# On the Design of Different Concurrent EDC Schemes for S-Box and GF(*p*)

J. Mathew[1], H. Rahaman[2], A. Jabir[3], S. P. Mohanty[4], and D. K. Pradhan[5]
Computer Science and Engineering, University of North Texas, USA.[4]
Department of Computer Science, University of Bristol, UK.[1,2,3,5]
Email-ID: saraju.mohanty@unt.edu[4], pradhan@compsci.bristol.ac.uk[5].

# Summary and Conclusions

- An attacker can retrieve confidential information from cryptographic hardware by introducing internal faults.
- Error detection/correction (EDC), through fault tolerance, is an effective way to mitigate such fault attacks in cryptographic hardware.
- We analyze the area, delay, and power overhead for designing the S-Box, one of the main complex blocks in the AES, with error detection and correction capability.
- The S-Box, GF(p), and Parity Predictions (PPs) circuits are synthesized from the specifications and the decoding and correction circuits are combined.
- The analysis shows a comparison of the different approaches characterized by their error detection capability.

# On the Design of Different Concurrent EDC Schemes for S-box and GF(P)

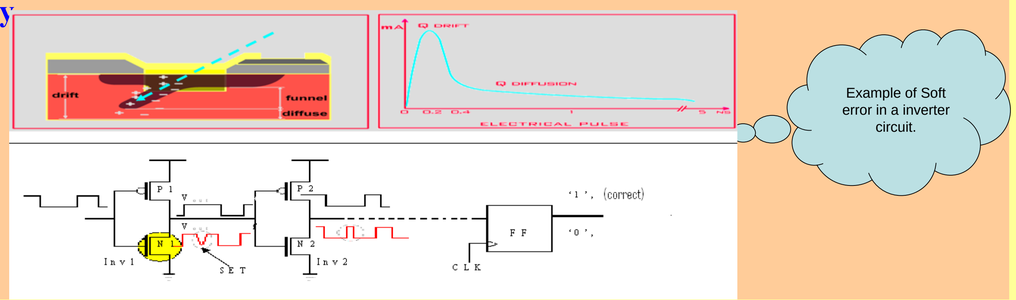
J. Mathew, H. Rahaman A.M. Jabir, Saraju P. Mohanty* D. K. Pradhan
University of Bristol, UK. *University of North Texas, USA.

