ERS 220

Practical report 2:

Hamming code

Group 35

Ausiku M.S. [redacted]

F. Botha 29042951

Omar M. W. [redacted]

Taghizadeh S. [redacted]

August 2011

UNIVERSITY OF PRETORIA

FACULTY OF _____

DEPARTMENT OF _____

The Department of _____, places great emphasis upon integrity and ethical conduct in the preparation of all written work submitted for academic evaluation.

While academic staff teach you about systems of referring and how to avoid plagiarism, you too have a responsibility in this regard. If you are at any stage uncertain as to what is required, you should speak to your lecturer before any written work is submitted.

You are guilty of plagiarism if you copy something from a book, article or website without acknowledging the source and pass it off as your own. In effect you are stealing something that belongs to someone else. This is not only the case when you copy work word-by-word (verbatim), but also when you submit someone else's work in a slightly altered form (paraphrase) or use a line of argument without acknowledging it. You are not allowed to use another student's past written work. You are also not allowed to let anybody copy your work with the intention of passing if of as his/her work.

Students who commit plagiarism will lose all credits obtained in the plagarised work. The matter may also be referred to the Disciplinary Committee (Students) for a ruling. Plagiarism is regarded as a serious contravention of the University's rules and can lead to expulsion from the University.

The declaration which follows must be appended to all written work submitted while you are a student of the Department of _____. No written work will be accepted unless the declaration has been completed and attached.

I (full names) _____

Student number _____

Topic of work _____

Declaration

1. I understand what plagiarism is and am aware of the University's policy in this regard.
2. I declare that this _____ (e.g. essay, report, project, assignment, dissertation, thesis etc) is my own original work. Where other people's work has been used (either from a printed source, internet or any other source), this has been properly acknowledged and referenced in accordance with departmental requirements.
3. I have not used another student's past written work to hand in as my own.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Signature _____

Plagiarism

Practical Guidelines

These guidelines must be read and followed by all students. It is expected that a copy of these guidelines be signed by each student and attached to the first practical report. Failure to do so will result in a zero for your practical and report.

1. Please remember this is a DEMO session i.e. your practical must be completed before the assigned time. Do NOT arrive at the session expecting three hours to work on your practical.
2. Every group member MUST have a lab book. Sharing is not permitted i.e. every group must have the respective number of completed lab books. Pro rata of the total for the practical will be deducted for every lab book not present.
3. If you are late for your demo session, as allocated, you will receive zero for the practical. This is not negotiable. If one of your group members fails to arrive they will receive zero, unless for medical reasons or prior arrangements have been made with the lecturer.
4. If your practical does not function correctly during the session you may be allowed a re-demo. This will occur during the next week's practical and will only be given if your group can demonstrate that work was completed on the practical prior to the session and all of your lab books are complete.
5. Your practical report (1 per group) must be handed to your demonstrator at the completion of your demonstration. It must have a cover page, stating your group number clearly, table of contents, plagiarism statement signed by ALL group members, group members who do not sign will receive zero for the report and failure to include a statement will result in the group receiving zero, a signed copy of these guidelines and all the headings listed in the study guide. Failure to hand in a report during the session will result in a zero for the report.
6. When you arrive at the practical session write your chosen bench number under your group number. As you have chosen the bench you will be expected to be at the bench given. Demonstrators will not wonder around the labs trying to find you. If you have not demonstrated your practical during your allocated time you will receive zero. This is not negotiable. If you fail to indicate your bench number your practical will not be marked and you will receive zero.
7. Arrive at least 15-20 minutes before your allocated time to ensure your practical's functionality.
8. Students have not been assigned to benches. As such if you choose a non-functioning bench, change benches. Non-functioning benches will NOT be accepted as an excuse for non-functioning practicals.
9. Each group must complete their own practical. If it is determined that more than one group has demonstrated the same circuit the groups that colluded will receive zero and NOT given the option to re-demo.
10. Students must buy their own components. As such not having correct components is NOT a valid reason to not complete a practical.

Declaration:
I have read and understood these guidelines and will follow them during all practical sessions.

