Digital Design Project I

Amena Hassan and Ali Yassine

25 March 2022

1 Introduction

The project is designed to work on 2 or 3 variables Kmaps. Thus, it first asks about the number of variables and limits the valid options to 2 and 3 using a while loop.
After that, the project is designed to ask for number of minterms. In this stage, regardless the number of variables, if the number of mintersm is 0, then the expression is "F=0" and if the number of variables is 4,in case of 2 variables, or 8, in case of 3 variables, the expression is "F=1".
Also, if the number of variables is 2, the user can only enter a maxium of 4 minterms and if the number of variables is 3, the user can enter a maximum of 8 minterms.
The third stage of inputting the data is entering the minterms.

2 Structures

We implemented 3 structures:
struct minterm: This is a struct that saves the binary representation, decimal value, number of instances, and boolean value (tick) that will help classify prime and essential prime implicants later.
struct pairs: The same as minterm struct but for each matched pairs of minterms.
struct pairs2: The same as minterm and pairs structs, but for two matched pairs (four grouped minterms).

3 Functions

3.1 Kmap2()-Kmap3()

Data Structures:

1. vector<vector<int>>kmap-kmap2.
   These are vector of vectors, (such as 2D array), that save the values of each minterm, if it is 0 or 1.

In this function, we loop over the vector of minterms and specifically the decimal value of each minterm. If this decimal value of the minterm is equal to the iterator, then we change the value of the Kmap cell to 1.
At the end, using a for loop we display the Kmap.

3.2 Expression2

This is a function used for 2 variable Kmap. It calculates all possibilities of the 2 variable Kmap and assigns an expression using if statements.

3.3 GroupingOnes

Data Structures

1. vector<vector<minterms>>group(4);

This function loops over the vector of minterms, and specifically the binary form of each minterm. There is a counter that ww declared. This counter is incremented according to number of ones in a binary represenation. After that, this minterm is added in the vector of vector "group" according to how many ones it contains.

3.4 Grouping

Data Structures

1. vector<vector<pairs>>matchingpairs(3);
2. vector<minterm>primeImps;
3. map<string, vector<int>>EPIs;
4. vector<vector<pairs2>¿ matching4s(2);

This function loops over group vector, which groups minterms based on number of ones in them, and compares each minterm in each two consecutive groups and counts the number of differences in their binary forms. If number of differences is equal to one, then these two minterms are considered a matched pair and are added into vector matchingpairs. Also, during the loops, we keep track of which index is different (in case of one difference) and replace the character at this index with $-$, and then set the binary representation of the two matching pairs with the new string. Then, we e set tick equal to true for each two paired minterms.

Next, we loop over all the minterms and add the binary and decimal values of those which have tick value of false, meaning that they were not matched, to EPIs, which is a map storing all essential prime implicants.

After that, we will loop over the binary representation of every pair and we compare this element with all the pairs of the next group (comparing binary representations). We have a counter that indicates how many different digits

between both binary representations.
For the cases, where the counter is 1, thus there is only one different digit in binary representation of both, we add both pairs into a vector of vectors.
Then we loop over each pair and check the Boolean value of "tick", and add the pairs of false value into the map of Essential prime Implicants "EPIS".

3.5 Expression

Data Structures:

1. vector<string> Essential: Vector of strings to save the binary representation of essential prime implicants with instances 1.

We loop over the decimal representations of the minterms and over the EPIs and then if the decimal representation of the minterms is present in the set of 4 decimals in the EPIs then increment the instance of this minterm.
After that, we take the minterms of instances 1 and then we check if the binary representation of this minterm is in the vector of strings "Essentials", if not then push it to the vector of strings.

Finally, we loop over Essential vector and produce the letter representation of each element (string of binary) in it. Then we print the final minimized expression.