

# LSV pa1 report

R14921060 歐信泓

## Problem2

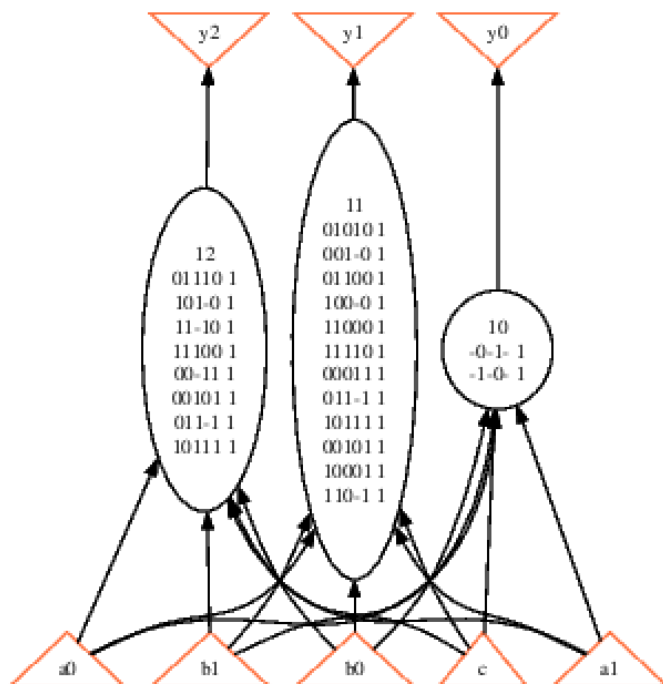
1. read the BLIF file into ABC (command “read”)
2. check statistics (command “print stats”)

```
abc 01> read lsv/pa1/alu.blif
abc 02> print_stats
alu : i/o = 5/ 3 lat = 0 nd = 3 edge = 15 cube = 22 lev = 1
```

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

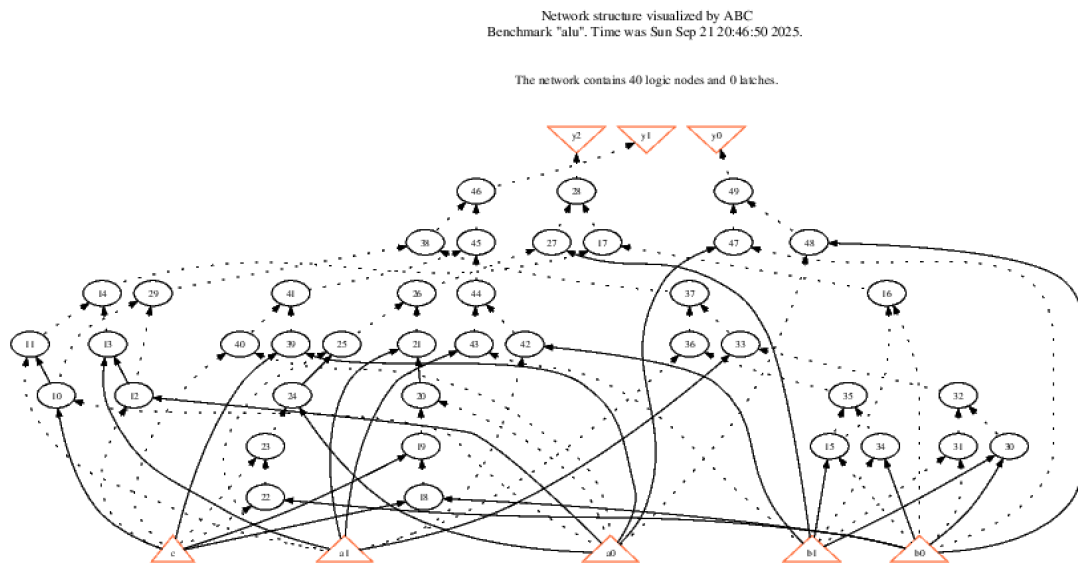
Network structure visualized by ABC  
Benchmark "alu". Time was Sun Sep 21 20:41:34 2025.

The network contains 3 logic nodes and 0 latches.



