Golay and Wavelet Error Control Codes in VLSI

Arunkumar Balasundaram*, Angelo Pereira, Jun Cheol Park*and Vincent J. Mooney III*
*Center for Research on Embedded Systems and Technology
School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, GA 30332-0250 U.S.A., Tel: 1-404-385-1722, Fax: 1-404-385-1746
arunkumar.balasundaram@gm.com, {angelop, jcpark, mooney}@ece.gatech.edu

Abstract – This paper presents a high speed VLSI implementation of wavelet and golay error control codes. The design has been fabricated by MOSIS in a TSMC 0.25 µm CMOS process. Experimental results show a maximum speed of 145 MHz and a data transfer rate of 870 Mb/sec.

I. Introduction

This work shows experimental results for two error control code implementation – a (12, 6, 4) wavelet encoder/decoder and a (24, 12, 8) golay encoder/decoder, where the (*N*, *M*, *d*) nomenclature stands for (*N*=code length, *M*=message length, *d*=distance) [1]. These codes have a correctable limit of one bit error and three bit errors, respectively. We transformed a wavelet filter bank, which uses the theory of circulant matrices, to implement encoding/decoding functions largely in combinational *XOR* logic. The golay decoder is the only exception and is implemented using a sequential logic block with a latency of 12 clock cycles. To the authors' best knowledge, this is the first silicon implementation of the wavelet encoder/decoder and the wavelet-based golay encoder/decoder.

II. Architecture

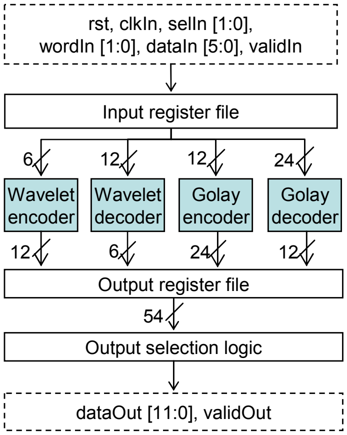

Fig. 1. Functional block diagram

The register transfer level (RTL) description of the circuit has been done using Verilog; the corresponding data flow is shown in Fig. 1. The design has been optimized for the wavelet encoder with 6 data input pins and 12 data output pins. Therefore, 12-bit and 24-bit data inputs are inserted serially taking 2 and 4 cycles, respectively. The *wordlIn* signal of Fig. 1 indicates which segment of the multi-word input sequence is being provided in the current cycle. Similarly, a 12-bit output is returned in two cycles. The *selIn* signal controls the *output selection logic* of Fig. 1 and decides which data from encoder/decoder modules to release to the output pins *dataOut*. The *validIn* and *validOut* signals of Fig. 1 indicate the validity of *dataIn* and *dataOut*, respectively, in a particular clock cycle.

The wavelet encoder/decoder and the golay encoder are single stage combinational blocks and thus take one cycle to compute the output. The golay decoder is a sequential logic block with a latency of 12 cycles. Since the golay code is an extension of the wavelet code and is the most complex functional block in the chip, we describe the architectural implementation of the golay decoder in this paper. The interested reader may refer to [5] for architectural details of the other blocks.

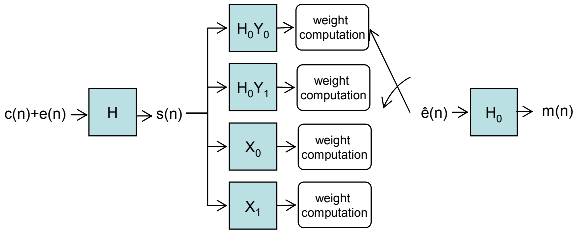

Fig. 2. Syndrome and message generation of golay

The golay decoder consists of the syndrome generator and the message reconstructor as depicted in Fig. 2, in which *s(n)* and *m(n)* represent the syndrome and message of *n* sequences, respectively; and shaded boxes represent filter banks [2]. The syndrome, which is uniquely associated with a set of variant codes within the hamming distance *d*, is defined as *s(n) = H(c(n)+e(n))* where *c(n)*, *e(n)* and *H* are the codeword, communication channel error of *n* sequences and syndrome filter bank, respectively. The generated syndrome can be mapped onto more than one error pattern. Therefore, the low *M*-dimension *s(n)* is interpolated into a higher *N*-dimension error pattern *ê(n)*, which is accomplished by choosing the maximum likelihood message according the weight computation in the reconstructor.

