ELectronics EXpress

IEICE Electronics Express, Vol.8, No.13, 1001–1007

Area-efficient reed-solomon decoder for 10 Gbps satellite communication

Botao Zhang[1a)], **Hengzhu Liu**[1], **and Xianqiang Yang**[2]

[1]*Computer School, National University of Defense Technology, Changsha, China*

[2]*China Academy of Space Technology (CAST), Beijing, China*

a) *botaozhang@nudt.edu.cn*

Abstract: This paper propose an area-efficient pipeline-balancing Reed-Solomon decoder for 10 Gbps satellite communication. The proposed RS (244,212) is based on TD-iBM Key Equation Solver architecture, and Fixed-Factor Syndrome Computation & Chien Search. The decoder is implemented and verified in FPGA, and can work at 178 MHz in Virtex2P. Thus a 8-channel FPGA implementation can be used for 10 Gbps satellite communication systems. Additionally, the decoder is also synthesized in Chartered 90 nm CMOS technology, and compared with previous decoders. The results show the decoder is more area-efficient than previous decoders. Meanwhile, by using this CMOS technology, the decoder can be clocked at about 1350 MHz, so a single-channel ASIC implementation can meet the requirement of 10 Gbps satellite communication.

Keywords: Reed-Solomon decoder, area-efficient, pipeline-balancing

Classification: Integrated circuits

References

[1] S. Lee, C. S. Choi, and H. Lee, "Two-parallel Reed-Solomon based FEC architecture for optical communications," *IEICE Trans. Electron. Express,* vol. 5, no. 10, pp. 374–380, 2008.

[2] B. Yuan, Z. F. Wang, L. Li, M. L. Gao, J. Sha, and C. Zhang, "Area-Efficient Reed-Solomon Decoder Design for Optical Communications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 6, pp. 469–473, June 2009.

[3] D. V. Sarwate and N. R. Shanbhag, "High-speed architectures for Reed-Solomon decoders," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 5, pp. 641–655, 2001.

[4] S. D. Shieh, Y. K. Lu, S. M. Chung, and J. H. Chen, "Design and implementation of efficient Reed-Solomon decoders for multi-modeapplications," *Inter. Sym. Circuits Syst. (ISCAS2006)*, Island of Kos, Greece, pp. 289–292, May 2006.

[5] J. Park, K. Lee, C. S. Choi, and H. Lee, "High-Speed Low-Complexity Reed-Solomon Decoder using Pipelined Berlekamp-Massey Algorithm," *7th Inter. SoC Design Conf.*, Busan, Korea, pp. 452–455, Nov. 2009.

[6] J. I. Park and H. Lee, "Area-efficient truncated Berlekamp-Massey architecture for Reed-Solomon decoder," *Electron. Lett.*, vol. 47, no. 8, pp. 241–243, 2011.

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011

1 Introduction

Reed-Solomon (RS) codes have been widely used in many kinds of modern communication systems, such as satellite communication system (DVB-S), star-earth link system (ShiJian-V), as well as optical communication system (10GBase-LR) [1, 2]. RS (244,212) is recommended by CAST for the wideband multimedia satellite communication system. Due to the increasing requirements for multi-channel high definition multimedia applications in wide-band satellite communication systems, high speed and low hardware cost RS decoder is needed urgently.

General RS decoders include Syndrome Computation (SC), Key Equation Solver (KES), and Chien Search & Error Correction (CSEC) blocks. KES block, which is considered as the most complex block, occupies more than 60%-80% area of the whole decoder [1]. Many implementations of KES have been proposed to downsize the VLSI area. They can be divided into two categories, namely, the Berlekamp-Massey (BM) based implementation and the modified Euclidean (ME) based implementation. In this paper, a new scheme named TD-iBM is proposed based on iBM, which adopts time division scheduling to increase the utilization of Galois Field (GF) multipliers, and reduce the complexity. We present an area-efficient pipeline-balancing RS decoder for 10 Gpbs satellite communication applications based on TD-iBM, and implement it on Virtex2P FPGA and Chartered 90 nm CMOS technology. Meanwhile, other main ideas for achieving low complexity in SC and CSEC designs are introduced in the proposed decoder.

