An Efficient VLSI Implementation of Double Error Correction Orthogonal Latin Square Codes

Malavika M. S.
M Tech Student
Electronics and Communication
Nagarjuna college of Engineering
Bengaluru, India

Rohith S.
Assistant professor
Electronics and Communication
Nagarjuna college of Engineering
Bengaluru, India

Venkatesh Kumar H., PhD
Associate professor
Electronics and Communication
Nagarjuna college of Engineering
Bengaluru, India

ABSTRACT
There is a growing interest in multi-bit Error Correction Codes (ECCs) to protect SRAM memories. This has been caused by the increased number of multiple errors that memories suffer as technology scales. To protect an SRAM memory, an ECC has to be decodable in parallel and with low latency. Among the codes proposed for memory protection are Orthogonal Latin Square (OLS) codes that provide low latency decoding and a modular construction. It is more effective to provide different degrees of error correction for the different bits. This is done with Unequal Error Protection (UEP) codes. In this paper, UEP codes are derived from Double Error Correction (DEC) Orthogonal Latin Square (OLS) codes. The derived codes are implemented for an FPGA platform to evaluate the decoder complexity and latency. The Proposed encoder and decoder are done by Verilog HDL and Simulated by ModelSim 6.4 c and synthesized by Xilinx tool.

Keywords
UEP codes, OLS codes, SEC-DED codes, OS-MLD codes

1. INTRODUCTION
Error Correction Codes (ECCs) are widely used to protect memories and other electronic circuits against errors. Traditionally, Single Error Correction Double Error Detection (SEC-DED) codes have been used. These codes can correct single bit errors in any word of the memory and can detect double bit errors, have moderate redundancy in terms of check bits and are relatively easy to decode. Decoding and correction are done via syndrome method which takes single cycle. A special class of SECDED codes known as Hsiao codes [Hsiao 70] was proposed to improve the speed, cost, and reliability of the decoding logic. There are also the double-error correcting triple-error-detecting (DEC-TED) codes, which come at the cost of much larger overhead in terms of both the check bits and more complex hardware to implement the error correction and detection. However in spite of their low check bits overhead and single cycle decoding, SEC-DED codes are not able to provide requisite reliability under certain conditions.

SRAM memories are one of the most commonly used electronic circuits. They are present as standalone devices and also embedded in most Digital Signal Processors (DSPs), microcontrollers, System on Chip (SoCs) and FPGAs Therefore their protection is critical to ensure system reliability. Traditionally, the ECCs used to protect SRAM memories have focused on providing Single Error Correction and Double Error Detection (SEC-DED). To correct multiple bit errors, more advanced ECCs are needed. Although there are many such codes, most of them do not fit the needs of an SRAM memory. To be used with an SRAM memory, encoding and decoding need to be done in parallel in less than one clock cycle.

However, there is a property called "one step majority logic decodable" (OS-MLD) that only a few ECCs have, that makes them suitable for fast parallel encoding. Orthogonal Latin Squares codes are derived from the concept of Latin Square and have been recently proposed to protect interconnects caches and memories. Their main advantage compared to other OS-MLD codes is that they provide a larger number of options in terms of word size and error correction capabilities rest of the parts are organized in section II Block diagram, section III Software requirements, Section IV simulation implementation respectively

2. BLOCK DIAGRAM

2.1 Encoder for the Proposed (48, 16, 16) DEC OLS Code Single Sub-Block Extended With SEC-DED
The encoder can be implemented as a combination of the DEC-OLS and SEC-DED encoders. DES OLS Encoder its Consist of Sixteen 4 input XOR Gate Design. The Inputs are Combination d and Outputs are Co. Another side we have SEC DED Encoder, It Consist of Four 3 Input XOR Gate design in Four Blocks. Both Encoders results are connected to two inputs XOR Gate and the Parity bits are generated. Parity Bits are named as C. This is illustrated in Figure 1. Basically, the encoders can operate in parallel and the final parity check bits are obtained doing the xor of the DEC-OLS results, (co bits in the Figure) and SEC-DED results, (ce bits in the Figure).