Name:_____ Signature:_____

Name:_____ Signature:_____

Name:_____ Signature:_____

Name:_____ Signature:_____

Name:_____ Signature:_____

Summary

In practical situations errors are often get introduced into code words that get transmitted from one point to another. Thus the need arises for these errors to be detected and corrected. Hamming codes are an effective way to add a error checking and correcting capacity to circuits without adding large amounts of redundancy, both to code words and circuits. This practical explores the circuitry used to detect errors in a distance-4 8-bit hamming code, by means of synthesizing a circuit to detect 1 bit errors and their positions, as well as detecting two bit errors. It also illustrates the usefulness of combinational logic to decode confusing code words into simple and user friendly form.

Table of contents

Introduction.....6
Problem Statement 6
Problem Formulation 6
Background 6

Design and Implementation.....7
Approach 7
Detailed Design and Implementation 7

Results and Discussion of Results.....8

Conclusion.....12

References.....12

Table of figures

Figure 1 Parity matrix_____7
Figure 2 Parity check circuit_____8
Figure 3 Karnaugh maps (1-cells or 0-cels indicated in blue)_____10
Figure 4 Decoding circuit_____11

Table of tables

Table 1 Truth associating syndrome vector with actual meanings and BCD code.....9

Introduction

Problem Statement

During transmission of digital data from one component to another through some type of transmission media code words may become corrupted.This in turn causes errors in the output of a digital system. The challenge is to improve the integrity of transmitted digital data and to find a means to verify received data. In real time applications such a voice or video communication knowing that there is an error in the transmitted data is not simply enough as there is no time to request a resend. In these situations it is useful to have a system for correcting received data without referring back to the originally transmitted data.

Problem Formulation

To investigate how the system of hamming codes can be used to detect errors in a received code word as well as the ability to correct erroneous code words.

Background

Hamming codes are a general method for constructing codes with a minimum distance of 3 or higher. Non code word words can be detected using parity groups. If we change one bit of a code word, in position j, then we change the parity of every group that contains the position j. since every information bit is contained in at least one group, at least one group has incorrect parity and the result is a non code word.

To design an error-correcting decoder, we first check all of the parity groups, if all have even parity then the received word contain no error. If at least one group has odd parity, then a single error has occurred. The pattern of groups that contain odd parity must match one of the columns in the parity check matrix; the corresponding bit position is assumed to contain the wrong value and is then complemented to correct the word. The required number of parity bits grows slowly with the width of the memory word.

Hamming codes offer a great advantage over other methods of error checking, namely that as the number of bits transmitted increase the number of parity bits remain relatively few. It also has advantages for systems where real-time action is required and there is not time for re-requesting data with errors. This is also beneficial in systems other than real-time by reducing errors and freeing up bus usage.

(Reddy, 2011)(Wakerly, 2006)

Design and Implementation

Approach

In this Practical a combinational logic design is used to decode a received 8-bit minimum-distanc-4 hamming code containing four information bits. The result is then displayed on a 7-segment display indicating no error detected, a two bit error detected, or a one bit error detected and the position of the error.

Detailed Design and Implementation

We designed a hamming code error detector for a distance-4 8-bit hamming code defined by the following parity matrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Figure 1 Parity matrix

The first step in decoding a hamming codeword is checking the parity of four groups of bits in the code word. This is physically realized using xor gates. If the parity of a group is even the check will produce a “0” value and if the parity is odd it will produce a “1” value. Odd parity is associated with an error. The four results for the parity checks are the combined into a 4 bit codeword known as a syndrome vector. Syndrome vector code words are associated with particular errors as shown in table 1. This code word is decoded using combinational logic that is synthesized using truth tables, Karnaugh maps, and Boolean algebraic simplification. There are two approaches decoding the syndrome vector. One approach is to decode the syndrome vector into a 1-on code used to drive an array of LED’s. This indicates the position of the error in the code word by illuminating the corresponding LED, or a LED indicating no error, or a LED indicating a two bit error. Another approach is to decode the syndrome vector into a four bit BCD code. This code is then used as the input to a 7-segment display driver. This will display the position of a one bit error in a more user friendly way on a seven segment display. The latter approach using a 7-segment display was used.

Results and Discussion of Results

To check the parity of the code word input (according to fig. 1) the following circuit was designed with xor gates.