ISQED Symposium 2010

March 22-24, 2010
DoubleTree Hotel
San Jose, CA USA

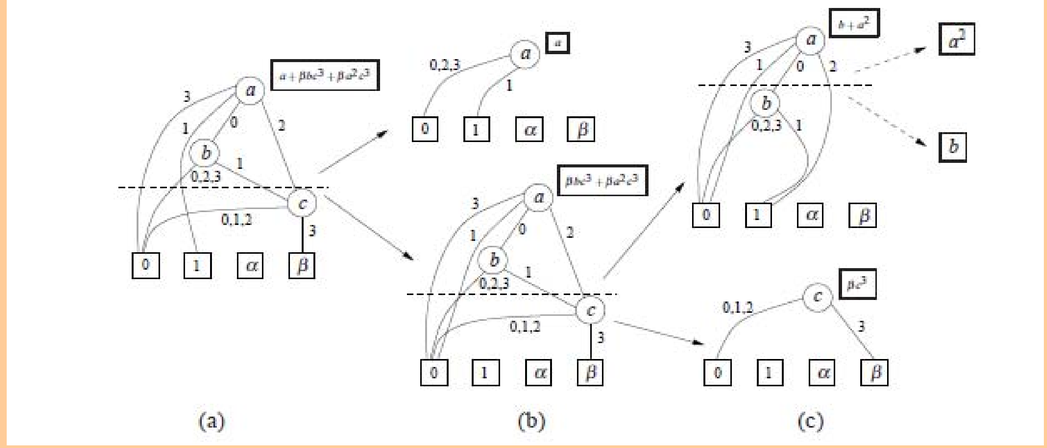
## Abstract


- Recent studies have shown that an attacker can retrieve confidential information from cryptographic hardware (e.g. the secret key) by introducing internal faults.
- A secure cryptographic implementation must detect/correct such a malicious attacks. Error detection /correction (EDC) is an effective way to mitigate such fault attacks in cryptographic hardware and further soft error problem in logic.
- To this end, we analayse the area, delay, and power overhead for designing of S-box which is one of the main complex blocks in the Advanced Encryption Standard(AES), with error detection and correction capability.
- We use multiple Parity Predictions (PPs), based on various error correcting codes, to detect and correct errors. Different coding techniques are presented, which include simple parity prediction, split parity codes, Hamming, Hsiao, and LDPC codes. The S-box, GF(P), and PP circuits are synthesized from the specifications and the decoding and correction circuits are combined to form the complete designs. The analysis shows a comparison of thedifferent approaches characterized by their error detection capability

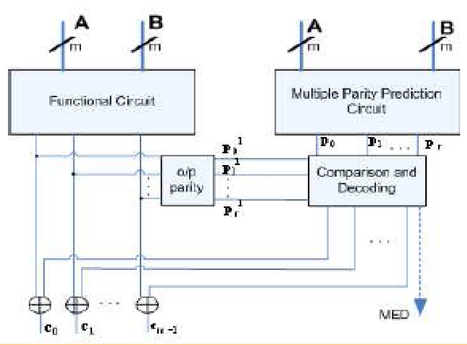

Example of Soft error in a inverter circuit.

## Synthesis and Optimization




- We consider the technique for synthesis and optimization of the multiple-output, multivariate polynomials over GF(2^m) based on[1].
- The circuits with and without the error correction schemes have been represented in terms of these polynomials, which we have synthesized with this technique. The polynomials are represented as the Shared Galois Polynomial Decision Diagrams (SGPDDs).

## Design




| Scheme | S-Box (area,delay,power) ($\mu m^2$, ns, mw) | PP block (area,delay,power) ($\mu m^2$, ns, mw) | % area Over-head |
|---|---|---|---|
| Simple parity | (5699.6, 2.75, 5.847) | (929.0, 1.13, 3.96) | 14.02% |
| Split Parity | (5699.6, 2.75, 5.847) | (1806.3, 1.64,3.47) | 24.06% |
| Hamming Code | (5699.6, 2.75, 5.847) | (3057.9, 2.21, 3.29) | 34.91% |
| LDPC | (5699.6, 2.75, 5.847) | (3838.5, 2.26, 4.01) | 40.24% |
| Hsiao | (5699.6, 2.75, 5.847) | (3941.7, 2.27, 4.2) | 40.88% |

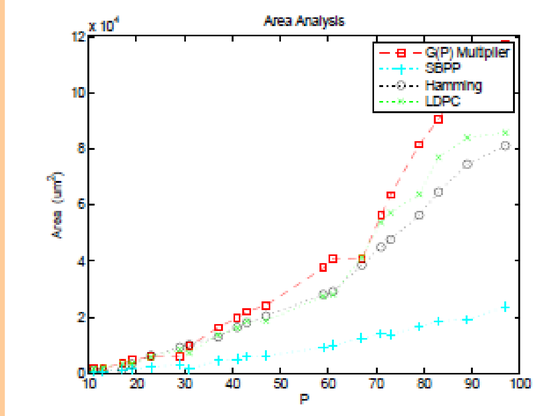
Broad Comparison of various Error Detection Schemes

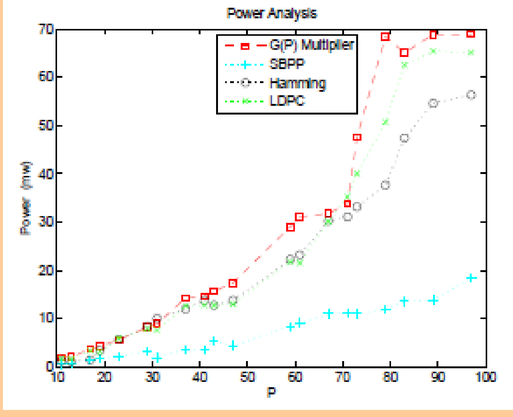
### Circuit with concurrent Error Detection and Correction

