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
[+]College of Computing
Georgia Institute of Technology
arunkumar.balasundaram@gm.com, {angelop, jcpark, mooney}@ece.gatech.edu

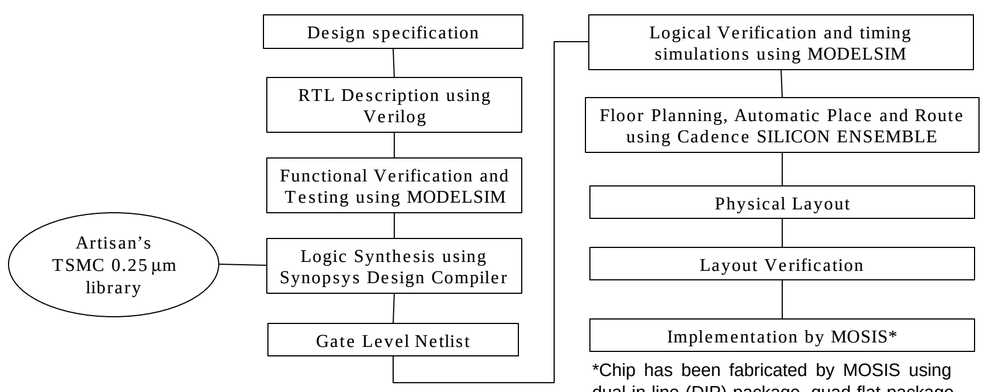
Introduction

- First-ever VLSI implementation of wavelet and wavelet-based golay error control codes [1, 2]
- Wavelet code (12, 6, 4)* corrects 1-bit errors
- Wavelet-based golay (24, 12, 8)* corrects up to 3-bit errors

*(N, M, d) : (N=code length, M=message length, d=distance)

[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, pp. 355-364, 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.

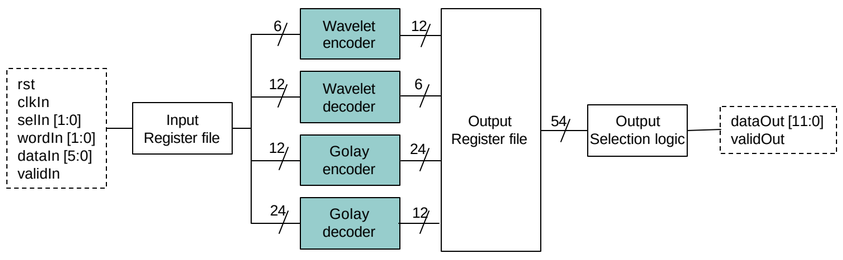
Design Flow



*Chip has been fabricated by MOSIS using dual in line (DIP) package, quad flat package (QFP) as well as bare die

Architecture

I/O

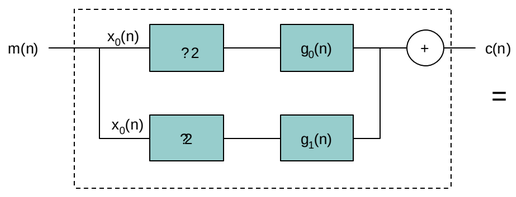


- RTL descriptions of the circuit designed in Verilog
- Four encoding and decoding modules (*selIn* signal chooses one of encoding/decoding modules)
- The width of Input and output is optimized for the wavelet encoder (i.e., 6-bit input and 12-bit output)
- 12- and 24-bit input requires 2 and 4 cycles, respectively; 24-bit output requires 2 cycles

Encoding/decoding

- Encoding/decoding functions are implemented largely in combinational *XOR* logic
- Wavelet encoding/decoding and wavelet-based golay encoding are implemented in single stage combinational block
- Wavelet-based golay decoder uses a sequential logic block with a latency of 12 cycles

