state column by column, considering each column as a four-term polynomial similar to mix column transformation.

$$\begin{bmatrix} d_{00} & d_{01} & d_{02} & d_{03} \\ d_{11} & d_{12} & d_{13} & d_{10} \\ d_{22} & d_{23} & d_{20} & d_{21} \\ d_{33} & d_{30} & d_{31} & d_{32} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \otimes \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

Figure 2.6: Inverse Mix Columns

3. Purposed Hamming Tolerant Model

3. 1 Model Description

The purposed model is based on single error correcting Hamming code. A single bit fault [8] in a byte is detected and corrected by Hamming code. At the end of the each transformation the hamming code is predicted from the Hamming code table. The EDC (error detection and correction) model is based upon the predicting the hamming code bits at the end of each transformation from hamming code memory table.

3. 2 Hamming Code Calculation

The hamming code of the each byte of the S-ox look up table is stored in the hamming code memory. Following are the procedure to calculate the hamming code bits. The parity check bits of each byte of the s-box look up table are pre-calculated.

h (SRD [a])--> hRD [a] (A)
h ((SRD [a] f{2g})--> h2RD [a]

Where a=state byte

h ((SRD [a] f{03g})--> h3RD [a] (1)

h=calculation represent humming code

h_{RD}, h_{2RD} and h_{3RD} is the Hamming code of the S-box LUT (S_{RD}), (S_{RD}@{02}) and (S_{RD}@{03}) respectively.

Each state byte can be represented by bits b_0, b_1, b_3, b_4, b_5, b_6, b_7, b_8. 4-bit hamming code of the state byte is represented by bits (h_3, h_2, h_1, h_0)

$H_3 = b_7$ xor b_6 xor b_4 xor b_3 xor b_1
$H_2 = b_7$ xor b_5 xor b_4 xor b_2 xor b_1
$H_1 = b_6$ xor b_5 xor b_4 xor b_0
$H_0 = b_3$ xor b_2 xor b_1 xor b_0

The hamming code for S_{RD} table are pre-calculated and stored in the hamming memory table (known as h_{RD} table).

The h_{2RD} table is obtained by the parity check bits of (S_{RD}× {02}). The galois field multiplication of a state byte, a, with {02}.

h_{2RD} = h ({02}× a)

The h_{3RD} table is obtained by the parity check bits of (S_{RD}× {03}). The galois field multiplication of a state byte, a, with {03}.

H_{3RD} = h ({03}× a) = h_{RD} XOR h_{2RD}

Hence by using h_{2RD} and h_{RD} we can calculate h_{3RD} table. Finally when we get all the hamming bits, the next action is to detect &correct the faults by predicted hamming code bits.

3.3 Fault Detection and Correction

The Hamming code sub byte transformation matrix is obtained (predicted) by h_{RD} table [8] . The Hamming code shift rows transformation matrix is obtained by simple cyclic rotation of the Hamming code sub byte transformation matrix. The Hamming code mix column transformation matrix is predicted by with the help of just two tables h_{RD} and h_{2RD}. for each transformation predicting Hamming code is obtained by using the parity check bit tables and also the Hamming code is calculated from the output of the transformation. By comparison of predicted and calculated Hamming code we can detect and correct fault as mentioned below.

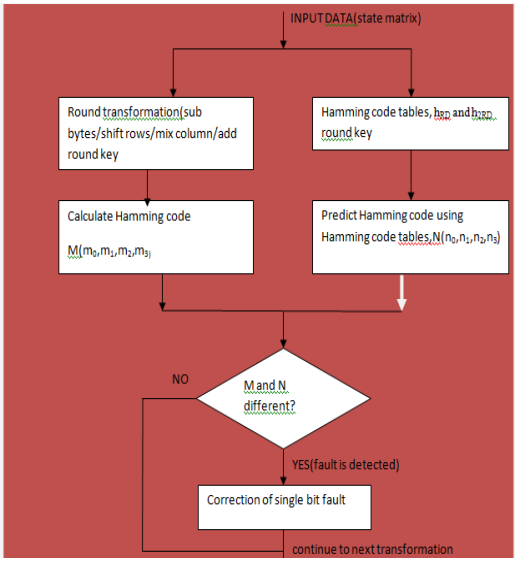

Figure 3.1: Hamming Fault Detection Model

Let the predicted check bits of the transformation input be showed by (m_3, m_2, m_1, m_0) and calculated check bits of the transformation output be showed by (n_3, n_2, n_1, n_0). By comparison of predicted and calculated Hamming check bits location of faulty bit is detected using the table 3. 1. Once the position of faulty bit is identified the fault correction is done by flipping that bit and then encryption process is continued.