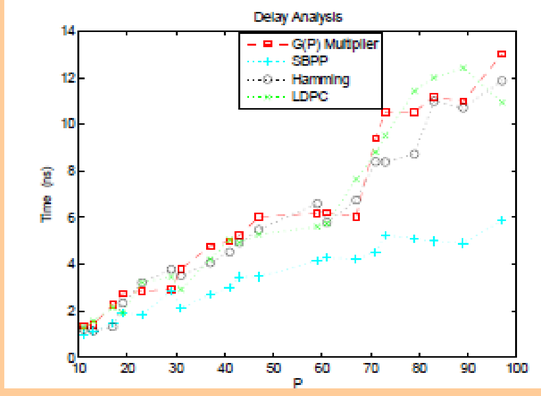
### Design Procedure

- Determine the number of parity bits (r) required.
- Construct the H matrix, with (m + r) non-zero r-bit column vectors. The dimension of the resulting matrix is r* (m+r).
- A column vector with a single 1 is assigned to parity Pi.
- The column vector with all 1s is assigned to output bit cm-1.
- The remaining m columns are assigned the output bits ci, without any constraints.
- Generate predicted parity expressions in terms of ci s. Next, generate the predicted output parities from the inputs.
- For Hsiao code, choose the parity check matrix such that the output bits are assigned to the columns with odd number of ones. In this case additional parity bits maybe required.
- Finally, combine the multiplier, PP, output encoder, decoder, andthe correction logic as shown in figure.

## Analysis



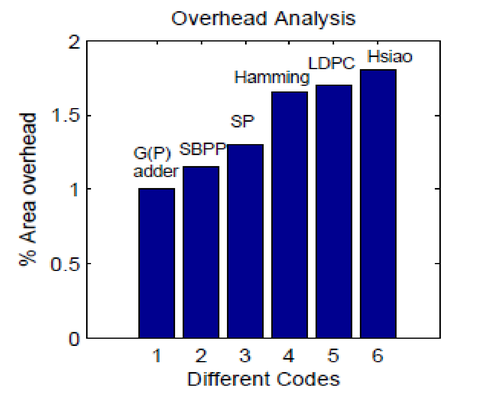




Figure shows area, power and delay Analysis

## Experimental Results

| Prim GF(P) | G(P) Adder (area,delay,power) ($\mu m^2$, ns, mw) | Single Bit Parity Prediction (area,delay,power) ($\mu m^2$, ns, mw) |
|---|---|---|
| GF(11) | (1677.30, 1.51, 1.96) | (638.6, 1.06, 0.68) |
| GF(13) | ( 2809.5, 1.81, 3.32) | (793.49, 1.14, 0.92) |
| GF(17) | (5977.03, 3.19, 6.05) | (1206.37, 1.54, 1.92) |
| GF(19) | (4461.01, 2.47, 4.66) | (1586.98, 1.77, 1.62) |
| GF(23) | (5254.51, 2.45, 5.94) | (1683.76, 1.96, 2.01) |
| GF(29) | (6180.25, 2.91, 8.19) | (2680.46, 2.41, 3.04) |
| GF(31) | (3406.24, 2.19, 4.49) | (2683.69, 3.00, 1.99) |
| GF(37) | (24711.371, 5.50, 25.52) | (4690.01, 2.79, 4.87) |
| GF(41) | (25217.80, 4.29, 15.16) | (4980.31, 3.04, 4.82) |
| GF(43) | (19989.08, 5.25, 26.93) | (5412.55, 3.09, 5.22) |
| GF(47) | (23756.59, 3.21, 10.34) | (5460.93, 2.88, 5.10) |
| GF(59) | (15279.71, 3.41, 19.55) | (9141.34, 3.77, 8.94) |
| GF(61) | (14224.94, 4.07, 18.57) | (9221.97, 3.88, 9.64) |
| GF(67) | (52622.36, 10.76, 64.88) | (11118.61, 4.77, 11.28) |
| GF(71) | (4572.07, 6.12, 35.61) | (14050.70, 4.93, 13.63) |
| GF(73) | (63457.37, 11.72, 79.20) | (14647.43, 4.74,13.96) |
| GF(79) | (21437.40, 3.96, 21.40) | (16850.52, 5.24, 14.58) |
| GF(83) | (51174.26, 6.82, 56.18) | (19021.34, 5.87, 18.0) |
| GF(89) | (59922.14, 11.90, 61.96) | (21930.81, 5.40, 19.7) |
| GF(97) | (57612.64, 6.99, 61.81) | (21605.03, 5.89,20.07) |