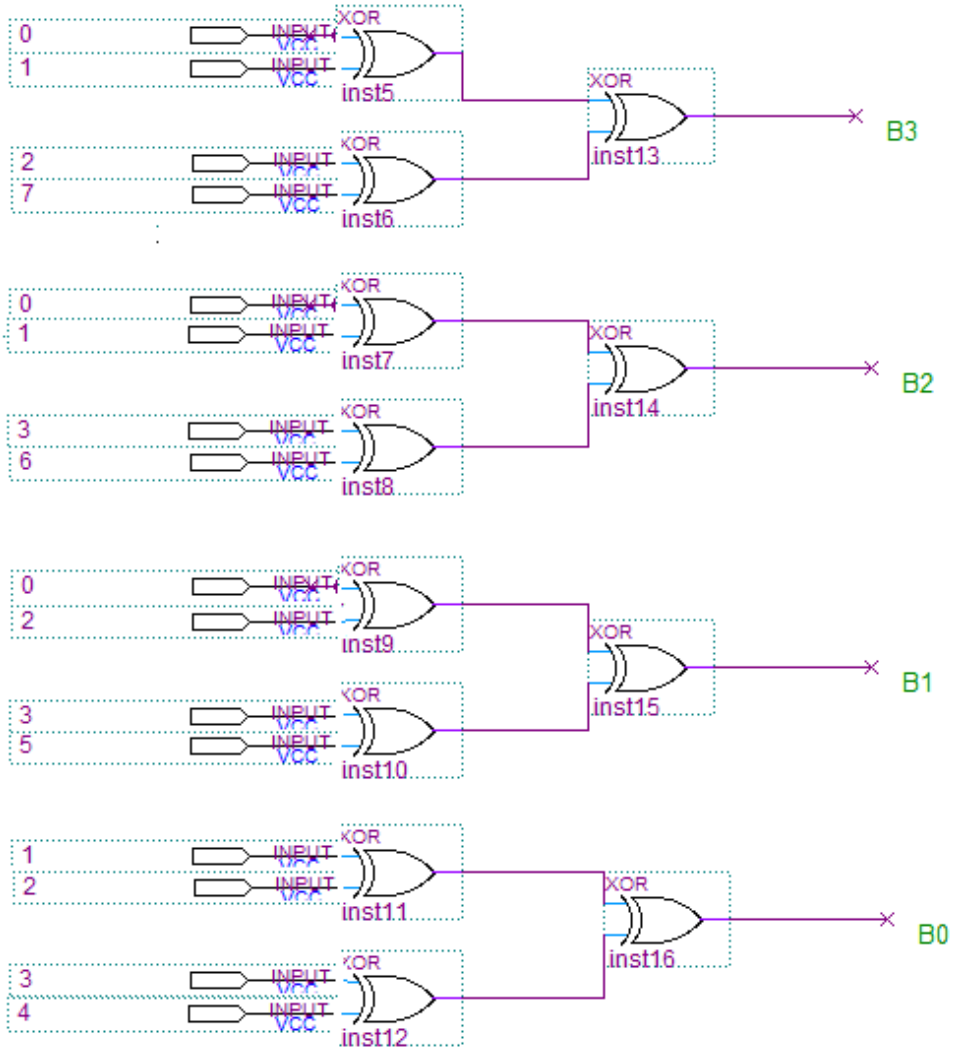


Figure 2 Parity check circuit

Careful inspection of the output of the above parity check circuit and referral to a table of known code words can be used to determine the meaning of the syndrome vector as shown in the center grouping of table 1. These meanings are then associated with the desired BCD codes that can be used by a driver circuit to display the results on a 7-segment display.

Decoder (syndrome to BCD code for 7-segment display)										
	Bit 3 (A)	Bit 2 (B)	Bit 1 (C)	Bit 0 (D)	No. of errors	position	Code 3	Code 2	Code 1	Code 0
0	0	0	0	0	0	-	1	0	0	1
1	0	0	0	1	1	4	0	1	0	0
2	0	0	1	0	1	5	0	1	0	1
3	0	0	1	1	2	-	1	0	0	0
4	0	1	0	0	1	6	0	1	1	0
5	0	1	0	1	2	-	1	0	0	0
6	0	1	1	0	2	-	1	0	0	0
7	0	1	1	1	1	3	0	0	1	1
8	1	0	0	0	1	7	0	1	1	1
9	1	0	0	1	2	-	1	0	0	0
10	1	0	1	0	2	-	1	0	0	0
11	1	0	1	1	1	2	0	0	1	0
12	1	1	0	0	2	-	1	0	0	0
13	1	1	0	1	1	1	0	0	0	1
14	1	1	1	0	1	0	0	0	0	0
15	1	1	1	1	2	-	1	0	0	0

Table 1 Truth associating syndrome vector with actual meanings and BCD code.

