CAD Design Project 4 – Primitive Boolean Operations with *Espresso*
Due: 23:55, Nov. 9, 2022

The *Espresso* is a software package developed at IBM for two-level Boolean function minimization in 1982. The *Espresso* is then integrated into the most famous logic synthesis engines, *SIS* and *ABC*, at UC Berkeley in 1992 and 2005, respectively. The *Espresso* adopts positional cube notation – a binary encoding technique to process Boolean cubes, a.k.a. implicants and product terms. Combined with the unate recursive paradigm (URP), we are capable of implementing primitive Boolean operations efficiently for 2-level and multilevel logic synthesis. In this project, you are required to use *Espresso* as a basis to implement three primitive Boolean operations, ***AND***, ***OR***, and ***XOR*** according to the following requirements:

1. Read a PLA file and two numbers of corresponding output functions, A and B.
2. Perform exact Boolean minimization (minimum product terms) for functions A and B.
3. Output the exact minimized PLA files, a.pla and b.pla. (NOTE: There is no don't cares.)
4. Perform 3 Boolean operations, AND, OR, and XOR, for functions A and B, respectively.
5. Perform exact Boolean minimization for the resultant functions.
6. Output the exact minimized resultant PLA files, and.pla, or.pla, and xor.pla.
7. Upload your source code tarball (*.tgz) to moodle (including your Makefile). (NOTE: The uploaded file name should be the same with your student ID.)

```
PLA Example: sample.pla

.i 4
.o 3
000- 1-1
001- 110
0100 0-1
0101 -0-
011- -1-
1000 1-1
1001 100
1010 1--
1011 101
1100 111
1110 --0
1111 10-
.e
```

SYNOPSIS

```
%> boolean PLA_FILE FUNCTION_A_NUM FUNCTION_B_NUM
```

Run-time Example:

```
%> boolean sample.pla 1 3


%> cat a.pla
.i 4
.o 1
.p 3
-0-- 1
--1- 1
1--0 1
.e


%> cat b.pla
.i 4
.o 1
.p 3
0-0- 1
--00 1
1-11 1
.e


%> cat and.pla
.i 4
.o 1
.p 3
000- 1
1-11 1
1-00 1
.e


%> cat or.pla
.i 4
.o 1
.p 4
0--- 1
-0-- 1
--1- 1
---0 1
.e


%> cat xor.pla
.i 4
.o 1
.p 4
01-- 1
0-1- 1
--10 1
1001 1
.e
```