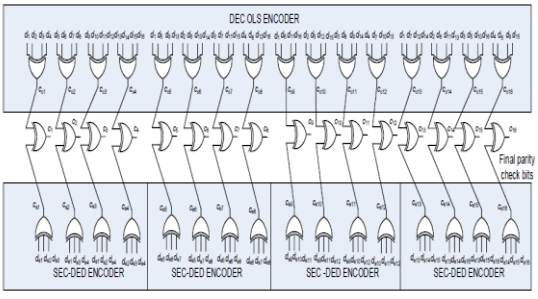

Fig 1 Encoder for the proposed (48,16,16) DEC OLS code single sub-block extended with SEC-DED

2.1.1 Single Mi Sub-Block

OLS codes can be extended by using the parity check bits on each of the Mi sub-blocks to form a new code. One option for DEC OLS codes is to use SEC-DED codes for the extended codes. For example, for the DEC OLS code with kOLS = 16 and n = 32, the sub-blocks have four parity check bits and a SEC-DED code that protects four bits can be formed for each sub-block. Therefore, sixteen additional bits can be protected with SEC-DED this means that a 32-bit data word can be protected with two protection levels, DEC and SEC-DED. Let us denote as kSEC-DED the additional data bits that are added. Then, we have a code with kOLS = 16, kSEC-DED = 16 and n = 48. For the SEC-DED codes, a Hsiao code with columns of weight three can be used so that when a double error occurs there is no miscorrection in the data bits. The obtained parity check matrix is illustrated in Figure 2. It can be observed that the added columns for the SEC-DED codes have all the ones in a single Mi sub-block. Conversely, the columns for the DEC OLS protected data bits have a single one per sub-block as discussed.

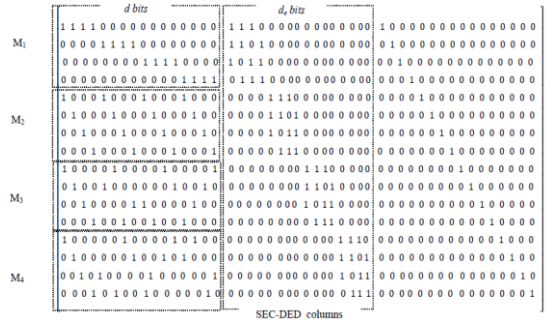

Fig 2 Single mi sub-block Parity check matrix H for OLS with K OLS=16 and n=32

2.2 Decoder for the Proposed (48, 16, 16) DEC OLS Code Single Sub-Block Extended With SEC-DED

The Proposed Decoder Consist of Three Major Units like Syndrome Computation OLS Majority Voters and SEC DED Block.

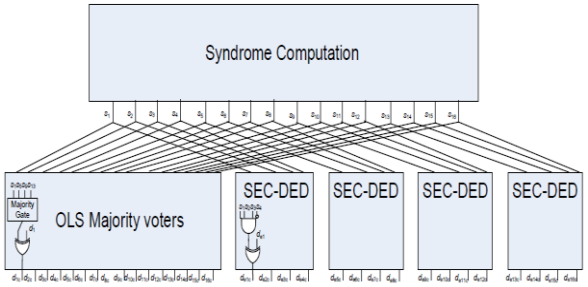

Fig 3 Decoder for the proposed (48,16,16) DEC-OLS code single sub-block extended with SEC-DED

2.2.1 Syndrome Computation Unit

For the syndrome computation, the parity prediction can be implemented by checking that the following two equations take the same value

$r1 = s1 \wedge s2 \wedge s3 \wedge \cdot \cdot \cdot \wedge s2tm$

$r2 = c1 \wedge c2 \wedge c3 \wedge \cdot \cdot \cdot \wedge c2tm$

Where s*i* are the computed syndrome bits.

