International Journal of Computer Science and Informatics

Volume 3 | Issue 3 Article 9

January 2014

DESIGN OF CONFIGURABLE IP CORE FOR ERROR DETECTION AND CORRECTION

AJILESH RK
National Institute of Electronics and Information Technology (NIELIT), Calicut, Kerala India,
itsmeunnirk@gmail.com

ANAND K
National Institute of Electronics and Information Technology (NIELIT), Calicut, Kerala India,
anandkcalicut@gmail.com

NANDAKUMAR. R
National Institute of Electronics and Information Technology (NIELIT), Calicut, Kerala India,
nanda24x7@gmail.com

SREEJEESH.S. G
National Institute of Electronics and Information Technology (NIELIT), Calicut, Kerala India,
sreejisg@gmail.com

DESIGN OF CONFIGURABLE IP CORE FOR ERROR DETECTION AND CORRECTION

AJILESH RK[1], ANAND K[2], NANDAKUMAR.R[3] & SREEJEESH.S.G[4]

[1,2,3,4]National Institute of Electronics and Information Technology (NIELIT), Calicut, Kerala India
Emai:{itsmeunnirk, anandkcalicut, nanda24x7,sreejisg }@gmail.com

Abstract— This paper addresses the design & implementation of configurable Intellectual Property (IP) core for double error detection and single error Correction. The encoding /decoding algorithms considered in this can be implemented with a simple and faster hardware. The block can be used for coding and decoding word having any length and correct single bit error occurred and detect double bit error, during transmission. The user can define the word length and the hamming bits required.

Keywords - Hamming Code, Error Correction, IP Core, Parity Error, ASIC.

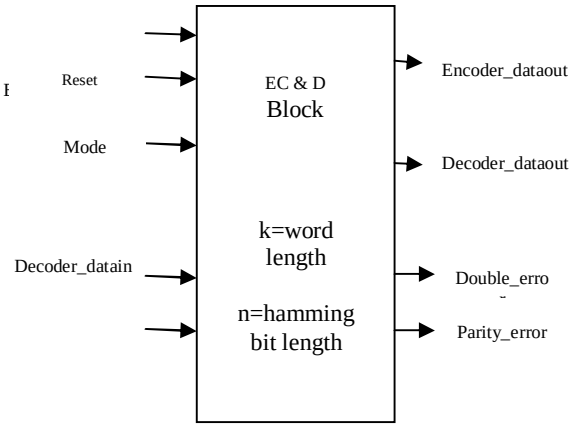
I. INTRODUCTION

When data is transmitted from one location to another there is always the possibility that an error may occur. There are a number of reliable codes that can be used to encode data so that the error can be detected and corrected. With this project we will explore a simple error detection-correction technique which can be used for word having any length, using Hamming Code. A Hamming Code can be used to detect and correct one-bit change in an encoded code word. By adding an extra parity, two bit changes can be detected. Verilog HDL is used to develop the hardware**.**

A. Forward error correction:
Digital communication systems, particularly those used in military, need to perform accurately and reliably even in the presence of noise and interference. Among many possible ways to achieve this goal, forward error-correction coding is the most effective and economical. Forward error-correction coding (also called 'channel coding') is a type of digital signal processing that improves reliability of the data by introducing a known structure into the data sequence prior to transmission.

This structure enables the receiving system to detect and possibly correct errors caused by corruption from the channel and the receiver. As the name implies, this coding technique enables the decoder to correct errors without requesting retransmission of the original information. Hamming code is a typical example of forward error correction. In a communication system that employs forward error-correction coding, the digital information source sends a data sequence to an encoder. The encoder inserts redundant (or parity) bits, thereby outputting a longer sequence of code bits, called a 'code word.' These code words can then be transmitted to a receiver, which uses a suitable decoder to extract the original data sequence.