The circuit shown in figure 4 was synthesized using table 1, the Karnaugh maps shown in figure 3. The circuit is designed to use only AND, OR and NOT gates the following logic equations where derived from the Karnaugh maps:

Code 3 = A'B'C'D' + ABC'D' + A'BC'D + AB'C'D + A'B'CD + ABCD + A'BCD'

+ A'BCD' + AB'CD'

= A'B'(C'D + CD') + AB(C'D' + CD) + A'B(C'D+CD') + AB'(C'D+CD')

= (A'B' + AB)(C'D' + CD) + (A'B + AB')(C'D + CD')

Code 2 = (A + B + C + D)(D' + B')(A' + D')(B' + C')(A' + C')(C' + D')(A' + B')

Code 1 = (A + B)(A' + B')(C + D')(C' + D)

Code 0 = (A' + C')(D' + B)(B' + D)(A + B' + C)

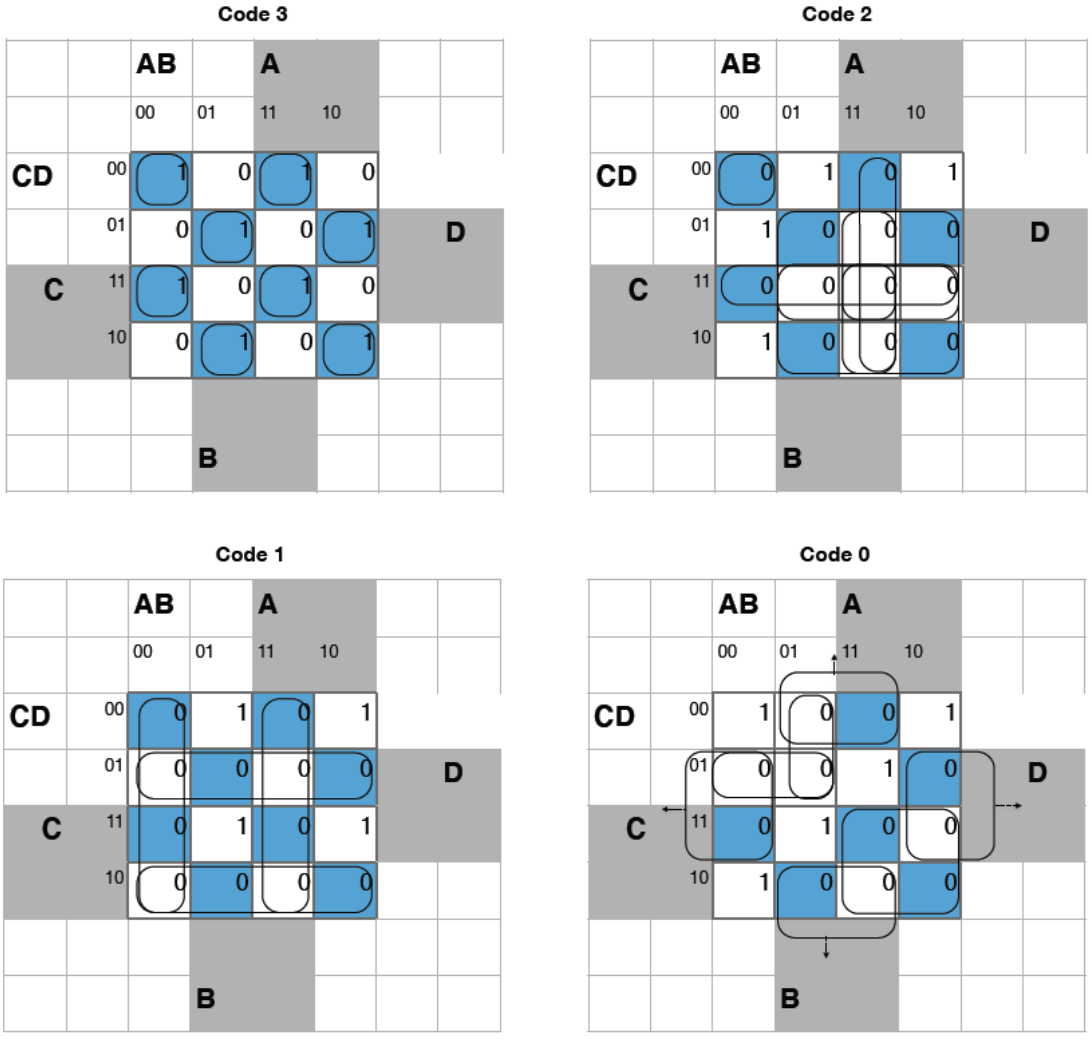


Figure 3 Karnaugh maps (1-cells or 0-cels indicated in blue)