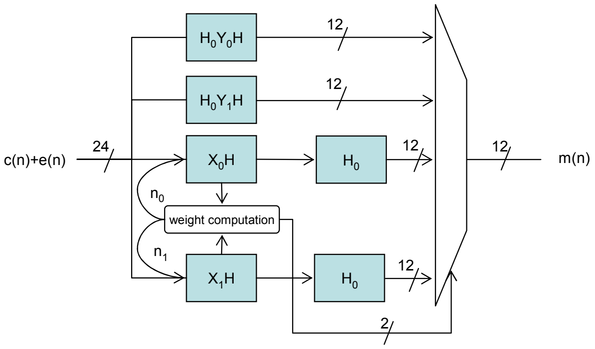

Fig. 3. Wavelet-based golay decoder block diagram

Fig. 3 shows the logic implementation of the golay algorithm of Fig. 2. The filter banks in shaded boxes are implemented with one stage *XOR* operations. We integrated the syndrome filter bank into the reconstructor filter bank for logic optimization. The computed weight is used to choose an appropriate filter bank for the message generation. The filter bank X_0H and X_1H require an M time cyclic shift of the integer value n_0 and n_1 to find the desired weight value for the error pattern. Therefore, since M=12, a maximum of 12 cycles are taken to complete all possible weight computations.

III. Design tool flow

The design was simulated for functionality and speed using Mentor Graphics' MODELSIM. The circuit was synthesized to an Artisan library for TSMC 0.25μm CMOS process using Synopsys Design Compiler. Cadence Silicon Ensemble was used for floorplanning and automatic place and route of the circuit. Antenna violations encountered during this step were fixed by inserting protection diodes and by the manual layer hopping method [3, 4]. The chip has been fabricated by MOSIS using dual in line (DIP) package, quad flat package (QFP) as well as bare die option to enable complete characterization. The chip layout taken from the DIP die is shown in Fig. 4. Further details with regard to design flow can be found in [5].

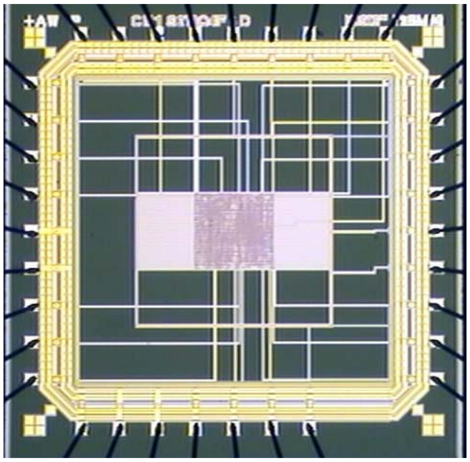

Fig. 4. Chip layout

IV. Experimental setup

The fabricated chip has been tested using an HP 83000 Digital IC Test system (courtesy of Dr. David Keezer, Professor, School of Electrical and Computer Engineering, Georgia Institute of Technology).

V. Experimental results

The complete encoder/decoder logic has been successfully tested for its functionality at a clock period of 6.9 ns which yields a speed of 145 MHz. The setup time for the inputs is about 2 ns and the delay on the outputs is about 5 ns. This correlates closely to the speed achieved in simulation taking into account the delay introduced by the package parasitics. The maximum speed obtained in simulations is about 5.2 ns or 192 MHz. The maximum time of flight delay caused due to the parasitics of the leads of the DIP package, as given by MOSIS, is about 0.21 ns. Taking into account the extra delay of 0.21 ns introduced by the package, the predicted maximum speed achievable by the circuit is 185 MHz. The difference in experimental results may be attributed to the uncertainty of the manual calibration of the tester.

VI. Summary and Conclusions

In this paper, we have presented an implementation of wavelet and golay error control codes, which has been optimized for the wavelet encoder. The design has been fabricated by MOSIS in TSMC 0.25 μm process. The fabricated chip has been successfully tested for its functionality at a high speed of approximately 145 MHz. For a 6-bit input at this clock frequency, the effective data throughput is 145x6=870Mb/sec.

Acknowledgements

Funding for this project was provided by a Georgia Tech College of Engineering Cutting Edge Research Grant as well as the Georgia Tech Packaging Research Center. We also acknowledge donations received from Artisan (through Dr. John Uyemura), Intel, Mentor Graphics, National Semiconductor, Sun and Synopsys.

References

[1] F. Fekri, S. W. Mclaughlin, R. M. Mersereau and R. W. Schafer, "Double circulant self-dual codes using finite field wavelet transforms," Springer Verlag Lecture Notes in Computer Science (LNCS); Applied Algebra, Algebraic algorithms and Error-Correcting Codes, pp355-364, 1999.

[2] F. Fekri, S. W. Mclaughlin, R. M. Mersereau and R. W. Schafer, "Decoding of half-rate wavelet codes; golay code and more," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), Vol. 4, pp. 2609-2612, 2001.

[3] W. Maly, H. T. Heineken, J. Khare, P. K. Nag, P. Simon and C. Ouyang, "Design manufacturing interface: Part II - Applications," Proceedings of Design for Manufacturability – Embedded Tutorial, (DATE 98), pp. 557-562, 1998.

[4] Source Link, Cadence Inc., http://sourcelink.cadence.com.

[5] A. Balasundaram, A. Pereira, J. C. Park and V. Mooney, "Golay and wavelet error control codes in VLSI," Technical Report GIT-CC-03-33, Georgia Institute of Technology, November 2003,
http://www.cc.gatech.edu/tech_reports/index.03.html.