TWO LEVEL LOGIC GATES MINIMIZATION

ECE1733 Topics in Switching Theory

Zeming Li (1001654845)
Yuchen Wang （1002821574）

ECE1733 Topics in Switching Theory

Zeming Li (1001654845)
Yuchen Wang (1002821575)

Abstract

This assignment is intended for students to get familiar with two level logic gate minimization and write a program to implementation of an optimization method. Two methods were introduced in class: one is a tabular method, a.k.a. Quine-Mccluskey method and a cubical method. Method II are used in this assignment and tested on given test nodes.

Implementation

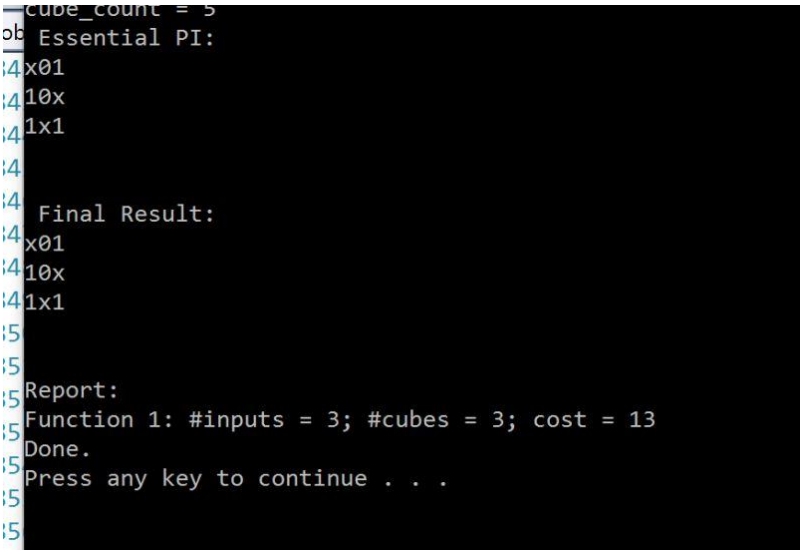
This program was written in C language with Microsoft Visual studio. An executable file (assign1.exe) was generated and can be directly run in command prompt.

a) Download the file ***assign1*** to a folder of choice and put test node files (.blif) under the same directory.
b) Rename file name to ***assign1.exe***
c) Open command prompt and use change directory command (cd) to the designation folder.
d) Input command assign1.exe <source BLIF file> (e.g. assign1.exe node1.blif) to execute assign1 with input data in node1.blif

The command prompt will run assign1.exe and the result will be printed in the same command window screen.