2.2.2 OLS Majority Voter

The OLS Majority Voter it Consist of Four Input Majority Gate and Xor Gate. The syndrome bits (s bits in the Block Diagram) are then used as inputs to the OLS majority voters that determine if the correction is needed for the OLS data bits (d bits in the Block Diagram). The syndrome bits that correspond to each of the sub-blocks are also compared with the SEC-DED syndrome patterns to determine if a correction is needed on the SEC-DED data bits (de bits in the Block Diagram). The Majority Gate Design is Consist of Four 3 Input AND Gate and the results are connected into two inputs OR Gate.

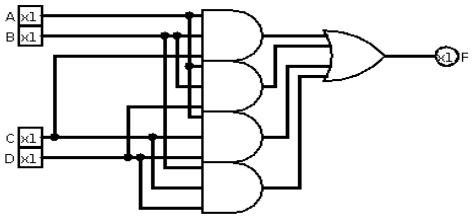

Fig 4 OLS majority voter

2.2.3 Orthogonal Latin Squares

The OLS codes were introduced decades ago to protect memories. On the end simultaneously have recently been proposed to protect caches and interconnects. The block sizes for OLS codes are k=m 2 data bits and 2tm parity bits Where t is the number of errors that the code can correct and mis an integer. For memories, the word sizes are typically a power of two and therefore mis commonly also power of two. The main advantage of OLS codes is that their decoding is simple and fast.

1) Each data bit participates in exactly 2t parity check bits;
2) Each other data bit participates in at most one of those parity check bits.

Therefore, for a number of errors or smaller, when one error affects a given bit, the remaining t −1 errors can, in the worst case affect t −1 check bits on which that bit participates. Therefore, still a majority oft +1 will trigger the correction on the erroneous bit. Conversely, when a given bit is correct, the errors on other bits will not cause miss correction as a majority of t +1 is needed

3. SOFTWARE REQUIREMENT

- Verification Tool
  - ModelSim 6.4c
- Synthesis Tool
  - Xilinx ISE 13.2

3.1 Modelsim

ModelSim SE - High Performance Simulation and Debug ModelSim SE is our UNIX, Linux, and Windows-based simulation and debugs environment, combining high performance with the most powerful and intuitive GUI in the industry.

3.2 Xilinxise

For two-and-a-half decades, Xilinx has been at the forefront of the programmable logic revolution, with the invention and continued migration of FPGA platform technology. During that time, the role of the FPGA has evolved from a vehicle for prototyping and glue-logic to a highly flexible alternative to ASICs and ASSPs for a host of applications and markets.

4. SIMULATION IMPLEMENTATION

Verilog HDL is a Hardware Description Language (HDL). A Hardware Description Language is a language used to describe a digital system, for example, a computer or a component of a computer. One may describe a digital system at several levels. For example, an HDL might describe the layout of the wires, resistors and transistors on an Integrated Circuit (IC) chip, i. e., and the switch level. Or, it might describe the logical gates and flip flops in a digital system, i. e., the gate level. An even higher level describes the registers and the transfers of vectors of information between registers. This is called the Register Transfer Level (RTL). Verilog supports all of these levels. However, this handout focuses on only the portions of Verilog which support the RTL level.

4.1 SEC DED Encoder

The below Fig 5 is the simulation snapshot of SEC-DED Encoder where de1 de2 de3 de4 are inputs and ce1 ce2 ce3 ce4 are the output obtained 4 input XOR Gate

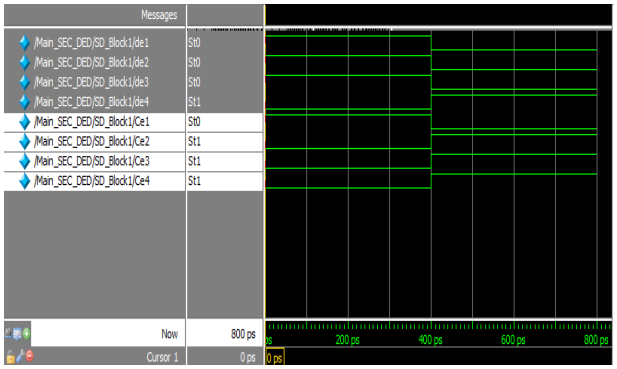

Fig 5 Output waveform of SEC DED Encoder

4.2 SEC-DED Main

The below Fig 6 is the simulation snapshot of SEC-DED Encoder where de1 to de16 are inputs and ce1 to ce16 are the output obtained sixteen 4 input XOR Gate design

