

“Implementing (7, 4) Hamming Code Encoding And Decoding System Using CPLD”

Leena¹, Mr. Subham Gandhi², Mr. Jitender Khurana³
PG Student, Associate Professor, Associate Professor
Deptt. Of ECE, SBMN Engg. College, AsthalBohar, Rohtak

Abstract: In this paper a new method has been developed to detect and correct the errors during transmission of signal using Hamming Code. The theory of error-correcting codes is a relatively recent application of mathematics to information and communication systems. In order to do that the proposed method uses a complex programmable logic device (CPLD). It is known that CPLD provides quick implementation and fast hardware verification. It gives facilities of reconfiguring the design construct unlimited number of times. Hence, an attempt is made to implement the Hamming Code for detecting and correcting errors.

Index Terms: -Complex Programmable Logic Device (CPLD), Very High Speed Integrated Circuit Hardware Description language (VHDL),

1. INTRODUCTION

Coding theory is concerned with reliability of communication over noisy channels. Error correcting codes are used in a wide range of communication systems from deep space communication, to quality of sound in compact disks and wireless phones. In computers, data of any kind is stored and processed as binary digits (or bits for short). A bit is a 0 or a 1. Every letter has an ASCII code. For example, the ASCII code of the letter ‘A’ is 01000001. Typically, data consists of billions of bits. Digital data is transmitted over a channel (which could be a wire, network, space, airc.,) And there is often noise in the channel. The noise may distort the messages to be sent. Therefore, what the receiver receives may not be the same as what the sender sends. The goal of coding theory is to improve the reliability of digital communication by devising methods that enable the receiver to decide whether there have been errors during the transmission (error detection), and if there are, to possibly recover the original message (error correction).[1]

Hamming's development [Ham] is a very direct construction of a code that permits correcting single-bit errors. He assumes that the data to be transmitted consists of a certain number of information bits, and he adds to these a number of check bits ‘p’ such that if a block is received that has at most one bit in error,

then ‘p’ identifies the bit that is in error (which may be one of the check bits). Specifically, in Hamming code ‘p’ is interpreted as an integer which is 0 if no error occurred, and otherwise is the 1-origin index of the bit that is in error. Let ‘k’ be the number of information bits, and ‘m’ the number of check bits used. Because the check bits must check themselves as well as the information bits, the value of ‘p’, interpreted as an integer, must range from 0 to which is distinct values. Because ‘m’ bits can distinguish cases, we must have

$$2^m \geq m+k+1$$

This is known as the Hamming rule[2].

This project attempts to correct and detect the errors using VHDL. As when we transmit any signal its bits can be changed due to external interference so the signal gets corrupted. Hamming codes are used to correct the error and when it is implemented in VHDL. Being an electronic system it is reliable, compact and maintenance free. VHDL makes the system versatile as the on off times can be easily varied by changing the delay loops through software. [3]

2. HAMMING CODE ENCODER (7,4)

Hamming Code (7,4) we take 4 input bits and 3 parity bits are added as redundant bits so the signal transmitted is 7 bits. The position of parity bits is ‘2^m’ where m is the parity bits added.

Implementing Hamming Code in VHDL is a reliable technique and with this method single error is corrected and double errors are detected.

FLOW CHART FOR HAMMING ENCODER

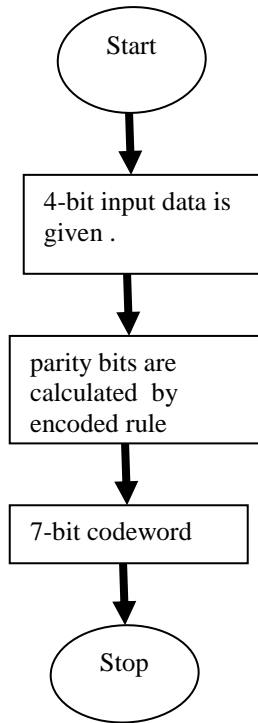


Fig 1. Flow Chart of Encoder

In a world of error free transmission decoding a hamming code is so much easy but this can not attained in real life. So for calculating error the flow chart is given below.