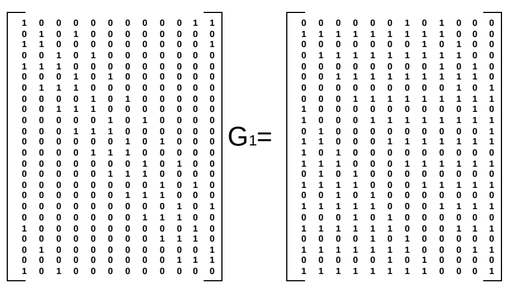
Wavelet-based golay encoder



Filter bank structure of wavelet-based golay encoder

$= c(n) = (G_0 + G_1)(m(n))^*$

G_0= G_1=

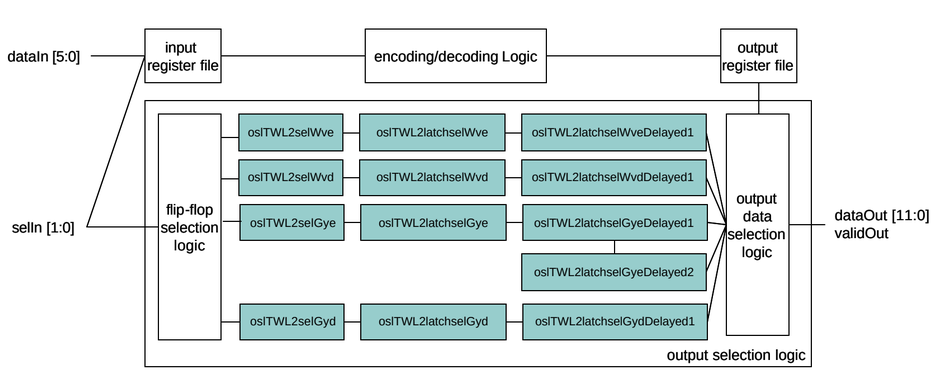


m(n): message,
c(n): codeword
n: nth sequence
G_0: matrix driven from upsampling and scaling sequence($g_0(n)$)
G_1: matrix driven from upsampling and mother wavelet ($g_1(n)$)

c(0) = m(0) ⊕ m(6) ⊕ m(8) ⊕ m(10) ⊕ m(11)
c(1) = m(0) ⊕ m(2) ⊕ m(4) ⊕ m(5) ⊕ m(6) ⊕ m(7) ⊕ m(8)
c(2) = m(0) ⊕ m(1) ⊕ m(7) ⊕ m(9) ⊕ m(11)
c(3) = m(1) ⊕ m(3) ⊕ m(5) ⊕ m(6) ⊕ m(7) ⊕ m(8) ⊕ m(9)
c(4) = m(0) ⊕ m(1) ⊕ m(2) ⊕ m(8) ⊕ m(10)
c(5) = m(2) ⊕ m(4) ⊕ m(6) ⊕ m(7) ⊕ m(8) ⊕ m(9) ⊕ m(10)
c(6) = m(1) ⊕ m(2) ⊕ m(3) ⊕ m(9) ⊕ m(11)
c(7) = m(3) ⊕ m(5) ⊕ m(7) ⊕ m(8) ⊕ m(9) ⊕ m(10) ⊕ m(11)
c(8) = m(0) ⊕ m(2) ⊕ m(3) ⊕ m(4) ⊕ m(10)
c(9) = m(0) ⊕ m(4) ⊕ m(6) ⊕ m(8) ⊕ m(9) ⊕ m(10) ⊕ m(11)
c(10) = m(1) ⊕ m(3) ⊕ m(4) ⊕ m(5) ⊕ m(11)
c(11) = m(0) ⊕ m(1) ⊕ m(5) ⊕ m(7) ⊕ m(9) ⊕ m(10) ⊕ m(11)
c(12) = m(0) ⊕ m(2) ⊕ m(4) ⊕ m(5) ⊕ m(6)
c(13) = m(0) ⊕ m(1) ⊕ m(2) ⊕ m(6) ⊕ m(8) ⊕ m(10) ⊕ m(11)
c(14) = m(1) ⊕ m(3) ⊕ m(5) ⊕ m(6) ⊕ m(7)
c(15) = m(0) ⊕ m(1) ⊕ m(2) ⊕ m(3) ⊕ m(7) ⊕ m(9) ⊕ m(11)
c(16) = m(2) ⊕ m(4) ⊕ m(6) ⊕ m(7) ⊕ m(8)
c(17) = m(0) ⊕ m(1) ⊕ m(2) ⊕ m(3) ⊕ m(4) ⊕ m(8) ⊕ m(10)
c(18) = m(3) ⊕ m(5) ⊕ m(7) ⊕ m(8) ⊕ m(9)
c(19) = m(1) ⊕ m(2) ⊕ m(3) ⊕ m(4) ⊕ m(5) ⊕ m(9) ⊕ m(11)
c(20) = m(4) ⊕ m(6) ⊕ m(8) ⊕ m(9) ⊕ m(10)
c(21) = m(0) ⊕ m(2) ⊕ m(3) ⊕ m(4) ⊕ m(5) ⊕ m(6) ⊕ m(10)
c(22) = m(5) ⊕ m(7) ⊕ m(9) ⊕ m(10) ⊕ m(11)
c(23) = m(1) ⊕ m(3) ⊕ m(4) ⊕ m(5) ⊕ m(6) ⊕ m(7) ⊕ m(11)

