**Digital Design Project**

We divided the project into SOP and POS. Hamza took the POS and Mohamed took the SOP.

How to build the program:

Take the c++ file in the DIGITALDESIGNPROJONE file and open it in your compiler.

How our program works:

A program that does the following: -Takes input as POS or SOP. -Does validation on the format. "abc + ad'c" or "(a + b + c)(c + d + e')". SoPValidation1() -Takes any variables (Doesn't need to be from the alphabet). -Validates that the number of variables is not more than 10 --> SoPValidation2() and PoSValidation(). -Takes the input string and divides it into parts using the SoPCleanUp() or PoSCleanUp() functions and stores it in vectors. -Takes the vectors and turns the strings into binary strings. SoPtoBinaryString() or PoStoBinaryString(). -Takes the binary string to the TTableBuild() or TTableBuild2() functions to build the truth table. -After the table is built, the final column is filled according to the function provided. FTTColFillSOP() or FTTColFillPOS() -The PrintMinMaxterms() function is then used to derive the canonical min and max terms in boolean form. -It then calls the SoPtoBinaryString2() or PoStoBinaryString2() to turn the boolean terms into their binary string form and then into their integer form. -The SoPtoBinaryString() is then used to convert the canonical minterms into binary strings. -The binary strings are then given to the GroupByOnes() to group them according to the number of ones. -The ImplicantGroupComparison() function derives the prime implicants and the essential prime implicants by comparing each group with the ones underneath it. The process is repeated until all essential prime implicants are derived.

Problems faced:

One of the problems that faced us in the beginning was deciding how to deal with each character In the string individually. So, we decided to divide the string into vectors and turn the strings into binary strings. The general problem we faced was that we knew how to solve the problems physically, but not through coding.

Problems with the code:

The validation is limited to checking if there are brackets in the SoP or missing brackets in the PoS. It also checks if there is less than 10 variables. We also didn’t have time to check if the PoS parts work or not so we derive the implicants using SoP only. Additionally, the complexity is quite high as there are 5 for loops in one function.