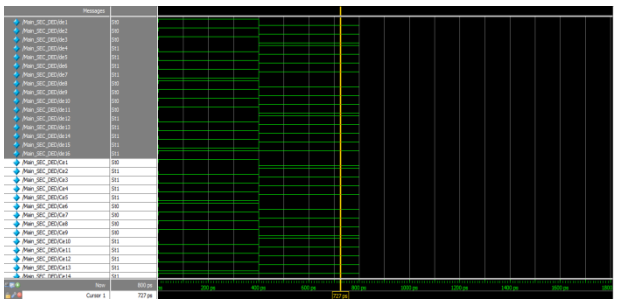

Fig 6 Output waveform of SEC-DED Main

4.3 Proposed DEC OLS Code Encoder

The below Fig7 is the simulation snapshot of DEC OLS Encoder where it provides the parity bits i.e. c1 to c16 provides encoded output

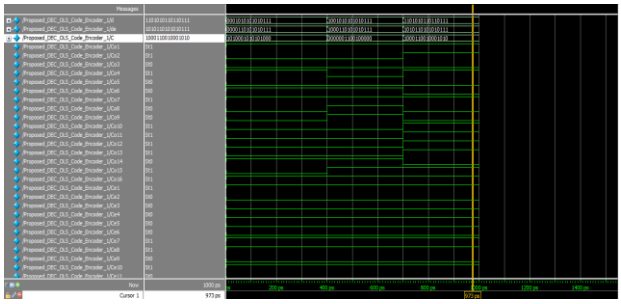

Fig 7 Output of DEC OLS Code Encoder

4.4 New proposed DEC OLS Code Encoder

The below fig 8 is the simulation snapshot of DEC OLS Code Decoder whenever Reset high no output when reset low output obtained at 40ps

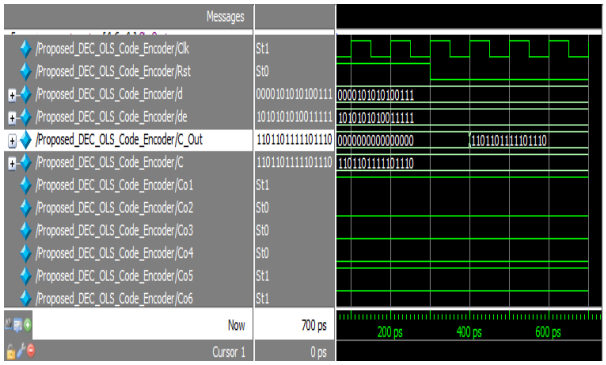

Fig 8 output of Proposed DEC OLS Code Encoder

4.5 New Proposed DEC OLS Code Decoder

The below fig 9 is the simulation snapshot of DEC OLS Code Decoder it is the output of syndrome computation unit here the error is checked and corrected

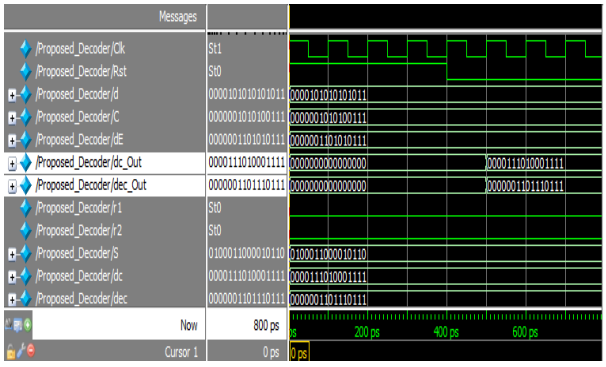

Fig 9 output of Proposed DEC OLS Code Decoder

4.6 Modified Decoder

The below fig 10 is the simulation snapshot of modified Decoder where it provides the output from OLS majority voters and SEC-DED block