FLOW CHART FOR HAMMING ENCODER

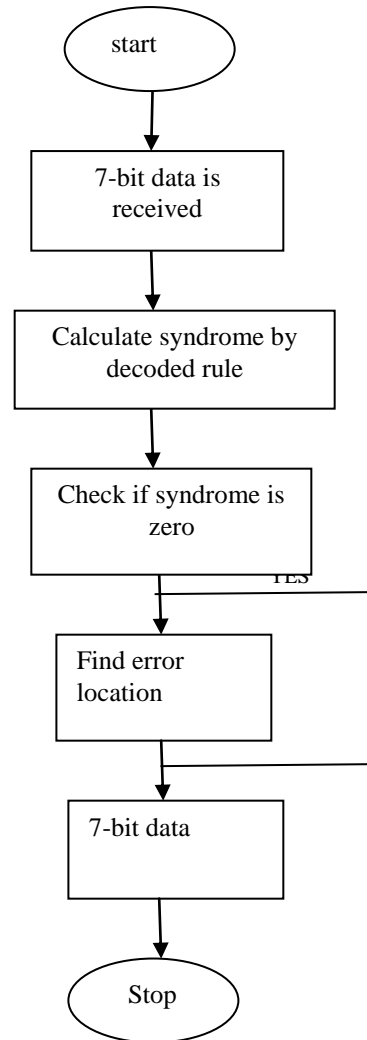


Fig 1. Flow Chart of Encoder

4. CPLD IMPLEMENTATION

Complex Programmable Logic Device (CPLD) A Complex Programmable Logic Device (CPLD) is a programmable logic device with complexity between that of PALs and FPGAs, and architectural features of both. The maximum clock frequency is 20 MHz and hence it is faster than microcontroller. The use of CPLDs (Field Programmable Gate Arrays) and configurable processors is an interesting new phenomenon in embedded development is a

combination of a fully programmable AND/OR array and a bank of macro cells. The AND/OR array is reprogrammable and can perform a multitude of logic functions.. Complex Programmable Logic Devices (CPLDs) are exactly what they claim to be. These are designed to appear just like a large number of PALs in a single chip, connected to each other through a cross point switch. They use the same development tools and programmers, and are based on the same technologies, but they can handle much more complex logic and more of it.

Xilinx is a vendor of CPLD products and manufactures a family known as the XC9500. In this project Xilinx ISE8.1 version is used. CPLDs enable ease of design, lower development costs, and more product revenue for your money, and the opportunity to speed your products to market. Ease of Design: CPLDs offer the simplest way to implement a design. Once a design has been described, by schematic and/or HDL entry, you simply use CPLD development tools to optimize, fit, and simulate the design. The development tools create a file that is used to customize (that is, program) a standard off-the-shelf CPLD with the desired functionality. This provides an instant hardware prototype and allows the debugging process to begin. If modifications are needed, you can enter design changes into the CPLD development tool, and re-implement and test the design immediately.



Fig 3. CPLD kit

5. PROPOSED WORK

In this paper we are implementing the error detection and correction of Hamming Code using CPLD kit on VHDL software. At the sender end we have 4 –bit data and 3-parity bits are added to make it 7-bit codeword..Now, 7-bit data is ready to send.

At the receiver end the 7-bit codeword is received. The receiver decode the codeword. During transmission if an error is introduced due to external interference or due to noise then an error occurs. This error is detected at receiver.

6. IMPLEMENTATION OF HAMMING CODE ON CPLD USING VHDL

A. ENCODING OF HAMMING CODE

Now, the design of Hamming Code (7, 4) is to be done on CPLD kit using VHDL. (7,4) means that there are 4-data bits and we need 3-parity bits to send along with these data bits to make it 7-bit codeword. Even parity is used in.Data bits are d2,d4,d5,d6; and parity bits are p1,p2,p3.

Where d0 is 1st bit which is given as : $d0=2^0=1$
 D1 is 2nd bit and is given as : $d1=2^1=2$
 D3 is 4th bit and is given as : $d3=2^2=4$

D0=	0	0	0	1
D1=	0	0	1	0
D2=	0	0	1	1
D3=	0	1	0	0
D4=	0	1	0	1
D5=	0	1	1	0
D6=	0	1	1	1

P1=d0, d2, d4, d6

P2=d1 ,d2 ,d4,d6;

P3=d3, d4,d5,d6;

0	1	1	P3	0	P2	P1
D6	D5	D4	D3	D2	D1	D0

Where 0110 is data and p1,p2,p3 are parity bits which take the position 2^m where m =0,1,2,3-----so on . which are check bits.