2 Proposed architecture and timing

2.1 Fixed-Factor Syndrome Computation block

SC block calculates $2t$ syndromes from the received stream, where t denotes the error correction capacity [1]. It can be implemented in full parallel or partly parallel. When it is implemented in P-parallel, which means P sets of GF adders and multipliers are designed, the processing time of one code T_{cycle} will be $2t \times n/P$, where n is the code length. Since decoding speed is one of the main optimization goals in our design, full parallel (32-parallel) implementation is adopted. The above part of Fig. 1 (a) shows the implementation detail of the full-paralleled SC block. The parameter t is 16, and n is 244, thus T_{cycle} is 244.

Since the origin element α is fixed in our design, and one operand of each multiplier in SC is constant, multipliers in SC block can be simplified into single operand Fixed-Factor (FF) multipliers. Fig. 1 (a) shows the detail of one type of FF multipliers which is used for calculating syndrome s_0. FF multipliers are designed by cutting the redundant logic of the general multipliers when one operand is fixed. The experiment shows the hardware cost of the FF multiplier of which detail is shown in Fig. 1 (a) is 3 slices, while the hardware cost of general multiplier is 30 slices. Since the most complex FF multiplier in the design occupies 16.7% hardware of the general multiplier, the FF scheme can significantly reduce the hardware cost.

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011

2.2 TD-iBM key equation solver block

The iBM algorithm is used to solve the key equation $\omega(x) \equiv \sigma(x)S(x) \, mod \, x^{2t}$, where $\sigma(x)$ and $\omega(x)$ are defined as the error locator polynomial and the error evaluator polynomial. In general iBM algorithm, $t+1$ multipliers are used to design KES, and the hardware overhead is too high. Moreover, the processing latency of KES block is far less than SC block, so the macro-pipeline