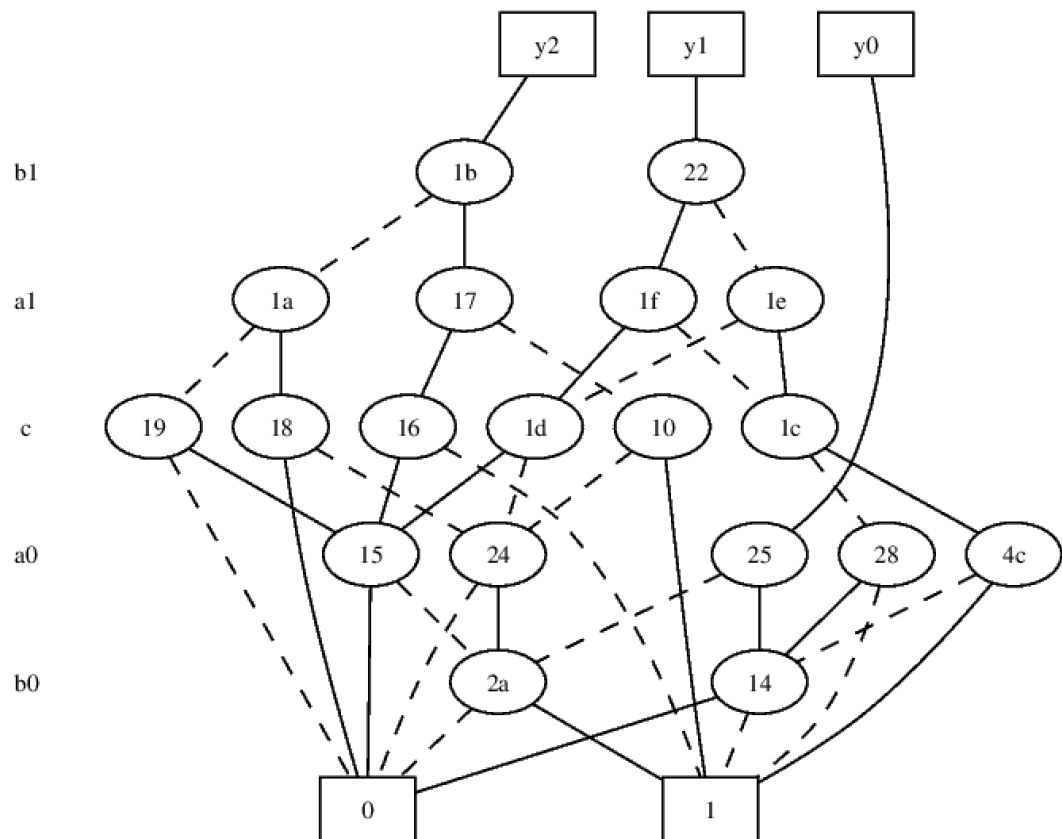
4. convert to AIG (command “strash”)

5. visualize the AIG (command “show”)



6. convert to BDD (command “collapse”)

7. visualize the BDD (command “show bdd -g”; note that “show bdd” only shows the first PO; option “-g” can be applied to show all POs)