For calculating p1:-
 Count the number of 1's in d1,d3,d5,d7 ,if there are even number of 1 then p1=0 otherwise it is 1.
 Therefore, p1=1

For calculating p2:-
 Count the number of 1's in d2,d3,d6,d7 ,if there are even number of 1 then p1=0 otherwise it is 1.
 Therefore, p2=1

For calculating p3:-
 Count the number of 1's in d4,d5,d6,d7 ,if there are even number of 1 then p1=0 otherwise it is 1.
 Therefore, p3=0

Therefore the 7-bit codeword is

0	1	1	0	0	1	1
D6	D5	D4	D3	D2	D1	D0

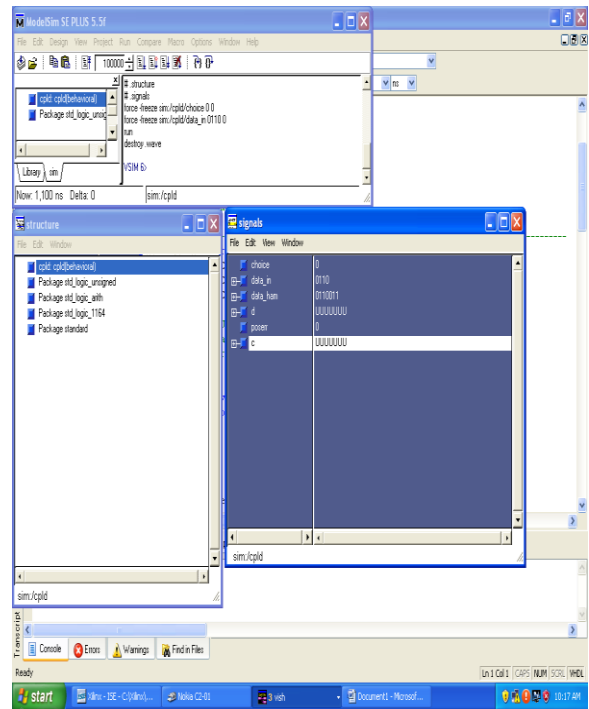


Fig5. 7-bit data

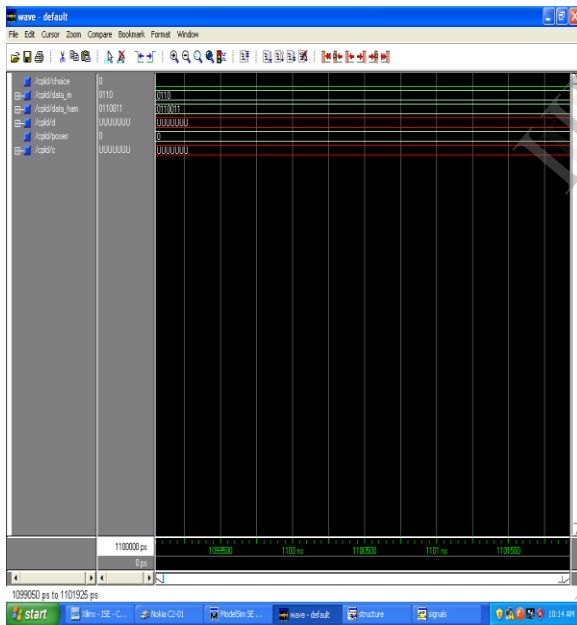


Fig 4. 7-Bit Codeword

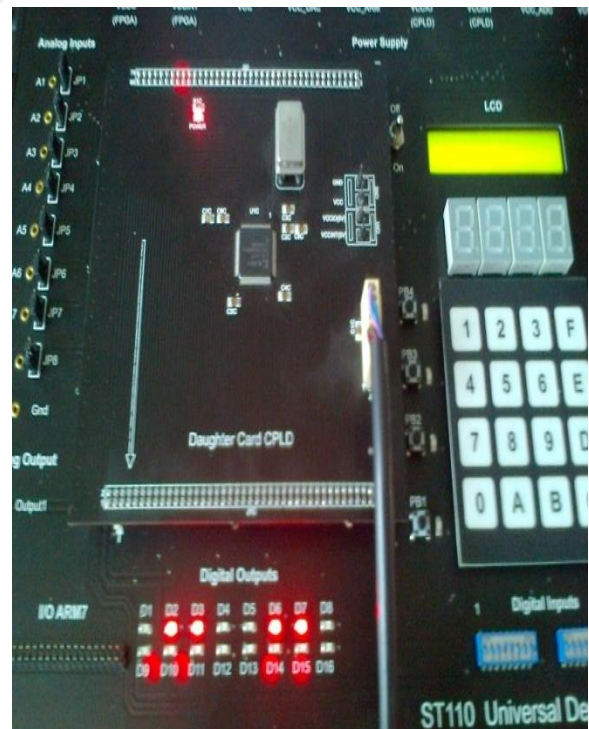
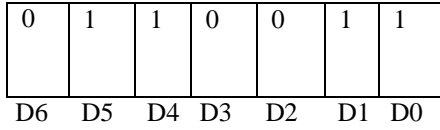


Fig 6. On CPLD kit

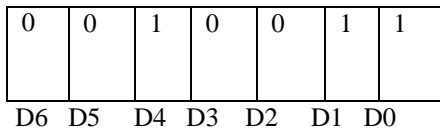
B. DECODING OF HAMMING CODE

When this 7-bit codeword is sent then during transmission error occurs which change the bits from 1 to 0 or 0 to 1.