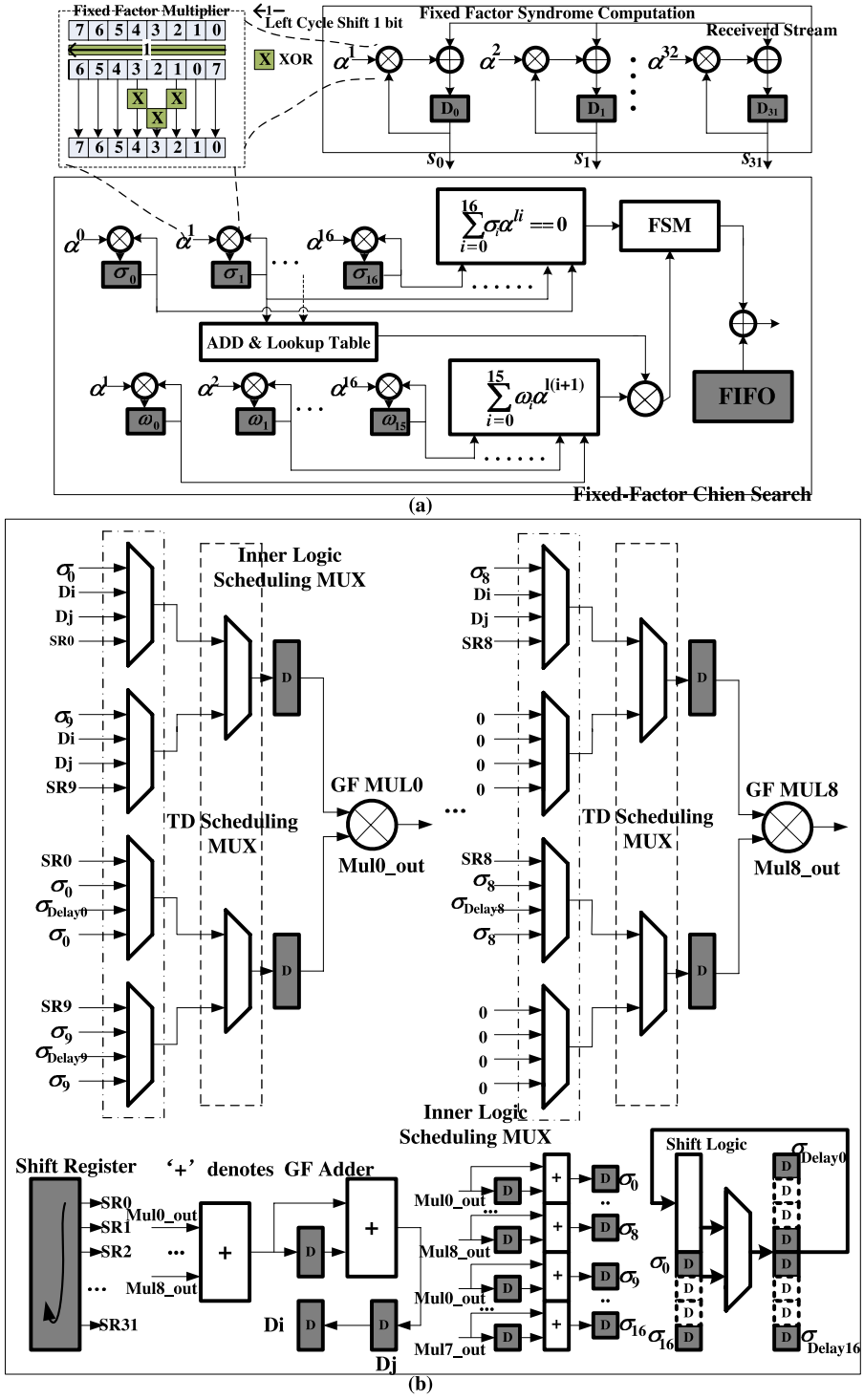

Fig. 1. Three main blocks of the proposed decoder (a) SC & CSEC Blocks (b) KES Block

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011

between KES and SC is not balanced. TD-iBM algorithm is a variation of iBM algorithm to reduce hardware cost and support micro-pipeline balance. Time Division (TD) scheduling is adopted to complete the KES block with $\lceil (t+1)/\Gamma \rceil$ multipliers, where Γ is the factor of the TD scheduling. When Γ increases, the hardware cost of KES block will be reduced, while the processing latency will be increased.

Algorithm 1 : TD-iBM Algorithm

Input: Syndrome set S: $\{s_i | i = 0, 1, .., 2*t-1\}$

Output: Sigma set Σ: $\{\sigma_i | i = 0, 1, .., t\}$;Omega set Ω: $\{\omega_i | i = 0, 1, .., t-1\}$

Initial: $D_i = 1, \sigma_i = 1$, others are all set to zero;

1: **for** j := 0 to 2*t-1 **do**
2: Realign S to generate S^*, **foreach** $s^*_n \in S^*$,$s^*_n = s_{j-n}$;
3: Divide set S^* and Σ^j into Γ subsets $S^*_1, .., S^*_\Gamma, \Sigma^j_1, .., \Sigma^j_\Gamma$,
foreach $s^*_{\gamma,n} \in S^*_\gamma$, $s^*_{\gamma,n} = s^*_{N*(\gamma-i)+n}$;
foreach $\sigma^j_{\gamma,n} \in \Sigma^j_\gamma$, $\sigma^j_{\gamma,n} = \sigma^j_{N*(\gamma-i)+n}$;
4: **foreach** subset S^*_γ and Σ^j_γ
$dpart_{\gamma,j} \Leftarrow \sum_{n=0}^{N-1} s^*_{\gamma,n} \otimes \sigma^j_{\gamma,n}$,where $\sum$ is the GF sum;
5: $D_j = \sum_{\gamma=1}^{\Gamma} dpart_{\gamma,j}$,where $\sum$ is the GF sum;
6: **foreach** $D_m, m \in \{-1, 0, .., j-1\}$ and $D_m \neq 0$
find the maximum of $(m - D_m)$, $i \Leftarrow m$
7: Divide Σ^i into Γ subsets,**foreach** $\sigma^i_{\gamma,n} \in \Sigma^i_\gamma$, $\sigma^i_{\gamma,n} = \sigma^i_{N*(\gamma-i)+n}$;
8: **foreach** subset Σ^i_γ and Σ^j_γ
9: **foreach** $\sigma^j_{\gamma,n} \in \Sigma^j_\gamma$, $\sigma^i_{\gamma,n} \in \Sigma^i_\gamma$
10: $delta'_{\gamma,n} \Leftarrow D_i \otimes \sigma^j_{\gamma,n} \ominus D_j \otimes \sigma^i_{\gamma,n}$;
11: $\sigma'^{(j)}_{N*\gamma+n} = delta'_{\gamma,n}$;
12: **for** m := 0 to t **do**
13: $\sigma^{(j+1)}_m \Leftarrow (D_j = 0)\ ?\ \sigma^{(j)}_m : \sigma'^{(j)}_m$;
14: **end for**
15: **end for**
16: **for** i:= 0 to t-1 **do**
17: Realign S to generate S^{**}, **foreach** $s^{**}_n \in S^{**}$,$s^{**}_n = s_{i-n}$;
18: Divide set S^{**} and Σ into Γ subsets $S^{**}_1, .., S^{**}_\Gamma, \Sigma_1, .., \Sigma_\Gamma$like (3),
19: **foreach** subset S^{**}_γ and Σ_γ
$omegapart_\gamma \Leftarrow \sum_{n=0}^{N-1} s^{**}_{\gamma,n} \otimes \sigma_{\gamma,n}$,where $\sum$ is the GF sum;
20: $\omega_i = \sum_{\gamma=1}^{\Gamma} omegapart_\gamma$,where $\sum$ is the GF sum;
21: **end for**

