Design and Simulation of Error Correction Codes

Chaitra Vardhini H1, Prof. Madhu Patil2
M. Tech Scholar, Department of ECE, NMIT, Bangalore, India1
Associate Professor, Department of ECE, NMIT, Bangalore, India2

Abstract: Natural interference like EMI, noise, crosstalk can happen over the communication channel like memory, which causes the original data to be different from the stored data. In order to find these errors a few techniques to recognize and correct the error is required. This work is an overview of different Error Correction techniques. These techniques guarantee to find and possibly correct the errors brought about by the stuck-at faults in the memory. The work is mainly focused on Hamming codes, Convolution codes, CRC and Turbo codes.

Keywords: Hamming codes, Convolution codes, CRC - Cyclic Redundancy Check and Turbo codes.

I. INTRODUCTION

Environmental interference and physical defects like noise, EMI and crosstalk in the communication channel can cause random errors during data transmission. The method of detecting and correcting these random errors to ensure information is transferred intact from its source to its destination is called Error Coding. Error coding is used for fault tolerant computing in computer memory, satellite communication, space communication, magnetic data storage and optical storage media network communications, cellular telephonic communications, and digital data communication. Error coding encodes the information bits into longer bits called code words for transmitting the data. These code words can be decoded at reception in order to get the original information bits. Redundancy is because of the extra bits that is present in the code word that will allow the reception to use the decoding process to find if the communication medium introduced errors and in some cases correct them so that the data need not be retransmitted.

II. METHODOLOGY

Initially, the input data bits are given to the encoder, where encoder converts input data bits into code words. Then this encoded sequence is stored on to the memory. Memory is introduced with a stuck-at-fault (error) which might be either stuck-at-0 or stuck-at-1. The data in the memory is read and given as the input to the decoder. Decoder detects the fault and possibly the position of the fault present in the memory.

III. HAMMING CODES

Hamming codes is called after its designer; Richard Hamming from Bell Labs is a Linear Error Correction Codes. Hamming codes can detect two bit errors and correct single bit error. If the Hamming distance between the sent data and the received data bit is less than or equal to one then reliable communication is possible. It is not reasonable for transmission circumstance where burst errors happen. Hamming Codes (n, k) adds n-k no of check bits to every k-bits of message data.

IV. CONVOLUTIONAL CODES

Convolutional codes are designed using two parameters:

1. The code rate
2. The constraint length

Code rate – (k/n) is defined as the ratio of the number of bits input to the encoder to the number of bits output by the encoder.

Constraint length (k) denotes the length of convolutional encoder, i.e. how many k-bit stages are available to input the combinatorial logic that produces the output code words.

‘m’ indicates the number of shift registers in the encoder design. It is simply the memory length of the encoder.

Convolutional codes are used in real time communication systems for error correction. The code words depend on the present ‘k’ message bits and some past input bits. The
main decoder combination of convolutional encoder is Viterbi Algorithm.

Figure 2: k=1, n=2 and r=1/2 Convolutional Encoder.

Figure 3: Viterbi Decoder Block Diagram.

Figure 4: Example of Convolution Codes.

V. CYCLIC REDUNDANCY CHECK – CRC

CRC - Cyclic Redundancy Check is a technique for detecting errors in digital information transmission, however not for error correction after errors are recognized. In the CRC strategy, some number of check bits is added. These check bits are called checksum. They are connected to the information bits that are being stored. The beneficiary figures out whether the check bits concur with the information bits. On the off chance that an error has happened then fault location in the memory is detected and replaced with the fault free memory.

VI. TURBO CODES

In 1993 Berrou, Glavieux and Thitimajshima proposed “another class of convolution codes called as turbo codes”. It a class of Forward Error Correction Codes widely used in communication system. A turbo code is a parallel concatenation of many RSC codes. The encoder consists of two identical RCS – Recursive Systematic Encoder in parallel. The output of encoder1 and encoder2 are got from the same massage bits. The output code rate of both the encoders is at a rate R=1/3 code.

Figure 5: Example of Generated code words using CRC.

Figure 6: Turbo Encoder Block Diagram.

Figure 7: Turbo Encoder example
VII. SIMULATION RESULTS

Figure 8: Hamming codes implemented for SRAM.

Figure 9: Hamming codes implemented for Dual Port Memory.

Figure 10: Convolutional Encoder generated code words.

Figure 11: Viterbi Decoder output.

Figure 12: Generated code words using CRC.

Figure 13: CRC code words stored on Memory with error.

Figure 14: Output of CRC using Chipscope Pro

Figure 15: Turbo Encoder output.

VIII. COMPARISON OF ERROR CORRECTION CODES TECHNIQUES

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>SRAM</th>
<th>DUAL PORT MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Slice Latches</td>
<td>8(1%)</td>
<td>15(1%)</td>
</tr>
<tr>
<td>No. of occupied Slices</td>
<td>12(1%)</td>
<td>18(1%)</td>
</tr>
<tr>
<td>Total no. of 4 input LUTs</td>
<td>25(1%)</td>
<td>21(1%)</td>
</tr>
<tr>
<td>Maximum path delay</td>
<td>8.348ns</td>
<td>6.530ns</td>
</tr>
<tr>
<td>Total On-Chip Power</td>
<td>69.18mW</td>
<td>60.56mW</td>
</tr>
</tbody>
</table>

Figure 16: Hamming Codes comparison with SRAM and Dual Port Memory.

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>SRAM</th>
<th>DUAL PORT MEMORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Slice Latches</td>
<td>8(1%)</td>
<td>368(5%)</td>
</tr>
<tr>
<td>No. of occupied Slices</td>
<td>8(1%)</td>
<td>336(9%)</td>
</tr>
<tr>
<td>Total no. of 4 input LUTs</td>
<td>14(1%)</td>
<td>325(4%)</td>
</tr>
<tr>
<td>Maximum path delay</td>
<td>10.991ns</td>
<td>10.290ns</td>
</tr>
<tr>
<td>Total On-Chip Power</td>
<td>59.79mW</td>
<td>60.00mW</td>
</tr>
</tbody>
</table>

Figure 17: CRC comparison with SRAM and Dual Port Memory
The aim of this work is to mainly concentrate on Error Correction Coding techniques like Hamming codes, Convolution codes, Cyclic Redundancy Check and turbo codes. These codes are designed with Verilog language and simulated on SPARTAN-3 using XILINX ISE 14.2 and Chipscope Pro.

Though it is very difficult to tell which the best technique is, it is better to run different techniques in parallel and take up the reliability and quality of different techniques to get best throughput.

REFERENCES