II. BLOCK DIAGRAM



Block diagram description:
This block have 5 inputs as clk, reset, mode, encoderdatain ('k'bits) and decoderdatain('n+k+1'bits).Outputs are encoderdataout('n+k+1'bitst),decoderdataout('k'bits) ,parity_error and double_error.The user can change the value of word length 'k' and hamming code length 'n' according to their requirement.if reset is set as logic low ,whole block will be reset and for mode=1, the block works as encoder and for mode=0,it works as Decoder. The working of encoder and Decoder are explained in below

III. IP CORE OPERATION:

A. Encoder:

The encoder has a 'k' bit input datain(D) and one output dataout(n+k+1).for a word having k bit length n bit length hamming code must use.this n can be calculated by equation $2^n \geq n+k+1$. Here we are using 'n' hamming (R1,R2,R3…..Rn) bits and one parity bit (p1).

Here for calculating the hamming bit , we are loading the 'k' bit length word to a (n+k+1) width register where 0 and power of 2 positions are loaded whit

zeros. Each positions are numbered from 1 to (n+k+1).next step is to find out the '1' in register starts from LSB.Let 8 bit word be (D8 D7 D6 D5 D4 D3 D2 D1) each D represent each bit (1 or 0) and the preceding number represents the position in 8bit word. Calculation of each hamming bit is done by xoring selected bits from the 8 bit word as follows.

R1 = (D1 xor D2 xor D4 xor D5 xor D7)
R2 = (D1 xor D3 xor D4 xor D6 xor D7)
R3 = (D2 xor D3 xor D4 xor D8)
R4 = (D5 xor D6 xor D7 xor D8)

These hamming bits are to be interspersed at bit positions 2n (n = 0, 1, 2, 3) with the original data bits. Hamming bit and data bit positions are shown below

13	12	11	10	9	8	7	6	5	4	3	2	1
P1	D8	D7	D6	D5	r4	D4	D3	D2	r3	D1	r2	r1

and the parity bit is calculated as given below

P1=(D1 xor D2 xor D3 xor D4 xor D5 xor D6 xor D7 xor D8 xor r1 xor r2 xor r3 xor r4)

These 13 bits are transmitted as Dataout.

B. Decoder:

In decoder section, 13 bit code word(D13 ,D12 ,D11,D10,D9,D8,D7,D6,D5,D4,D3,D2,D1) received as Datain and decoded. For this, hamming bits and parity bit are calculating for this 13 bit code word (as done in encoder).
R1=D1 xor D3 xor D5 xor D7 xor D9 rxor D11
R2=D2 xor D3 xor D6 xor D7 xor D10 xor D11
R3=D4 xor D5 xor D6 xor D7 xor D12
R4=D8 xor D9 xor D10 xor D11 xor D12
parity=D1 xor D2 xor D3 xor D4 xor D5 xor D6 xor D7 xor D8 xor D9 xor D10 xor D11 xor D12 xor D13
Using these hamming bits and parity bits, error correction and detection are done.
If the calculated hamming bit(in R4 R3 R2 R1 order) is 0000 and 0 ,then there is no error ≥occurred in the data. So the actual word bit can be extract from the code word whit out any modifications.

If the calculated hamming bit(in R4 R3 R2 R1 order) is a none zero value and parity bit is 1 ,then there is an error occurred in bit position in 13 bits word, which is indicated by the hamming code. So, by inverting the bit pointed by the hamming code, the correct word can be retrieve.

If the calculated hamming bit (in R4 R3 R2 R1 order) is a none zero value and parity bit is 0 ,this indicates that error occurred in two bit positions in the code word and the output Double_error become high.

If the calculated hamming bit (in R4 R3 R2 R1 order) is 0000 and parity bit is 1 ,then output Parit_ error become high, which indicates that error occurred in the parity bit.

C. *Conditions for error detection:*

Hamming bit value	Parity bit	Error type	Note
0	0	No error	
≠0	1	Single error	correctable
≠0	0	Double error	No correctable
0	1	Parity error	

D. Numerical example:

At the encoder:
Input data Datain=11010011

D8	D7	D6	D5	D4	D3	D2	D1
1	1	0	1	0	0	1	1

Hamming bits:
R1=1 xor 1 xor 0 xor 1 xor 1= 0
R2=1 xor 0 xor 0 xor 0 xor 1=0
R3=1 xor 0 xor 0 xor 1=0
R4=1 xor 0 xor1 xor 1=1
Parity bit p1=0 Dataout= 0110110010100