The TD-iBM algorithm is showed above. The symbol '$\otimes$', '$\oplus$', and '$\ominus$' denote GF multiplier, adder, and subtracter. In TD-iBM algorithm, $\sigma(x)$ and $\omega(x)$ are calculated in sequence: the first 15 lines are used for $\sigma(x)$ calculating and the rest ones are for $\omega(x)$ based on the $\sigma(x)$ results. TD-iBM employs TD scheduling in three operation sections: the first one is line 2-5, the second one is line 7-11, and the last one is line 17-20. Taking the first one as the

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011

example, the two operand sets: Σ and S^* are divided into Γ subsets, each subset has N elements ($N = \lceil t + 1/\Gamma \rceil$, the element number of last subset may be less than N, zero will be used for filling in), all the subsets will complete the operation (4) in sequence. When TD scheduling is not employed, $t+1$ multipliers are needed to generate d_i in full parallel. When Γ-factor TD scheduling is used, d_i will be generated in Γ cycles, so the multiplier number will be reduced to $\lceil (t+1)/\Gamma \rceil$. According to Algorithm 1, the processing latency will be $2 \times t \times 3 \times \Gamma + t \times \lceil t/\lceil (t+1)/\Gamma \rceil \rceil$. When TD-iBM algorithm is used for different applications, the factor Γ should be selected carefully to balance the processing latency and the hardware cost.

Fig. 1 (b) shows the main datapath of the TD-iBM architecture. The TD factor Γ is optimized as 2 in the proposed decoder. The main components of TD-iBM architecture are 9 multipliers and related multiplexers. The variables in TD-iBM first pass the Inner Logic Scheduling MUXs, which are implemented as MUX4, and then pass the TD Scheduling MUXs, and are sent into the register for GF multiplier. The GF multiplier results will be used to generate the new variables for the next cycle. The generation circuit of D_i, D_j, σ set, and σ_{delay} set are also showed in Fig. 1 (b). As the datapath of ω set generation circuit is similar to σ set, it is omitted in Fig. 1 (b). According to line 2 and line 17 in TD-iBM algorithm, one shift register is designed to prepare the syndrome set for d_i calculation. The '*LoadEnable*' ports of all the D-latches and Shift Register are controlled by the decoding FSM according to the processing timing.

2.3 Fixed-Factor chien search block

The architecture detail of CSEC block is showed in Fig. 1 (a). In the block, Chien Search algorithm and Forney algorithm were used to calculate the error location and value. Similar to SC block, all the multipliers in the block were implemented as FF multipliers. The above part of CSEC block, containing 17 FF multipliers, is the Chien Search part, and the below one, containing 16 FF multipliers, is used for error correction. According to the algorithm, 212 cycles are needed to complete the operations in CSEC block.

2.4 Balanced 3-stage macro-pipeline architecture

Fig. 2 shows total architecture and FPGA implementation layout of the single-channel decoder for RS (244,212). SC, KES, and CSEC block work in 3-stage macro-pipeline scheme. When 'FS' signal is invoked, SC block starts to accept and process in serial mode. After 244 cycles, all the 32 syndromes for one code frame are calculated, then SC block invokes KES block. KES block accepts the input syndromes in parallel, calculates results in 224 cycles, and then invokes CSEC block after waiting for 20 cycles. CSEC block accepts σ and ω sequences and outputs the decoding results in serial mode. According to the decoding timing, the pipeline utilization of SC, KES, and CSEC block are 100%, 91.8%, and 86.9%.

