CAD Design Project 3 – Multi-level Boolean Network Generation
Due: 23:59, Nov. 22, 2023

Nowadays, most combinational circuits are implemented as multi-level Boolean networks. The fine granularity of logic gates provides circuit designers with a wide variety of choices for power, area, and performance optimizations. However, the design flexibility of multi-level logic synthesis incurs large computational complexity. Only a few restricted methods are formulated for multi-level Boolean optimization. In this project, you are required to implement a heuristic for algebraic multi-level Boolean network generation with a minimal number of literals. By applying *algebraic division*, *decomposition*, *kernel extraction*, and *substitution* operations, output a multi-level Boolean network in BLIF format according to the following requirements:

1. Read a two-level BLIF file. (The file can be either single output or multiple outputs.)
2. Perform algebraic multi-level logic optimization to minimize the total literal count.
3. Output a multi-level Boolean network in BLIF format. (file name: out.blif)
4. Upload your source code tarball (*.tgz) with your Makefile to portal.
   (NOTE: The uploaded file name should be the same with your portal account.)

```
SYNOPSIS
%> multilevel BLIF_FILE

BLIF Example: sample.blif

.model sample
.inputs a b c d e f g h i j k l m n
.outputs o p
.names a b c d e g h i j k o
01111---1- 1
01011--11- 1
010111--0- 1
11111---11 1
11011-111- 1
.names c d e f g h l m n p
1111---1- 1
-001--0-0 1
101111--0 1
.end
```

```
Run-time Example:
%> multilevel sample.blif
Original literal count: 52
Optimized literal count: 34

%> cat out.blif
.model sample
.inputs a b c d e f g h i j k l m n
.outputs o p
.names b d e v o
1111 1
.names f y p
11 1
.names a c g j w v
---11 1
0010- 1
.names a c h i k x w
-1--1- 1
0----1 1
-011-- 1
.names c i x
1- 1
-1 1
.names c d e l n z y
1-1--1 1
-0000- 1
.names d g h m n z
1--1- 1
011-0 1
.end
```