13	12	11	10	9	8	7	6	5	4	3	2	1
P1	D8	D7	D6	D5	R4	D4	D3	D2	R3	D1	R2	R1
0	1	1	0	1	1	0	0	1	0	1	0	0

At the decoder:
Transmitted data is same as the dataout of above encoder. Here input of Decoder

Case 1) Datain:0110110010100 (No Error)

13	12	11	10	9	8	7	6	5	4	3	2	1
P1	D8	D7	D6	D5	R4	D4	D3	D2	R3	D1	R2	R1
0	1	1	0	1	1	0	0	1	0	1	0	0

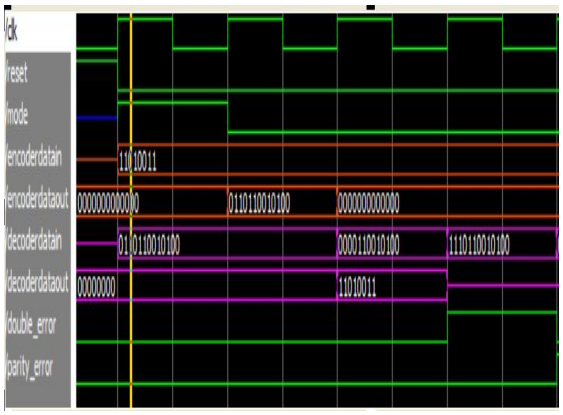
R1=0 xor 1 xor 1 xor 0 xor 1 xor 1=0
R2=0 xor 1 xor 0 xor 0 xor 0 xor 1=0
R3=0 xor 1 xor 0 xor 0 xor 1=0
R4=1 xor 1 xor 0 xor 1 xor 1=0
parity=0 xor 0 xor 1 xor 0 xor 1 xor 0 xor 0 xor 1 xor 1 xor 0 xor 1 xor 1 xor 0=0hamming bir=0000 and Parity=0
Dtaout=11010011 (same as actual word)

Case 2) Datain: 0010110010100(Error occurred in the 12th bit position)
R1=0 ; R2=0 ; R3=1 ; R4=1 and parity=0 so error occurred in 12th bit position(1100)
By inverting this bit, Dataout=11010011

Case 3) Datain: 0000110010100(Error occurred in the 12th and 11th bit position)
R1=1 ; R2=1 ; R3=1 ; R4=0 and parity=0 so double bit error occurred, which cannot be corrected

Case 4) Datain: 1110110010100(Error occurred in the parity bit position)
R1=0 ; R2=0 ; R3=0 ; R4=0 and parity=1 . so Parrity_error become 1.

IV. SIMULATION RESULTS:



Function Table (Ecoder):

Datain	Reset	clk	Dataout
11010011	1	1	0000000000000
11010011	0	1	0110110010100

Function Table (Decoder):

Datain	Reset	clk	Dataout	Double_error	Prity_error
0110110010100	1	1	00000000	0	0
0110110010100	0	1	11010011	0	0
0010110010100	0	1	11010011	0	0
0000110010100	0	1	xxxxxxxx	0	1
1110110010100	0	1	zzzzzzzz	1	0

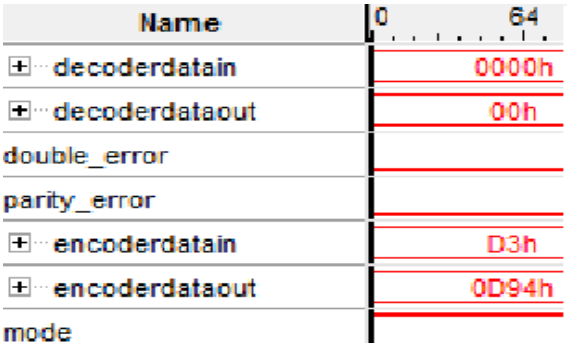
V. FPGA PROTOTYPING RESULTS:

