

Derek Molloy Modules Mis à jour le 25 mai 2020 à 07:28

UTILISER CE MODÈLE

Derek Molloy Modules

EE223 - Digital and Analogue Electronics

Search this site

389

days since
Start of Module[Main Page](#)[Course Notes](#)[Examinations](#)[Experiments](#)[1. A Simple Power Supply Circuit](#)[2. Introduction to Logic Gates](#)[3. Logic Minimisation](#)[4. Encoders and Decoders \(in progress\)](#)[5. Clock Generator Circuits](#)[6. Flip-Flops \(in progress\)](#)[7. Counters and 7-Segment Displays](#)[8. Introduction to the Arduino \(In Progress\)](#)[9. LED Dot Matrix Display \(In Progress\)](#)[Appendix: Kit Components List](#)[Laboratories](#)[Laboratory Manual](#)[Laboratory Session 1 - Combinational Logic](#)[Laboratory Session 2 - The Diode and Adders](#)[Laboratory Session 3 - The Transistor](#)[Experiments](#) >

3. Logic Minimisation

Description:

In this experiment we are going to design a logic system to solve a particular problem. To do this we are going to use the formalised method of using Karnaugh Maps to minimize the number of gates that are required for the implementation of the solution.

Note: The full set of videos is presented on the YouTube channel: [DerekMolloyDCU](#)

Equipment Required:

- Breadboard - with Voltage Regulator setup from 1.
- AND gates (7408),
- OR gates (7432),
- NAND gates (7400),
- Suitable resistors (pull-up/pull-down),
- LED with suitable resistor.

Table of Contents:

Contents

[1 Description:](#)[2 Equipment Required:](#)[3 Table of Contents:](#)[4 Introduction](#)[5 Problem for you to solve:](#)[6 Worked Solution:](#)[7 Conclusions](#)

Introduction

We often need to determine the minimum circuitry for a particular Boolean expression in order to reduce the cost and surface area of the circuit o

[Traduire](#)

[Laboratory Session
4 - BCD Counters &
Sequential Logic](#)
[Laboratory Session
5 - Arduino
ADC/DAC
Assignment](#)

[Tutorial Materials](#)

[Useful Links](#)

[Archive](#)
[Announcements](#)

[Sitemap](#)

3. Logic Minimisation - EE223 - Digital and Analogue Electronics

improve the performance of our circuit. Karnaugh maps provide an alternative solution to Boolean algebra in determining the minimum circuitry required to implement a logic circuit. Boolean algebra can be very tedious and it can be difficult to determine a consistent solution to a minimisation problem. Karnaugh maps on the other hand provide a formalised method for solving such a problem and, if the rules are followed correctly, it will result in a consistent solution. Karnaugh maps work well for up to 5/6 variable, but after that they too become unwieldy and a software assisted solution is required.

There is a very good discussion on Karnaugh maps in the notes: [Course Notes](#)

Problem for you to solve:

The problem that we wish to solve is as follows:

A system has 4 momentary push buttons (A, B, C, D). We wish to light a single LED when the following conditions arise:

- Any 3 buttons are pressed at the same time.
- All 4 buttons are pressed at the same time.
- The outer, or inner pair of buttons are pressed (i.e. AD or BC)

In all other circumstances the LED should be off.

You should produce a minimum logical expression and **implement your solution using a mixture of AND/OR gates**. Also, you should provide a solution **using only NAND gates**.

Worked Solution:

Please try to solve this problem yourself and if you have problems then work through the videos below. The solution to this problem is provided in two parts, which is due to duration constraints with YouTube videos.

Experiments 2.5: Digital Logic - Karnaugh ...



Part 2 Continued (YouTube limits video length).

Experiments 2.5: Digital Logic - Karnaugh ...



Conclusions

Hopefully this will give you a good idea of Karnaugh maps and how we can use them to solve logic minimisation problems. Please read the course notes to ensure that you have a good feel for all of the rules for minimisation.

Commentaires

Vous n'êtes pas autorisé à ajouter des commentaires.

[Se connecter](#) | [Activités récentes sur le site](#) | [Signaler un abus](#) | [Imprimer la page](#) | Avec la technologie de **[Google Sites](#)**