7 – bit codeword at Transmitter



7 – bit codeword at Receiver



Thus an error occurs at 5-th bit

So the error is detected using VHDL

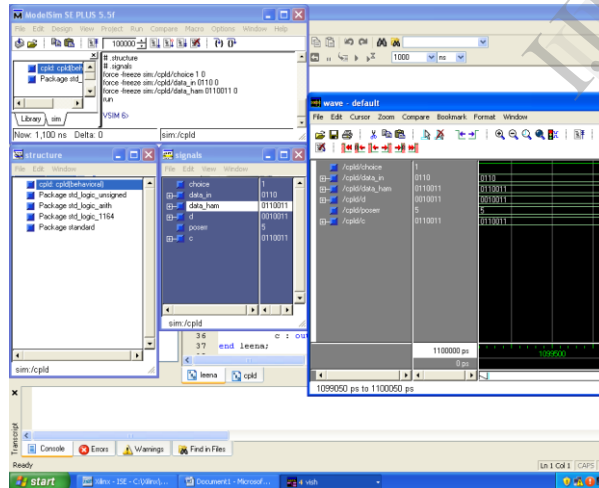
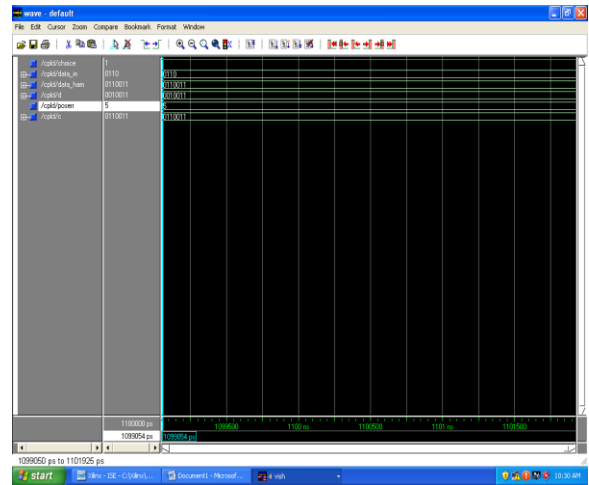


Fig 7. Data with error



ERROR ON 5-BIT

Thus the error is detected and it is corrected for successful transmission of data.

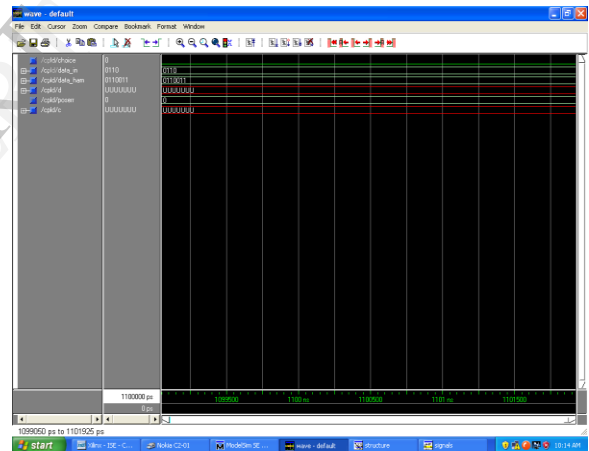


Fig 8. Data after correction.

7. CONCLUSION

Thus from this we concluded that it is easy to detect and correct error in VHDL software as compared to hardware as it is much complicated but in software we can implement it many times we want.

REFERENCES

- [1] Nuh Aydin:An Introduction to Coding Theory via Hamming Codes Department of Mathematics Kenyon College .
- [2] Hamming, R. W. 1950. Error-detecting and error-correcting codes. Chapter15-2.
- [3] Rashmi, Anurag Sharma, VHDL based Multiple Traffic Lights Controller, International Journal of electronics & communication technology 259 IJECT Vol.2, Issue 3sept- 2011
- [4] R. E. Blahut, "Theory and Practice of Error Control Codes," Addison-Wesley, New York, 1983
- [5] Behrouz A. *Forouzan: Data Communications and Networking*

IJERT