Test Results

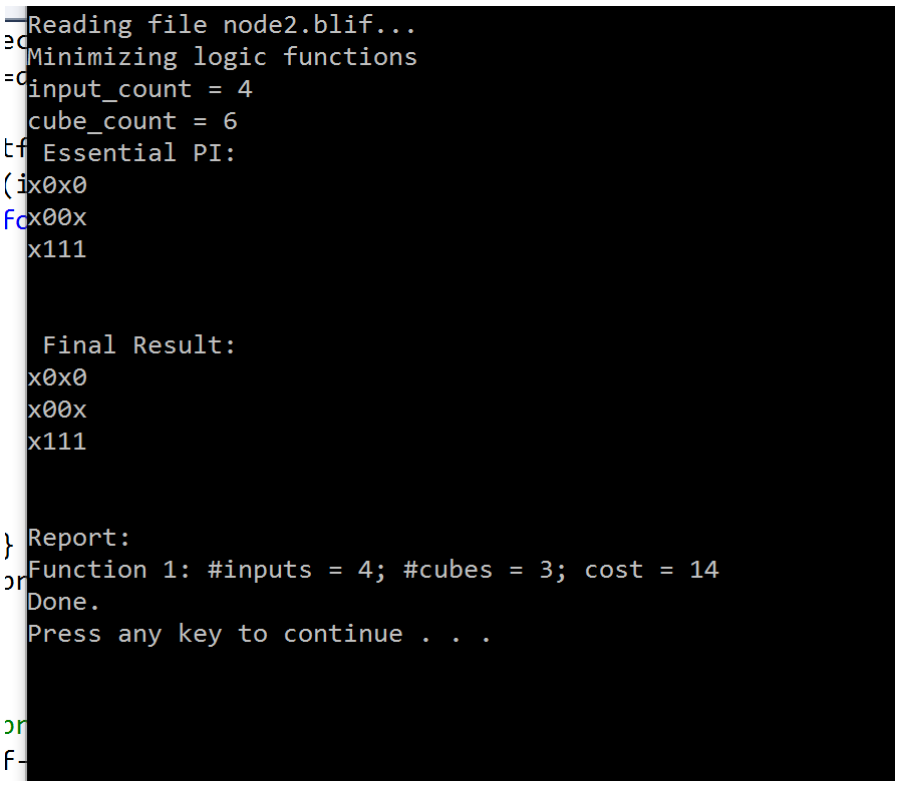
- **Test Node 1**



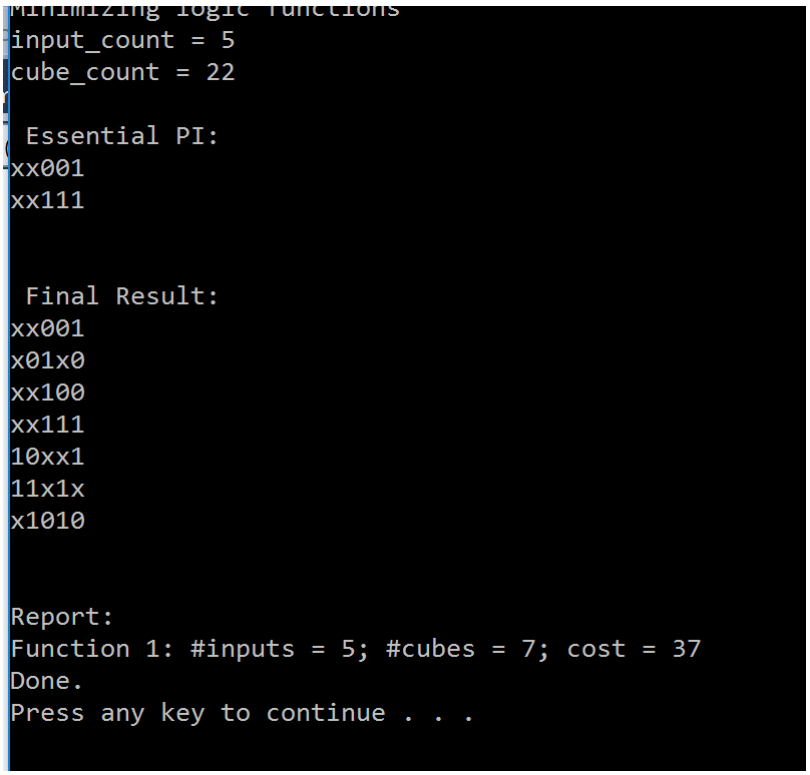
ECE1733 Topics in Switching Theory

Zeming Li (1001654845)
Yuchen Wang (1002821575)