| Prim GF(P) | GF(P) Multiplier (area,delay,power) ($\mu m^2$, ns, mw) | Single Bit Parity Prediction (area,delay,power) ($\mu m^2$, ns, mw) |
|---|---|---|
| GF(11) | (1609.57, 1.36, 1.89) | (603.18, 0.99, 0.536) |
| GF(13) | (1919.22, 1.40, 2.04) | (616.09, 1.14, 0.665) |
| GF(17) | (3419.12, 2.27, 3.68) | (1254.75, 1.47, 1.41 ) |
| GF(19) | (4602.91, 2.73, 4.28) | (1545.06, 1.89, 1.70 ) |
| GF(23) | (5947.99, 2.84, 5.57) | (2177.27, 1.83, 2.13 ) |
| GF(29) | (6180.25, 2.91, 8.19) | (3219.14, 2.86, 3.19 ) |
| GF(31) | (9583.25, 3.80, 8.86) | (1490.22, 2.13, 1.90 ) |
| GF(37) | (16276.35, 4.77, 14.20) | (4602.92, 2.69, 3.51 ) |
| GF(41) | (19756.76, 4.98, 14.59) | (4951.28, 2.98, 3.60 ) |
| GF(43) | (21843.73, 5.22, 15.62) | (6167.34, 3.45, 5.35 ) |
| GF(47) | (24253.26, 6.03, 17.42) | (6225.40, 3.46, 4.40 ) |
| GF(59) | (37565.26, 6.18, 28.97) | (9302.63, 4.15, 8.30 ) |
| GF(61) | (40832.80, 6.22, 31.04) | (9941.29, 4.28, 9.12 ) |
| GF(67) | (40703.76, 6.05, 31.73) | (12476.60,4.21, 10.95) |
| GF(71) | (56231.86, 9.40, 33.82) | (14111.97,4.51, 11.04) |
| GF(73) | (63540.93, 10.54, 47.55) | (13741.02,5.24, 11.01) |
| GF(79) | (81549.51, 10.54, 68.44) | (16647.31,5.10, 11.85) |
| GF(83) | (90335.94, 11.18, 59.03) | (18553.61,4.98, 13.52) |
| GF(89) | (106212.2, 10.98, 68.85) | (19156.77,4.90, 13.85) |
| GF(97) | (117379.2, 13.02, 68.88) | (23679.10,5.88, 18.36) |

## Conclusion

The paper has aimed at comparing the performance of different error detection and correction techniques, which are used to mitigate malicious attacks. We presented an overhead analysis for designing S-Box and GF(P) arithmetic blocks. We used a heuristic gate as well as word-level synthesis and optimization technique for the analysis. Moreover, with regards to several performance index parameters, such as area, delay, and power a large set of experimental circuits has been designed. In conclusion, what clearly comes out from the experiments is, as evident, there is a linear increase in overhead as the number of error detection features increases. The performance figures also closely match those of the structural technique.

References
. A. Jabir and D. Pradhan. A Graph-Based Unified Technique for Computing and Representing Coefficients Over Finite Fields. IEEE Trans. Comp., 56(8):1119-1132, Aug. 2007.
. A. Reyhani-Masoleh, and M. Anwar Hasan, ”Fault Detection Architectures for Field Multiplication Using Polynomial Bases”, IEEE 91(11), ”,IEEE Trans. Computers, vol. 55, No. 9, Sept. 2006.
. C. Wang, V. Singal, and M. Ciesielski. BDD Decomposition for Efficient Logic Synthesis. In Int. Conf. Comput. Aided Design (ICCAD), pp. 626-631, 1999.
. S. Mitra and E. McCluskey, ”Which Concurrent Error Detection Scheme to Choose?,” in Test Conference,2000. Proceedings. International, pp. 985-994, 2000