# Logic Synthesis & Verification, Fall 2023

## Programming Assignment 1

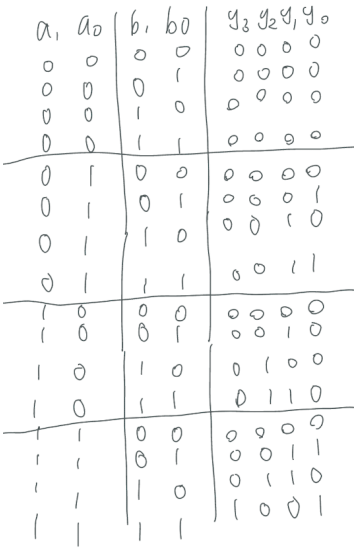
r12921045 電機碩一劉虹伶

## 2. [Using ABC]

(a) reate a BLIF file named “mul.blif”

LSV > LSV-PA > mul.blif

```
1   .model mul
2   .inputs a1 a0 b1 b0
3   .outputs y3 y2 y1 y0
4   .names a1 a0 b1 b0 y3
5   1111 1
6   .names a1 a0 b1 b0 y2
7   101- 1
8   1110 1
9   .names a1 a0 b1 b0 y1
10  011- 1
11  10-1 1
12  1101 1
13  1110 1
14  .names a1 a0 b1 b0 y0
15  01-1 1
16  11-1 1
17  .end
```




(b) Perform the following steps to practice using ABC with the two-bit unsigned multiplier example.

1. read the BLIF file into ABC (command “read”)

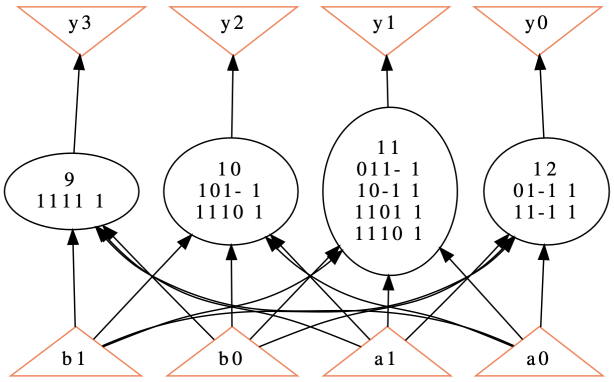
```
abc 01> read mul.blif
```

2. check statistics (command “print stats”)

```
abc 02> print_stats
mul                : i/o =    4/    4  lat =    0  nd =     4  edge =     16  cube =     9  lev = 1
```

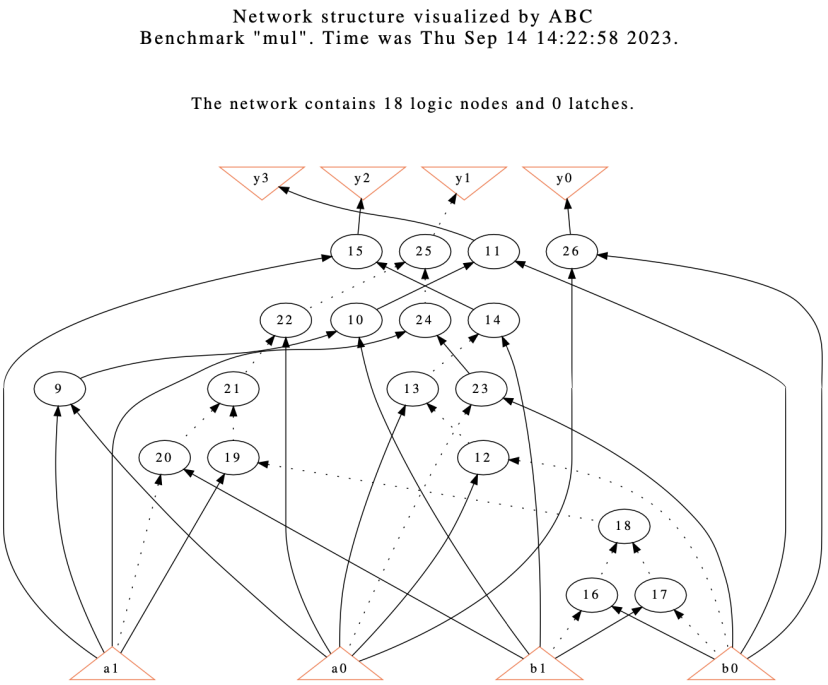
3. visualize the network structure (command “show”)

```
abc 02> show
```