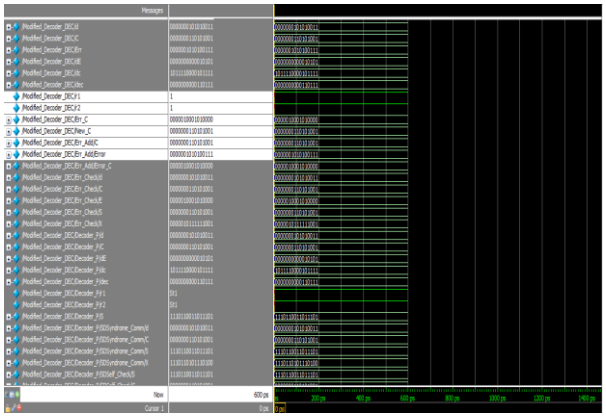

Fig 10 output of Decoder

5. DEVICE UTILIZATION SUMMARY

Table 1 Device utilization summary for encoder

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	32	3,840	1%	
Logic Distribution				
Number of occupied Slices	16	1,920	1%	
Number of Slices containing only related logic	16	16	100%	
Number of Slices containing unrelated logic	0	16	0%	
Total Number of 4 input LUTs	32	3,840	1%	
Number of bonded IOBs	48	97	49%	
Total equivalent gate count for design	192			
Additional JTAG gate count for IOBs	2,304			

Table 2 Device utilization summary for decoder

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	90	3,840	2%	
Logic Distribution				
Number of occupied Slices	48	1,920	2%	
Number of Slices containing only related logic	48	48	100%	
Number of Slices containing unrelated logic	0	48	0%	
Total Number of 4 input LUTs	90	3,840	2%	
Number of bonded IOBs	78	97	80%	
Total equivalent gate count for design	636			
Additional JTAG gate count for IOBs	3,744			

6. CONCLUSION

This paper consists two techniques to derive Unequal Error Protection (UEP) codes from Double Error Correction (DEC) Orthogonal Latin Squares (OLS) codes have been presented. The derived UEP codes can protect part of the word with DEC and the other part with SEC-DED. The codes can be decoded in parallel with low latency. Finally they do not require any additional parity bits compared to a standard OLS code. The implementation results for an FPGA platform have confirmed the low complexity and latency of both the encoder and the decoder. Future work will consider the derivation of UEP codes from OLS codes that can correct more than two errors. For example a TEC OLS code can be extended to also provide DEC for additional bits. It would also be interesting to derive a formal and general proof of the UEP capabilities of the proposed codes that does not rely on a case by case analysis and exhaustive error pattern testing.

7. REFERENCES

[1] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, "Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances", Springer Verlag, 2010.

[2] T.Heijmen, "Soft Errors from Space to Ground Historical Overview," Soft Errors in Modern Electronic Systems, Springer, 2011.

[3] M. Nicolaidis, "Design for soft error mitigation", IEEE Trans. on Device and Materials Reliability, vol. 5, no. 3, pp. 405–418, Sept. 2005.

[4] A. DeHon, N.Carter, H.Quinn, "Final Report for CCC Cross-Layer Reliability Visioning Study", Computing Community Consortium,

[5] G.C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re and A. Salsano, "Fault tolerant solid state mass memory for space applications", IEEE Trans. Aerosp. Electron. Syst., vol. 41, no. 4, pp. 1353-1372, Oct. 2005.

[6] W. K. Huang, Y.-N. Shen, and F. Lombardi, "New approaches for the repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Trans. Computer-Aided Des. Integr. Circuits Syst., vol. 9, no. 3, pp. 323–328, Mar. 1990.

[7] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: a state-of-the-art review", IBM J. Res. Develop., vol. 28, no. 2, pp. 124-134, Mar. 1984.

[8] E. Fujiwara, "Code Design for Dependable Systems: Theory and Practical Application", John Wiley & Sons, Inc., 2006.

[9] M.Y. Hsiao "A class of optimal minimum odd-weight column SEC-DED codes", IBM J. Res. Develop.,vol. 14, no. 4, pp. 395–301, Jul. 1970.