- whether to compute square roots in $\mathbb{GF}(2^m)$ – as well as in $\mathbb{GF}(2^m)[z]/g(z)$ – by an algorithm [5] or by table look up
- whether to transform to and from normal bases in order to profit from the simple arithmetic using normal bases

is more beneficial in terms of space and/or in terms of time. According to expectation, each alternative is easily implemented in SAGE.

If the blocks are not chained then block-wise data parallelism can be exploited. Linear feedback shift registers are suitable for the matrix operations. A (very) deep pipeline for the decoding algorithm will increase latency but speed up the decryption accordingly, i.e. proportional to the number of stages, given enough hardware to prevent structure hazards. Of course, the SAGE implementation cannot answer this type of design questions which are inherent to the target FPGA which puts the SAGEs help to implement the McEliece PKCS into perspective.

References

[1] D. J. Bernstein, J. Buchmann, E. Dahmen (Eds.): Post-Quantum Cryptography; Springer 2009

[2] D. J. Bernstein, T. Lange, C. Peters: Attacking and defending the McEliece cryptosystem; PQCrypto 2008, LNCS 5299, pp. 31–46, 2008 http://eprint.iacr.org/2008/318.pdf

[3] R. T. Chien, B. D. Cunningham, I. B. Oldham: Hybrid methods for finding roots of a polynomial with application to BCH decoding; IEEE Trans. Inform. Theory, vol. IT15, pp. 329-335, 1969

[4] T. Eisenbarth, T. Güneysu, S. Heyse, C. Paar: MicroEliece: McEliece for Embedded Devices; CHES 2009: 49-64 www.crypto.rub.de/imperia/md/content/texte/publications/conferences/ches2009_microeliece.pdf

[5] K. Huber: Note on Decoding Binary Goppa Codes; Electronics Letters, 32:102-103, 1996

[6] Y.-X. Li, D.-X. Li, C.-K. Wu: How to Generate a Random Nonsingular Matrix in McEliece's PKCS; ICCS/ISITA '92, Singapore 1992, IEEE 268-269

[7] R. McEliece: Public Key Cryptosystem based on Algebraic Coding; DSN Progress Report 42-44, January/February 1978, 114–116 www.cs.colorado.edu/~jrblack/class/csci7000/f03/papers/mceliece.pdf

[8] N. J. Patterson: The Algebraic Decoding of Goppa Codes; IEEE Trans. on Information Theory, Vol IT-21, No 2, March 1975 203-207

[9] V. Rijmen: Effcient Implementation of the Rijndael S-box; was available www.esat.kuleuven.ac.be/~rijmen/rijndael/sbox.pdf, now e.g. www.comms.scitech.sussex.ac.uk/fft/crypto/rijndael-sbox.pdf or [10]

[10] Th. Risse: SAGE, ein open source CAS vor allem auch für die diskrete Mathematik; Wismarer Frege-Reihe, ISSN 1862-1767, Heft 03/2010, S.34-40 www.weblearn.hs-bremen.de/risse/papers/Frege2010_03

[11] Ron M. Roth: Introduction to Coding Theory; Cambridge University Press 2006 www.cs.technion.ac.il/~ronny

[12] NN: System for Algebraic and Geometric Experimentation, SAGE www.SAGEmath.org together with servers like www.SAGEnb.org or e.g. http://SAGE.informatik.hs-bremen.de

[13] W. A. Stein et al.: Sage Mathematics Software (Version 4.7.1); The Sage Development Team, 2011, www.sagemath.org

[14] A. Salomaa: Public Key Cryptography; Springer EATCS Monographs on Theoretical Computer Science Vol 23, 1990

[15] P. W. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer; Proc. 35^{th} Ann. Symposium on Foundations of Computer Science, Santa Fe, NM, Nov. 20–22, 1994 http://arxiv.org/pdf/quant-ph/9508027v2