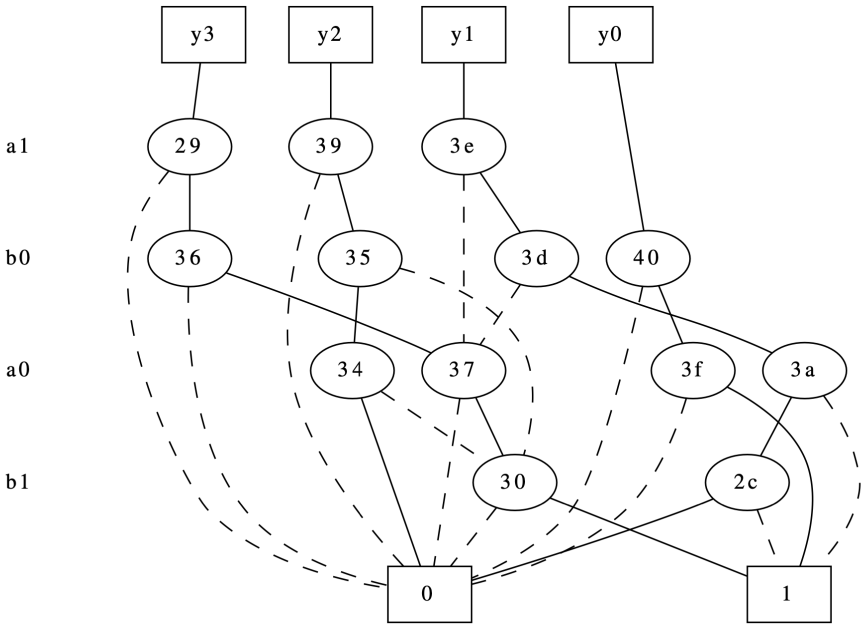
4. convert to AIG (command “strash”)
5. visualize the AIG (command “show”)

```
abc 02> strash
abc 03> show
```




6. convert to BDD (command “collapse”)
7. visualize the BDD

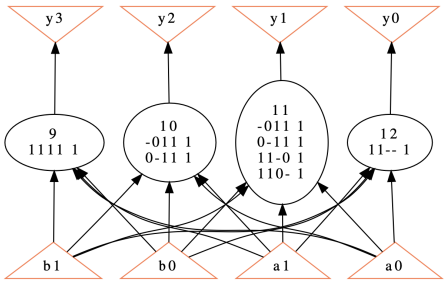
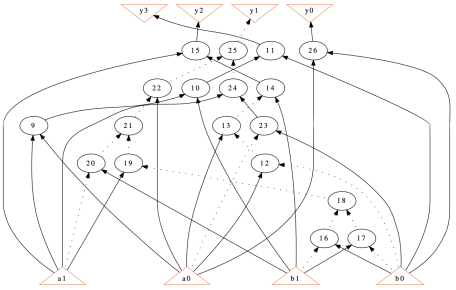
```
abc 03> collapse
abc 04> show_bdd -g
```