- **Test Node 2**



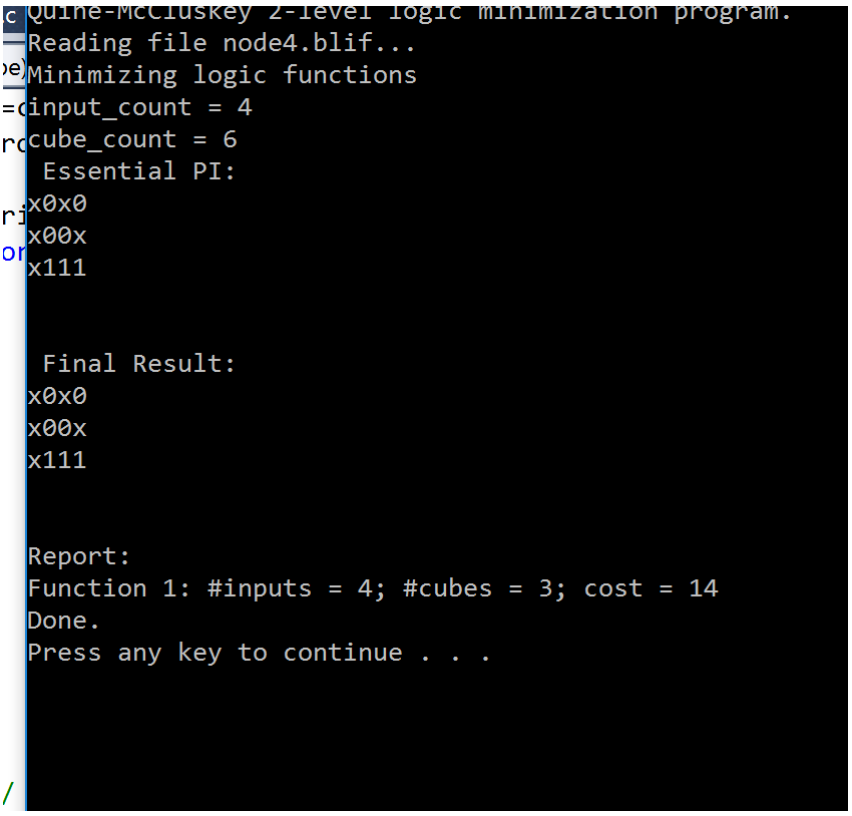
- **Test Node 3**



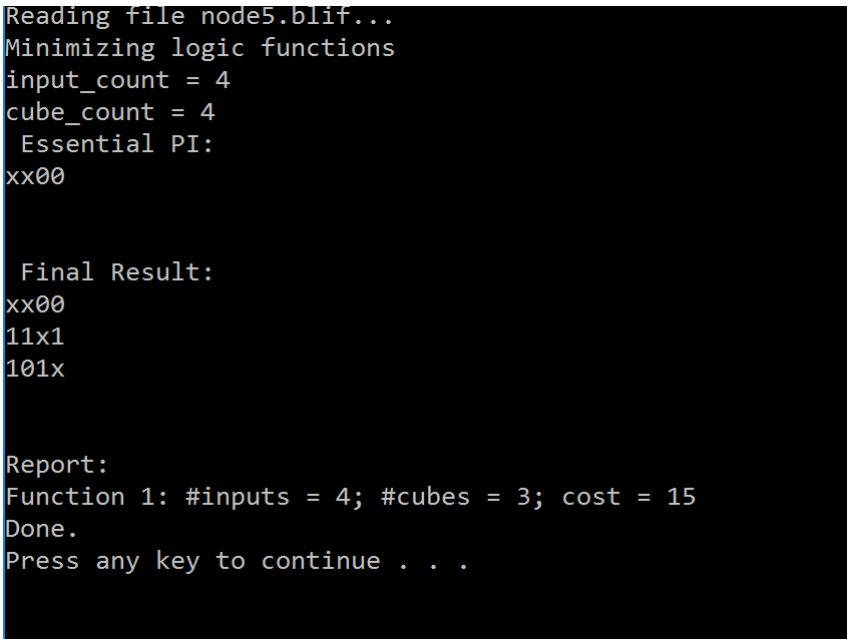
ECE1733 Topics in Switching Theory

Zeming Li (1001654845)
Yuchen Wang (1002821575)

- **Test Node 4**



- **Test Node 5**



ECE1733 Topics in Switching Theory

Zeming Li (1001654845)
Yuchen Wang (1002821575)

Discussion

- **Minimization Algorithm**

This program has followed the standard algorithm for the cubical method:

a) Use star operation to generate cube cover. Use sharp operation to find redundant cube and delete from cover set. Repeat process till cover C_n and C_{n-1} are the same.

b) Before producing essential PIs, all PI in the resulting array was checked to see if it covers only don't care (DC) minterms by checking if at least one minterm in the ON set is covered by this PI.

c) In stage 3, for checking all the minterms of the function are covered, a series of #_operations is used in this stage. As long as one #_operation result is null, the minterm is fully covered by PIs. As for other situations, the results of a series of #_operations, which are the remaining uncovered parts, are stored in an array and available in next stage.

d) In stage 4, a sort algorithm is used before branch & bound. The elements in non-essential PIs set are sorted by the number of DC they have, which means non-essential PI with more DC stores before other PIs in the array. When non-essential PIs are selected to cover the remaining minterms, those have more DC and less cost will be selected in priority. It results in less execution time and cost.

- **Program Features**

The highlight of this program is simplicity. Cubical method is generally faster and simpler than the QM method. Total program line number is within 1000 lines. This is even much smaller comparing with other groups. More, we found other group were experiencing a lot of memory crashes and need to reallocate memory after each operation. We have kept results (result[][]) in one location. All operation was repeatedly overwritten on the same array while keeping the total rows of the result array as an index pointer.

Most of the operation were directly written in the main simplify function rather than calling different functions. However, the program file is written in sections and well commented; hence it's very clear for others to read and for team partner to work on the same program. All internal results were checked and printed to keep track of the process.

- **Limitation**

The number of signal in the cube is pre-defined. Hence some arrays are not be reallocated based on the size of output, so it is a waste of store space. Some arrays only can be used when f->input_count less and equal to five, because they are allocated based on assumption of no more than five elements in each cube. If more signals are in the cube, the system may be not executable as a result of memory crash.