

On-demand Syndrome Calculation for BCH decoding

Hyeonkyu Kim, Soyeon Choi, and Hoyoung Yoo

Department of Electronics Engineering
Chungnam National University
Deajeon, Republic of Korea

hkkim.cas@gmail.com; sychoi.cas@gmail.com; hyyoo@cnu.ac.kr

Abstract—This paper presents a new syndrome calculation method to reduce hardware complexity for BCH decoding. Compared to previous works that calculate all $2t$ syndromes simultaneously in the syndrome calculation (SC) stage, the proposed architecture schedules the syndrome calculation to reduce the hardware complexity; odd-indexed syndromes are computed in the SC stage, while even-indexed syndromes are computed when they are needed in the key-equation solving (KES) stage. Experimental results show that the proposed architecture saves 65% of hardware resources compared to the conventional architecture.

Keywords—Binary BCH code, syndrome calculation, schedule optimization.

I. INTRODUCTION

Among various error-correction codes used to recover corrupted codewords in communications and storage systems, the Bose-Chaudhuri-Hocquenghem (BCH) code is one of the most widely used algebraic codes due to its superior error-correction performance. The binary BCH code has been employed in diverse systems such as advanced solid-state storages [1], and optical fiber communication systems [2], as it can correct multiple erroneous bits. Most of those applications are continuously demanding ever higher decoding throughput as well as ever larger error-correction capability, and these requirements necessitate architectural innovations such as parallel processing and pipelining [3][4]. Though advanced techniques contribute to low latency and high performance, the hardware overhead associated with such advanced techniques is serious in practical implementations. Therefore, an efficient architecture is inevitable to reduce the hardware complexity of a strong BCH decoder.

A BCH decoder that can correct t bits at maximum is usually composed of three pipelined blocks: syndrome calculation (SC), key-equation solving (KES), and Chien search (CS) [2]. Given a received codeword, the SC block computes $2t$ syndromes, and the KES block generates the error locator polynomial $\Lambda(x)$ by using the syndromes. Finally, message bits are recovered by taking into account errors found in the CS block. Many optimization methods such as folding [4] and common substructure sharing [5] have been presented to relax the complexity of the KES and CS. Though SC has been regarded as being associated with a relatively less complexity, those optimizations make it have almost the same complexity as the other blocks [3]. Despite the fact that the

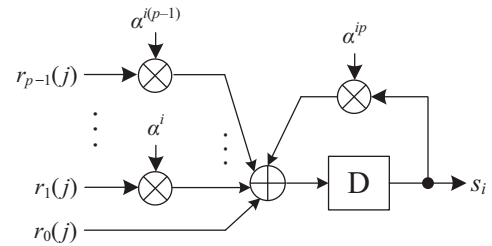


Fig. 1. A traditional p -parallel unit for syndrome calculation.

massive-parallel SC takes almost a third of the whole decoder area, a few works have been reported to reduce the complexity of a parallel SC [6][7].

In this paper, we present an area-efficient architecture to compute syndromes. In contrast to the previous architectures that generate all syndromes simultaneously, the proposed architecture computes only odd-indexed syndromes in the SC block. The rest even-indexed syndromes are generated in the KES block only when they are demanded to solve the key equation. As only one even-indexed syndrome is additionally required in a processing step of the KES block, the even-indexed syndromes can be serially generated with a small hardware unit. As a result, the proposed method can reduce the complexity of the SC block significantly.

II. PREVIOUS BCH DECODERS

Let us consider a binary BCH (n, k, t) code, where n , k , and t denote the code length, the message length, and the error-correction capability, respectively. In the BCH decoding, the received n -bit codeword $(r_0, r_1, \dots, r_{n-1})$ is interpreted as a polynomial, $R(x) = r_0 + r_1x^1 + \dots + r_{n-1}x^{n-1}$. The SC block checks if there are any errors in the received codeword by carrying out the following syndrome computations:

$$S_i = R(\alpha^i) = \sum_{j=0}^{n-1} r_j \alpha^{ij} = r_0 + r_1 \alpha^i + \dots + r_{n-1} \alpha^{i(n-1)} \quad (1)$$

A typical structure to compute a syndrome is shown in Fig. 1, where p represents the parallel factor denoting the number of received bits being considered in a cycle. To compute $2t$ syndrome in parallel, the p -parallel SC block consists of $2t$ such units.