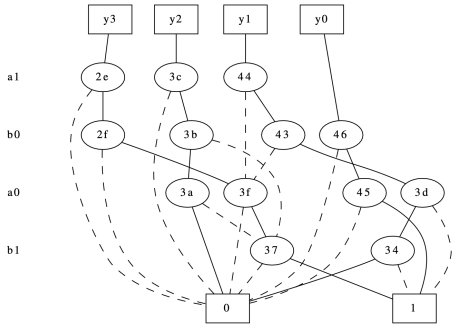
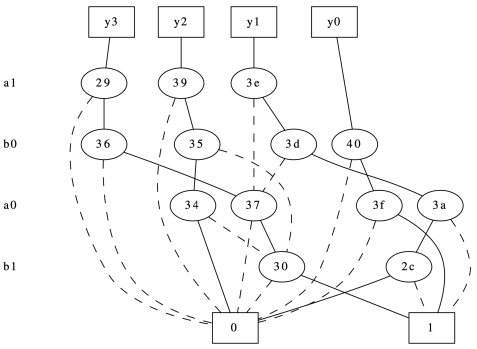
# 3 [ABC Boolean Function Representations]

(a) Compare the following differences with the two-bit unsigned multiplier example. Screenshot the results and briefly describe your findings in your report.

1. logic network in AIG (by command “aig”) vs.structurally hashed AIG (by command “strash” )

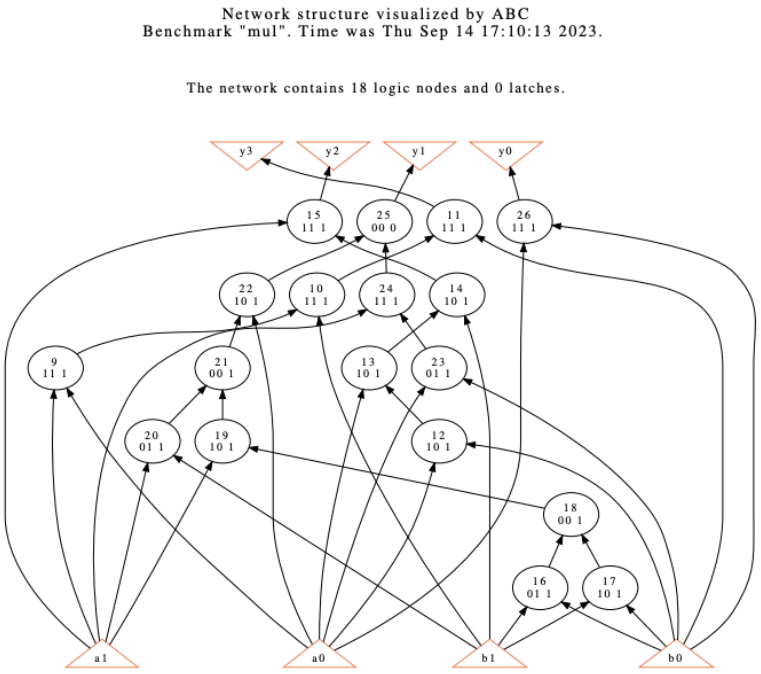
| | aig | strash |
|---|---|---|
| result |  |  |
| finding | Converts local functions of the nodes to AIGs.<br><br>It’s still the same as the original one. | Transforms the current network into an AIG by one-level structural hashing. The resulting AIG is a logic network composed of two-input AND gates and inverters represented as complemented attributes on the edges. |

2. logic network in BDD (by command “bdd”) vs.collapsed BDD (by command “collapse”)

| | bdd (show_bdd -g) | collapse (show_bdd -g) |
|---|---|---|
| result |  |  |
| finding | Converts local functions of the nodes to BDDs.<br>-<br>It’s still the same as the collapsed one. | Recursively composes the fanin nodes into the fanout nodes resulting in a network, in which each CO is produced by a node, whose fanins are CIs. Collapsing is performed by building global functions using BDDs and is, therefore, limited to relatively small circuits. After collapsing, the node functions are represented using BDDs. |

(b) Given a structurally hashed AIG, find a sequence of ABC commands to convert it to a logic network with node function expressed in sum-of-products (SOP). Use the two-bit unsigned multiplier example to test your command sequence, screenshot the results, and put them in your report.
ANS: command “logic”

```
abc 01> read mul.blif
abc 02> strash
abc 03> logic
```




Reference
https://people.eecs.berkeley.edu/~alanmi/abc/