a)Synthesis Result

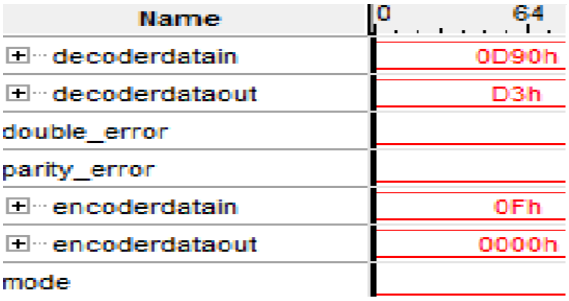
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	61 / 33,216 (< 1 %)
Total combinational functions	54 / 33,216 (< 1 %)
Dedicated logic registers	31 / 33,216 (< 1 %)
Total registers	31
Total pins	47 / 475 (10 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

b) Hardware Test Result:

Encoder (Mode=1)



Decoder (Mode=0)



c) Power analysis result:

PowerPlay Power Analyzer Summary	
PowerPlay Power Analyzer Status	Successful - Thu Jan 19 14:26:59 2012
Quartus II Version	11.0 Build 157 04/27/2011 SJ Full Version
Revision Name	value_hamming_core
Top-level Entity Name	value_hamming_core
Family	Cyclone II
Device	EP2C35F672C6
Power Models	Final
Total Thermal Power Dissipation	114.64 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	79.94 mW
I/O Thermal Power Dissipation	34.70 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data

VI. ASIC IMPLEMENTATION RESULTS:

A) Area utilization Result:

Report : area
Design : IPCORE_EDCB
****

Library(s) Used: lsi_10k
Number of ports: 47
Number of nets: 165
Number of cells: 140
Number of combinational cells: 117
Number of sequential cells: 23
Number of macros: 0
Number of buf/inv: 21
Number of references: 21
Combinational area: 192.000000
Noncombinational area: 185.000000
Total cell area: 377.000000

Total area: undefined
****

B) Timing Analysis Result:

Design : IPCORE_EDCB
****

Path Type: max

Point	Incr	Path
parity_error_reg/CP (FD1)	0.00	0.00 r
parity_error_reg/QN (FD1)	3.92	3.92 r
decoderdataout_tri[8]	0.75	4.67 f
decoderdataout[8] (out)	0.00	4.67
data arrival time		4.67

C) ***Power Analysis Result (ASIC):***

****

Design : IPCORE_EDCB
Version: E-2010.12-SP2
****

Global Operating Voltage = 5
Power-specific unit information :
Voltage Units = 1V
Capacitance Units = 0.100000ff
Time Units = 1ns
Dynamic Power Units = 100nW
Leakage Power Units = Unitless
Cell Internal Power = 0.0000 nW (0%)
Net Switching Power = 78.5172 uW (100%)
Total Dynamic Power = 78.5172 uW (100%)

VII. CONCLUSION:

An IP core for error detection and correction has been designed and the hardware performance of the same has been carried out, as discussed above. The IP core finds wide spectrum of applications in secure communication in addition to its utility in computer architectures like that for [6] Soft Error and Variation-Induced Defect Tolerance for memories particularly like SDRAM, Cache, CAM etc.

REFERENCES

[1] R. Micheloni, A. Marelli And R. Ravasio, "Error Correction Code For Non-Volitile Memory," *Qimonda Italy Srl, Design Center Vimercate, Italy*.

[2] D.G Hoffman, *D.A Leaonard,C.C Lindner,K.T Phelps,C.A Rodger,J.R Wall*,

"Coding Theory-The Essentials"

[3] Simon Tam"Single Error Correction And Double Error Detection," Xilinx application Notes, XAPP645(V2.2)AUG 9 2009.

[4] Hamming, R. W. "Error Detecting And Error Correcting Codes." *Bell Systemtech. Jour.*, 29 (1950): 147–160.

[5] S. Lin and D. J. Costello, Jr., Error Control Coding:, II Ed., Prentice Hall, Englewood Cliffs, NJ, 2004

[6] Luong D. Hung, Hidetsugu Irie, Masahiro Goshima, Shuichi Sakai, Utilization of SECDED for Soft Error and Variation-Induced Defect Tolerance in Cache,available online at http://www.mtl.t.utokyo.ac.jp/publications/paper/2007/E07-coference-hung-1.pdf