

Minterm Minimizer

Quine Mc-Cluskey method

Ahmed Hussein – Bassam Aiman | Digital Fundamentals | 16\5\2017

Overview

**This project is based upon Quine Mc-Cluskey’s and Petrik’s methods of simplifying minterms of a given function.**

**It’s implemented in both Windows and Android operating systems.**

**Using the concepts of object-oriented programming and Java programming language, we’ve managed to deliver an application that functions both efficiently and with the simplest of algorithms.**

Main modules and their functions

The project is divided in two stages:

**Stage 1:**

The user input the minterms and the Don’t Cares in the GUI then click calculate

Our rule here is to take the two Strings then return list of essential prime implicants

**Main functions:**

**getPrimeImplicants** : it takes the String from the user and return a list contains the integers which are the required implicants

**getFirstColumnOfMintermArray** : it takes the implicants and start the process by crossing the elements together to generate the first crossed columns

**getColumnsArrray:** it takes the result from previous function to generate the rest columns and continue crossing elements together till the remaining are all essential

**getSortedImplicants:** it is the final step where it returns the implicants sorted in the list to be ready for the next stage

**Stage 2:**

Here we take the result from stage 1 sorted and minterms entered by the user but in their integer form then we use petrick method directly to obtain the result.

The algorithm here as follows:

**FormExpandedMinterms:** you are given here the implicant in the form 2(1,4) and return a list which contains the expansion in the form 2 , 3 , 6 ,7

**MintermCover:** here u are given the minterms and array of list of implicant (the position of x's in the table and return a list with the index)

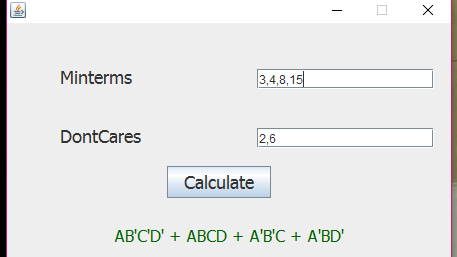
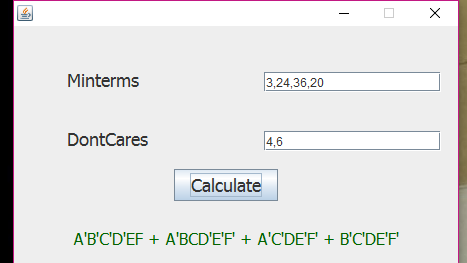
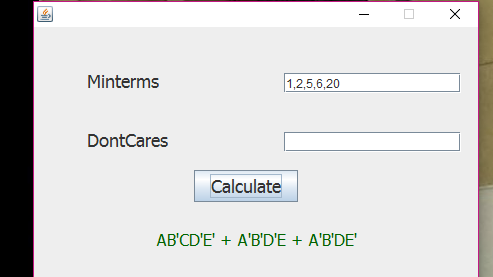
**StartPetrick:** here it multiply out the implicants which are covered in the table and return the result in binary where example : 3 means 0011 means 1st implicant and 2nd one 5 means 0101 means 1st and 3rd implicants 4 means 0100 means only the 3rd implicant.

**MinCost:** here it compare the result of Petrick’s method and return the minimal cost as an integer that is the indices of the implicants.

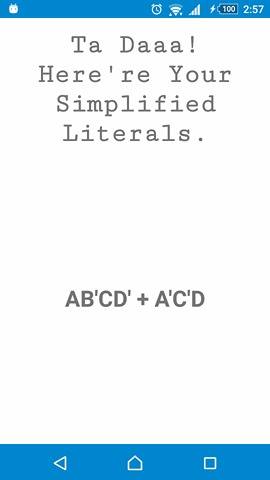
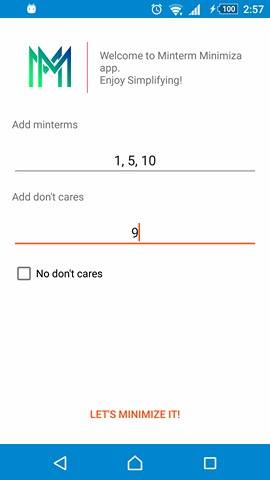
**Display:** here u give this function the result from petrick in binary and it display it in the form of literals.

This the final Step which appears in the GUI in the label called Result.

Screenshots and Sample RUns



Here are some ScreenShots from our Mobile application:



User Guide

There are only two textboxes only you have to enter the minterms in the first one and the Don’t Cares in the second one and Press Calculate.

The result will be immediately appear in the result label.