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011

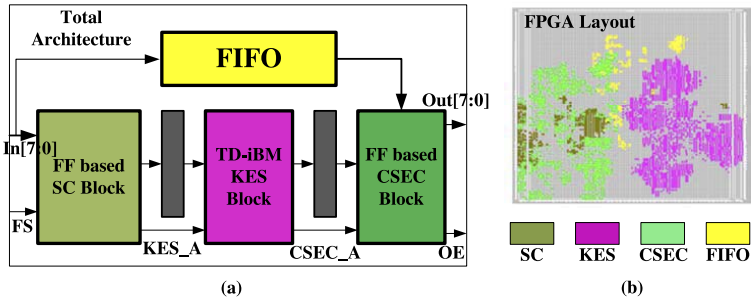

Fig. 2. (a)Total Architecture (b)Virtex2P FPGA Layout

3 Results and comparison

The proposed decoder was modeled in Verilog HDL, implemented and verified in FPGA Virtex2P with the Xilinx ISE tool chain. The hardware cost is 1832 Slices, including 3262 LUTs, 2236 FFs, and one BRAM. The decoder can operate under 178 MHz in Virtex2P, and can offer 1.4 Gbps throughput. Hence, an 8-channel implementation can be used in 10 Gbps satellite communication.

Table I. Area-efficient comparison of the related decoders

Designs	C.T.	Tech.	Throughput	Gates	Efficiency
rDCME [2]	t=8	180 nm	8 bpc	18400	3.56
RiBM [3]	t=8	90 nm	8 bpc	24320	2.69
M-M RiBM [4]	t=8	180 nm	8 bpc	22931	2.85
pRiBM [5]	t=8	90 nm	8 bpc	43600	1.50
TiBM [6]	t=8	90 nm	8 bpc	19730	3.32
Proposed	t=16	90 nm	8 bpc	28591	4.58

To compare the proposed decoder with the previous decoders, the HDL model was synthesized in Chartered 90 nm CMOS technology with Synopsys Design Compiler. The area report of Design Compiler is 80696 um^2 without the FIFO block, which is identical to 28591 NAND2 gates. The timing report shows the maximum clock frequency is 1350 MHz. Table I shows the area-efficient comparison results of the related decoders, C.T. is shorted for Code Type. All the values in *Gates* column are not including the FIFO block. The metric of *Throughput* column is *bit per cycle*, which is much fairer than *Gbps*, as the gain from the CMOS technology is eliminated. The gate is used for comparison to eliminate the variation of CMOS technology in different decoders. Moreover, the code type proposed in this paper is not popular, so RS (255,239) decoders were used for comparison. The most difference between RS (244,212) and RS (255,239) is the parameter t, which will affect the complexity of the decoder and the error correction capacity.

To compare these decoders fairly, index $1024 \times Throughput \times C.T./Gates$ is used to denote the hardware efficiency. The throughput, VLSI cost, and error correction capacity are all considered in this index, while the influence of

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011

the CMOS technology is eliminated. The efficiency of the proposed decoder is 4.58, which outperforms any other decoders. That is mainly because of the FF multiplier implementation, and the proposed TD-iBM architecture which improves the efficiency of the GF multiplier.

4 Conclusions

This paper proposed an area-efficient Reed-Solomon decoder for 10 Gbps Satellite Communication. TD-iBM algorithm and architecture were introduced to downsize the VLSI area and balance the pipeline. Meanwhile, Fixed-Factor multipliers were also used to reduce the hardware cost. The decoder architecture was implemented in both Xilinx FPGA and Chartered 90 nm CMOS technology. The results showed the proposed decoder is more area-efficient than previous decoders. And an 8-channel FPGA implementation or a single-channel ASIC implementation can be feasible in the 10 Gbps satellite communication system.

Acknowledgments

This work was partly supported by National Natural Science Foundation of China (no. 60970037) and China Academy of Space Technology (no. 20080302).

DOI: 10.1587/elex.8.1001
Received April 18, 2011
Accepted June 08, 2011
Published July 10, 2011