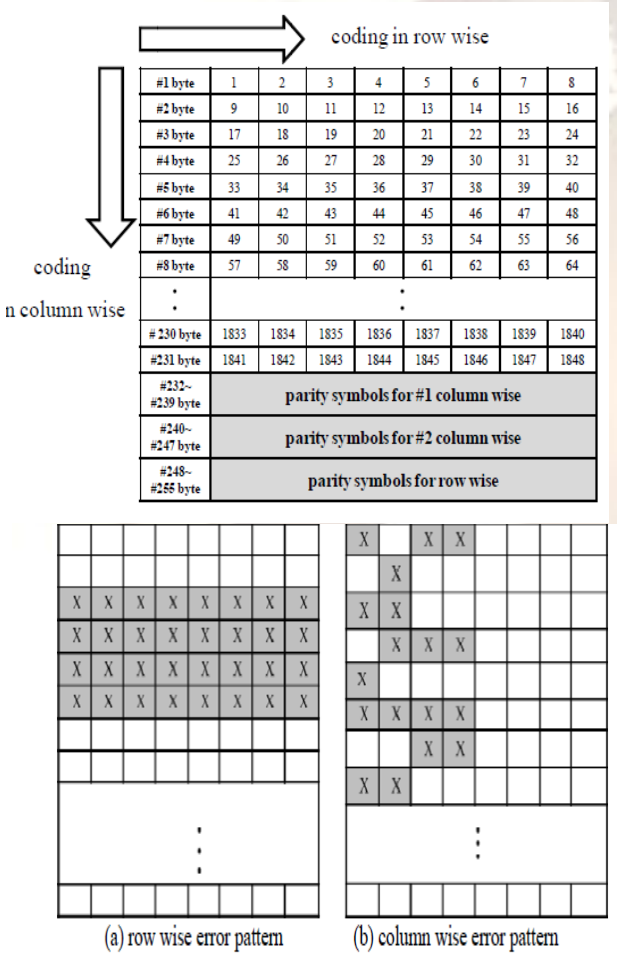


Fig: structure of proposed RS encoder

8. FPGA IMPLEMENTATION

The proposed product RS code is implemented by an Altera's Stratix II FPGA chip, that contains 48,352 adaptive look-up tables and 2Mbit of

on-chip memory. The operation is measured Tektronix's TLA 5203 Logic Analyzer. The entire circuits are designed by a verilog HDL language using an Altera's Quartus II 7.0 program. The implementation experiment shows that the proposed code have 43,501 gates and 554.5 clock latency as shown in fig. The coding controller operates 1.07 Gbps at the maximum frequency of 300 MHz and consumes 26.4 mW in 1.2 V operation voltages. Summarizes the comparison with other results reported. The proposed code shows the improved error correction capability and coding gain over a conventional RS code. It has faster operation frequency and more improved bandwidth than other coding schemes it provide more imprudent in correction and detection the data. In comparison to other codes. A BCH codes, hamming codes, and a ordinary RS code such as the proposed code has more powerful error correction capability.

ECC	Gate count		Latency (clk)	Operation Freq. (MHz)	Bandwidth (Gbps)	Power (mW)
Proposed Product Code	Total	43,501	554.5	290	1.07	26.44
	sRS	6,075 6,089				
	Conventional RS	6,080				
	Transfer for sRS	7,880				
	Transfer for RS	17,337				

Table : Data for FPGA implementation of the proposed RS codes.

9. SIMULATION RESULTS

which contain real data captured on modelsim, it is combines high performance and high capacity with the most advanced code coverage and debugging capabilities in the industry. ModelSim offers unmatched flexibility by supporting 32 and 64 bit UNIX and Linux.As shown in fig the simulation result,

Fig: The simulation output waveform of the top module of Reed Solomon.