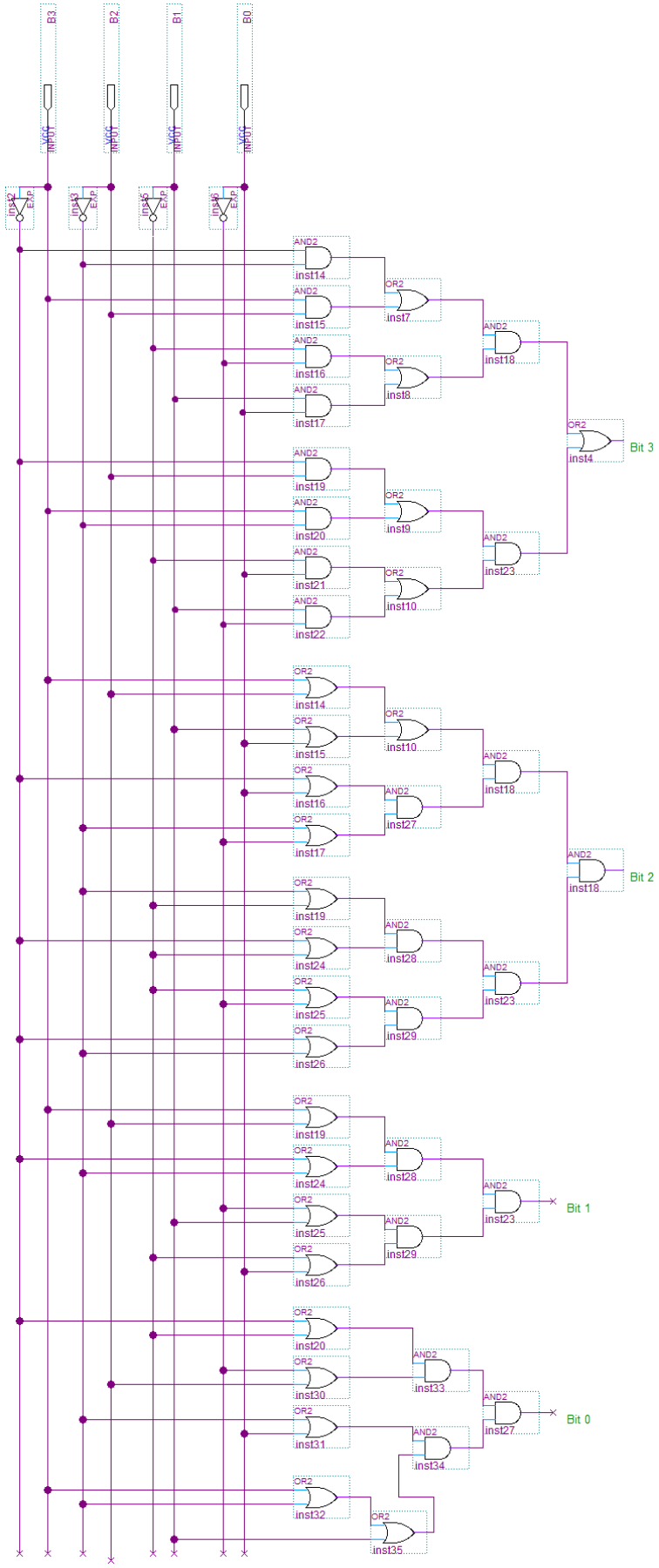


Figure 4 Decoding circuit

Conclusion

This practical clearly illustrates how hamming codes can be used to detect errors. The concept of parity groups and checking parity is truly useful in adding error checking and recovery of lost data without adding large amounts of redundancy to circuits and code words. It also illustrates how combinational logic can be used to extract meaningful information from a seemingly random code, i.e. decode data. This greatly aids user friendliness.

References

1. Reddy, S. 2011. *Study Manual, Digital Systems.* University of Pretoria
2. Wakerley, J.F. 2006. *Digital Design, Principles and practices* Pearson, Pretence Hall