ADAGE: An Automated Synthesis tool for Adaptive BCH-based ECC IP-Cores

ADAGE: An Automated Synthesis tool for Adaptive BCH-based ECC IP-Cores

Stefano DI CARLO, Michele FABIANO, Marco INDACO, Paolo PRINETTO
Politecnico di Torino
Dipartimento di Automatica e Informatica
Corso Duca degli Abruzzi 24, I-10129, Torino, Italy
Email: {name.familyname}@polito.it

Abstract

Bose-Chaudhuri-Hocquenghem (BCH) codes are a family of Error Correction Codes (ECCs) largely applied in modern Flash-based Hard Disks to significantly improve their endurance and reliability. ADAGE is an advanced ESL tool for the automatic generation of BCH-based ECC IP-Cores with adaptable correction capability. End-users can freely and dynamically change it on-the-fly. In addition, ADAGE supports architectural exploration of both decoders and encoders.

1. Introduction

NAND flash memories are a widespread technology for the development of low-power, low-cost and high data throughput mass storage systems. Manufacturers are pushing flash technologies to further reduce the cost per unit of storage. This includes moving from traditional single-level cell (SLC) technologies, able to store a single bit of information, to multi-level cell (MLC) technologies, thus storing more than one bit per cell. The strong transistor miniaturization and the adoption of an increasing number of levels per cell introduce serious issues related to reliability and endurance [1]

Error correction codes (ECCs) must therefore be systematically applied. The *Binary Bose and Ray-Chaudhuri* (BCH) codes are, in particular, a well-known correcting code used for NAND flash-memories [2]. BCH codes are a family of ECC constructed over the *Galois Fields* (GFs). Flash memories support ECCs by providing spare storage cells dedicated to system management and parity bit storage. Choosing the correction capability of an ECC is a trade-off between reliability and design complexity. It is therefore a strategic decision in the design of a flash-based storage system.

2. ADAGE

In this poster we present ADAGE, an advanced ESL tool for the automatic generation of BCH-based ECC IP-Cores with adaptable correction capability. ADAGE supports a systematic analysis and exploration of different architecture alternatives. This environment is strongly intended to be: *user-driven, automatic, parametric.* The availability of an adaptable correction capability proved to be very important. Since the reliability of a NAND flash memory continuously decreases over time, it's in fact highly desirable that the required correction capability could be dynamically adapted during the device lifecycle. Tuning the correction capability on-the-fly results both in saving power consumption and in a reduced decoding latency.

The overall explanation of the adaptable and optimized BCH-based ECC IP-Core architecture is detailed in [3].

Three main functional steps compose the design flow provided by ADAGE: 1) *Design Requirements*, 2) *Code Characterization* 3) *BCH IP-Core generation.*

1) *Design Requirements:* the user provides the length of the message to be encoded, the HW parallelism (e.g., 32-bit), and two values of BER. The former is the raw bit error rate (RBER), i.e., the BER before applying the error correction, that is technology dependent. The latter is the uncorrectable bit error rate (UBER), i.e., the BER after the application of the ECC, which is application dependent. The two BER values are needed to calculate the proper error correction capability. Several architectural options are provided to trade-off area and decoding latency.

2) *Code Characterization*: the second step choices the proper error correction capability and the actual GF. Given the GF, all the mathematical parameters (e.g., polynomials) are computed. The HW complexity of the BCH IP-Core is strictly dependent on these parameters.

3) *BCH IP-Core generation*: the VHDL description of the encoder and decoder modules is generated. A complete framework for validating the correctness of the BCH code's HW architecture is automatically generated, as well.

The whole framework combines Matlab software modules with custom C programs.

The full framework code can be freely downloaded from http://www.testgroup.polito.it in the Tools section of the website.

3. References

[1] D. Ielmini, "Reliability issues and modeling of flash and post-flash memory," *Microelectronic Engineering*, vol. 86, pp. 1870–1875, 2009

[2] R. Micheloni, A. Marelli, and R. Ravasio, *Error Correction Codes for Non-Volatile Memories*. Springer Publishing Company, 2008.

[3] S. Di Carlo, M. Fabiano, M. Indaco, P. Prinetto, "Generation of Adaptable BCH Codecs for NAND Flash", submitted to *Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 2012