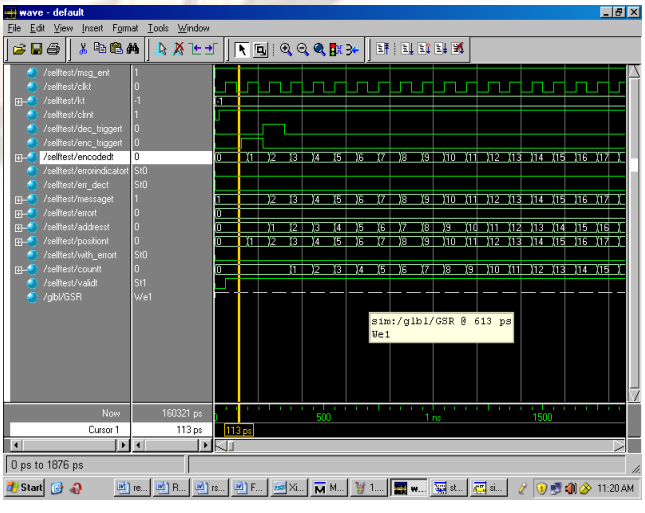
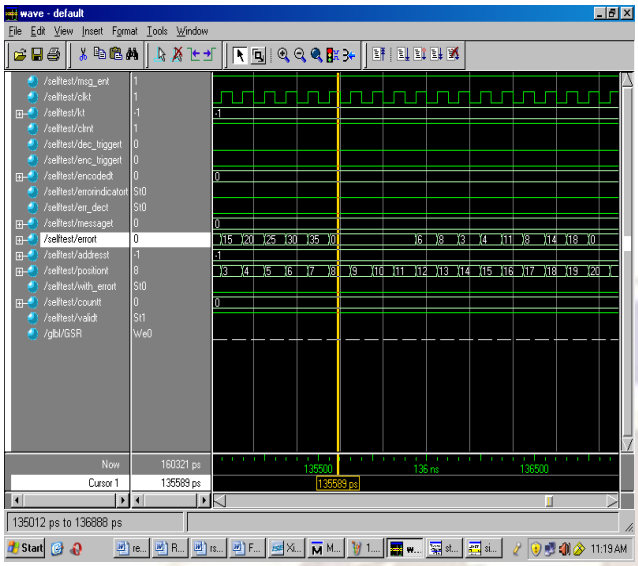


Fig: The simulation output waveform of the top module of Reed Solomon



10. SYNTHESIS REPORT

Table Device Utilization Using Xilinx Spartan3E.

Logic unit	Used	Available	Utilization
Number of Slices	2885	4656	61%
Number of Slice Flip Flops	3557	9312	38%
Number of 4 input LUTs	4499	9512	48%
Number used as logic	3343		
Number used as Shift registers	444		
Number used as RAMs	712		
Number of Ios	23		
Number of bonded IOBs	23	232	9%
Number of BRAMs	9	20	45%
Number of MULT18X18SIOs	12	20	60%
Number of GCLKs	3	24	12%

In this table shows the number of lagics and memory cells used.

11 . CONCLUSION

The RS Codes are proposes a (255, 247) product Reed-Solomon code for multiple burst errors and random errors . The proposed code provides data consistedof two shortened Reed-Solomon codes in a column-wise and a conventional Reed-Solomon code in a row-wise by using two dimensional array. The proposed code becomes powerful against multiple burst errors and random error. The proposed code can corrects 16 symbol to 32 bit symbols errors. The code has the coding gain of 1.8 dB and the bandwidth of 1.07 Gbps when operated at 290 MHz with the power consumption of 26.4 mW.

REFERENCES

1. R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading :3rd edition Addison-Wesley Publishing Company, 1983.
2. A. R. Masoleh and M. A. Hasan, "Low complexity bit parallel architectures for intel polynomial basis multiplication over GF(2m), computers," *IEEE Trans. Comput.*, vol. 53and 55, no. 8, pp. 945–959, Aug. 2004.
3. J. Gambles, L. Miles and J. Has, W. Smith, and S. Whitaker, "An ultra-low power, radiation-tolerant reed Solomon encoder for space applications," in. *IEEE Custom Integr. Circuits Conf.*, 2003, pp. 631–634.

Chandra Naik .K has received his B.Tech in ECEI in 2010 from ACET ,Allagadda and current pursing M.Tech in VLSI System design from AITS,Rajampet under the guidance of Mr J.Chinna Babu his field of interesting Digital Signal Processing and VLSI design.

Mr.J.Chinna Babu has received his M. Tech degree in VLSI System Design. Currently, he is working as Assistant Professor in the Department of Electron-ics & Communicati on Engineering,AITS,Rajampet, Kadapa, A.P, and India. He has published a number of research papers in various National and International Journals and Conferences. His areas of interests are VLSI, Micro processor, Embedded Systems and Signals and Systems.

Dr.K.Padmapriya, Assistant professor in JNTUCE, Anantapur, A.P, India

Mr .Arun Kumar,Assistant professor ,Dept of IT In AITS,Rajampet, Kadapa,A.P,India.

Dr V.R.ANITHA, Prof in ECE,SVEC in Tirupathi,AP