Table 3.1: Hamming code Bit Match Table to Locate a Faulty Bit

Hamming code bits comparision	Faulty bit position in output
$(m_3,n_3)\&(m_2,n_2)$	0
$(m_3,n_3)\&(m_1,n_1)$	2
$(m_3,n_3)\&(m_0,n_0)$	5
$(m_2,n_2)\&(m_1,n_1)$	3
$(m_2,n_2)\&(m_0,n_0)$	6
$(m_1,n_1)\&(m_0,n_0)$	7
(m_1,n_1)	1
(m_0,n_0)	4

4. Simulation Result

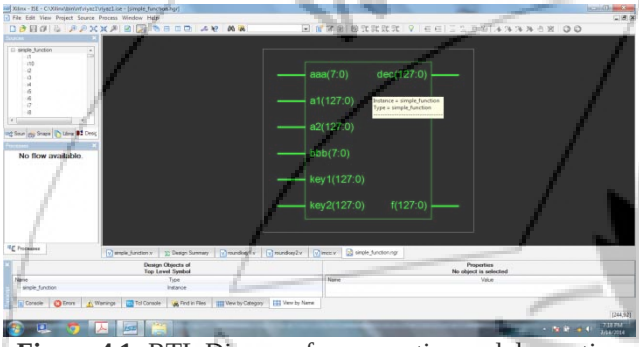

Figure 4.1: RTL Diagram for encryption and decryption

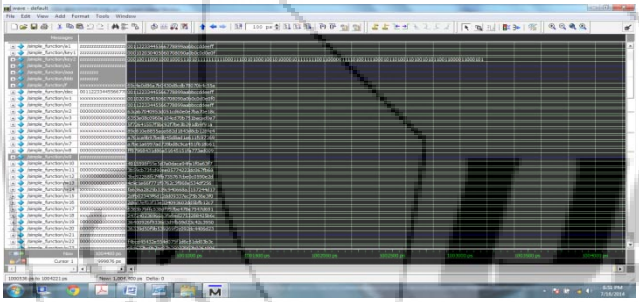

Figure 4.2: Simulation result for encryption and decryption

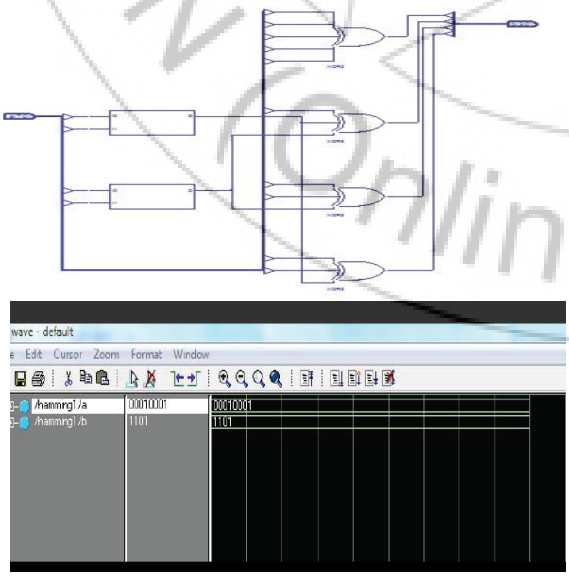

Figure 4.3: RTL Diagram for Hamming code calculation

5. Conclusion and Future Work

5. 1 Conclusion

This paper presents a unique research method of cryptographic algorithms for satellite communication. Satellite operates in radiation environment and encryption processor is sensitive to radiation and due to radiation induced faults. so the encryption algorithm should be free from radiation. to achieve error free encryption, a fault tolerant model has been proposed in this paper.

5.2 Future Work

The proposed model will detect and correct single bit errors. However this model can be extended for multiple bit errors during encryption by using hamming code, Reed-solomon codes etc. This paper was planned to connect the fault tolerant AES IP core to the LEON processor through the bus. But thesis area of research was not covered because of lack of time. The purposed model is internal to the AES algorithm. This model can be extended to implement an EDAC function that is external to the AES algorithm.

References

[1] Horatiu Paul "Twofish Encryption algorithm" conference proceeding 2004
[2] Amandeep Singh "FPGA implementation and analysis of DES and Twofish Encryption algorithms" Thaper university Patiyala, 2010
[3] A. RuhanBevi "an effective symmetric key recovery scheme for secure onboard satellite applications" in IJCSI journal 2011
[4] Mg Suresh "area optimized and pipelined FPGA implementation of AES encryption and decryption" in IJCER volume 2 Issue 7, 2012
[5] AMANDEEP KAMBOJ "high speed parallel concurrent error detection scheme for robust AES hardware" IJAREEIE Volume 2, Issue 10, October 2013
[6] SUMALATHA PATIL "Design of High speed 128 bit AES algorithm for data Encryption" IJCET special issue September 2013
[7] AMRUTHA K" advanced encryption standard algorithm implementation using Verilog HDL" IJVES Volume 4 article 6 July 2013.
[8] PRAVEEN. H. L. "satellite image encryption using AES" IJCSEE Volume 1, Issue 2, 2012
[9] Archana Garg " efficient field programmable gate array implementation of advanced encryption standard algorithm using VHDL" Volume 4, Issue 9, September 2013
[10] Vinoth Vijay " high performance fault detection and correction scheme for advanced encryption standard" IJSETR Volume 2, Issue 4, April 2013

Author Profile

Md. Riyaj received the B. E. degrees in Electronics and Communication Engineering from JNIT Jaipur and currently doing M-Tech VLSI in Suresh Gyan Vihar University Jaipur.

Vipin Gupta received his B. E degree in Electronics and Communication Engineering from SBCET Jaipur in 2009 and M-Tech degree in VLSI from MNIT Jaipur in 2011. He is currently working as a assistant professor in department of electronics and communication Engineering at Suresh Gyan Vihar University Jaipur.