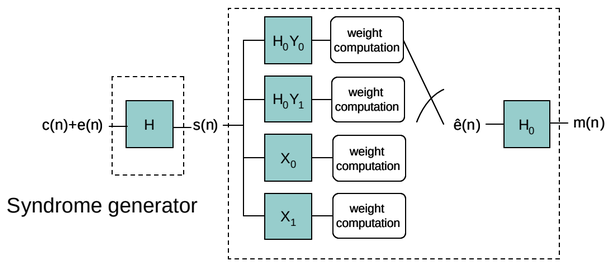
Implementation of wavelet-based golay encoder using XOR

Output selection



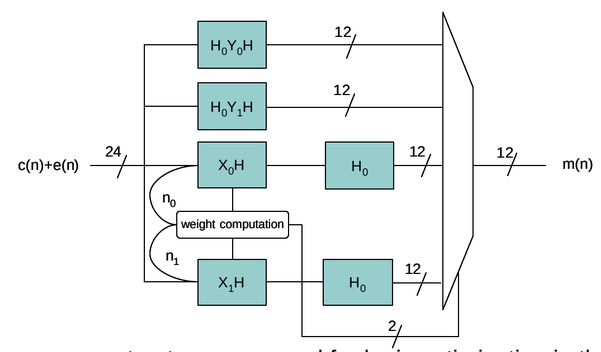
- Output selection logic selects 12-bit data from the 54-bit output register file
- Output selection logic uses four internal pipelines
- Wavelet-based golay encoder has extra 1 stage to transfer 24-bit data using 12-bit wide output port

Wavelet-based golay decoder



Syndrome generator

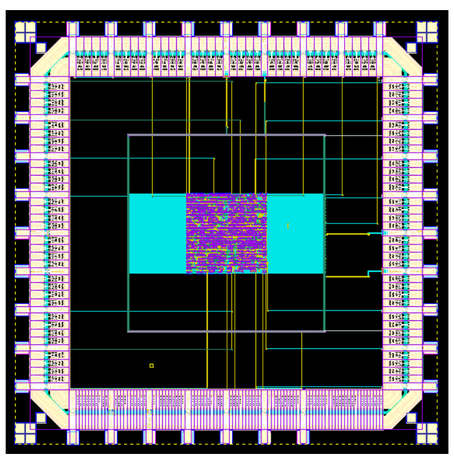
Message reconstructor



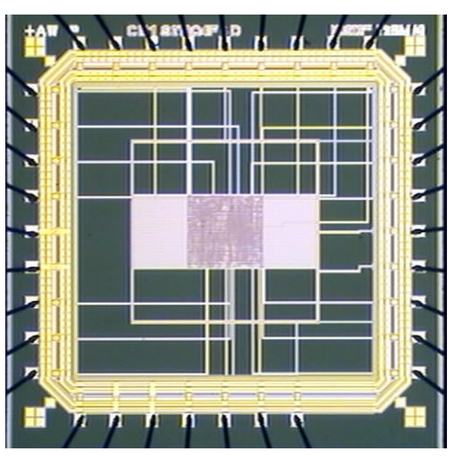
- Wavelet-based golay decoder algorithm consists of syndrome generator and message reconstructor
- A syndrome (s(n)) is a sequence that is uniquely associated with a set of codewords within hamming distance d
- e(n) is error pattern in communication channel
- Shared rectangles represent filter banks
- Message reconstructor generates message according to the weight computation
- Wavelet-based golay decoder can detect transmission error less than (d-1) bits and correct errors upto 3 bits

- Syndrome and message reconstructor are merged for logic optimization in the wavelet-based golay decoder architecture
- Filter banks in shaded rectangle are implemented with one stage XOR operations
- 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
- Since M=12, a maximum of 12 cycles required for all possible weight computations

Layout

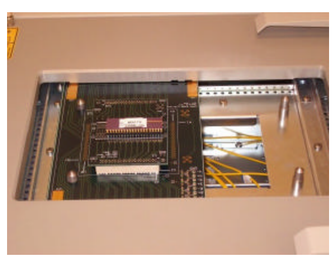


Physical layout prior to fabrication
Silicon size: 2637 x 2640 microns=6.9mm^2
(as given by MOSIS)



Chip layout with wire connections to the DIP after fabrication (taken at Georgia Tech)

Testing & Result



- Tested using HP 83000 Digital IC Test system
- The encoder/decoder logic has been successfully tested for its functionality
- A clock period of 6.9 ns (a speed of 145 MHz) achieved
- The effective data throughput is 145Mhz x 6bits=870Mb/sec.