### Problem3

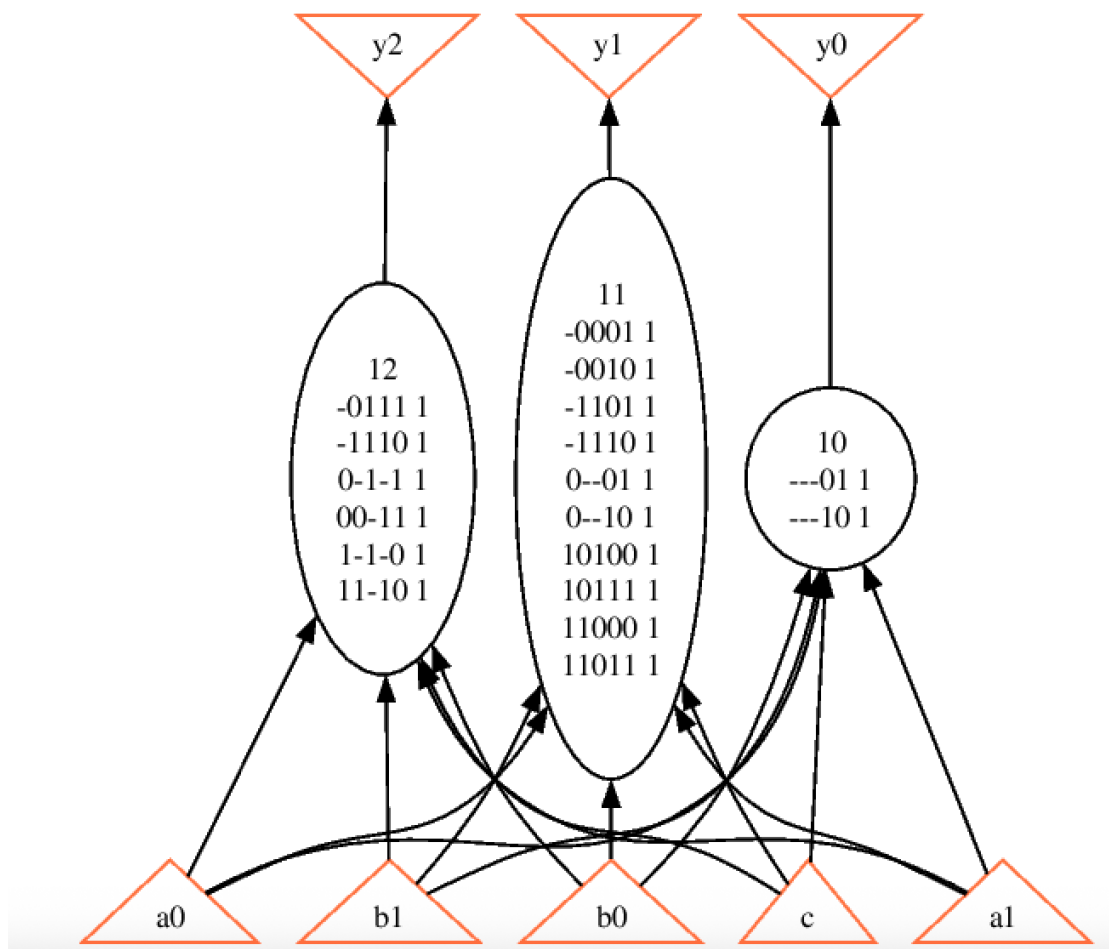
(a) Compare the following differences with your “alu.blif”. Screenshot the results and briefly describe your findings in your report.

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

logic network in AIG:

Network structure visualized by ABC  
Benchmark "alu". Time was Sun Sep 21 20:56:56 2025.

The network contains 3 logic nodes and 0 latches.

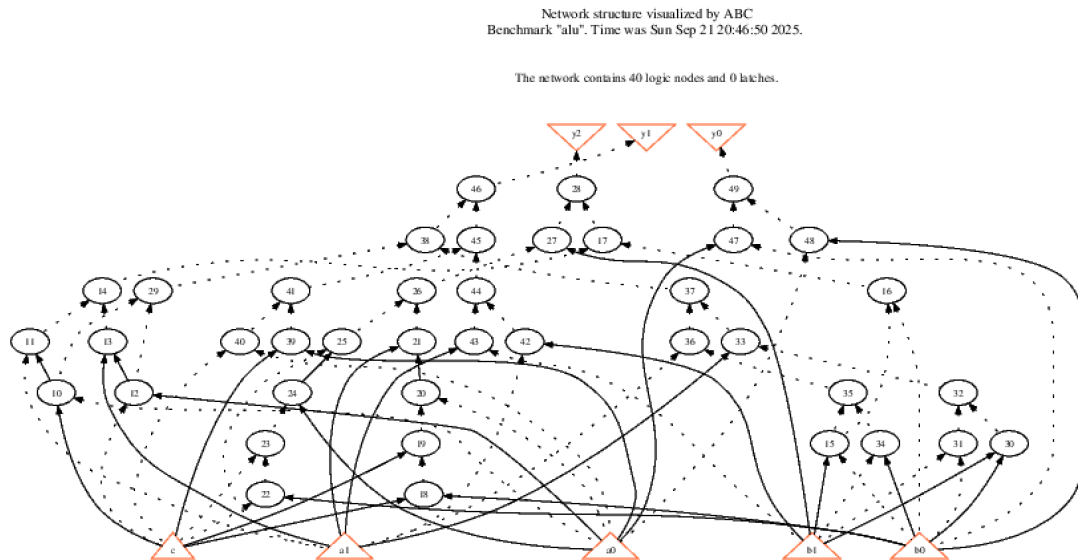


```

abc 06> aig
Error: The logic network is already in the AIG form.
abc 06> print_stats
alu : i/o = 5/ 3 lat = 0 nd = 3 edge = 15 aig = 43 lev = 1

```

structurally hashed AIG:



```

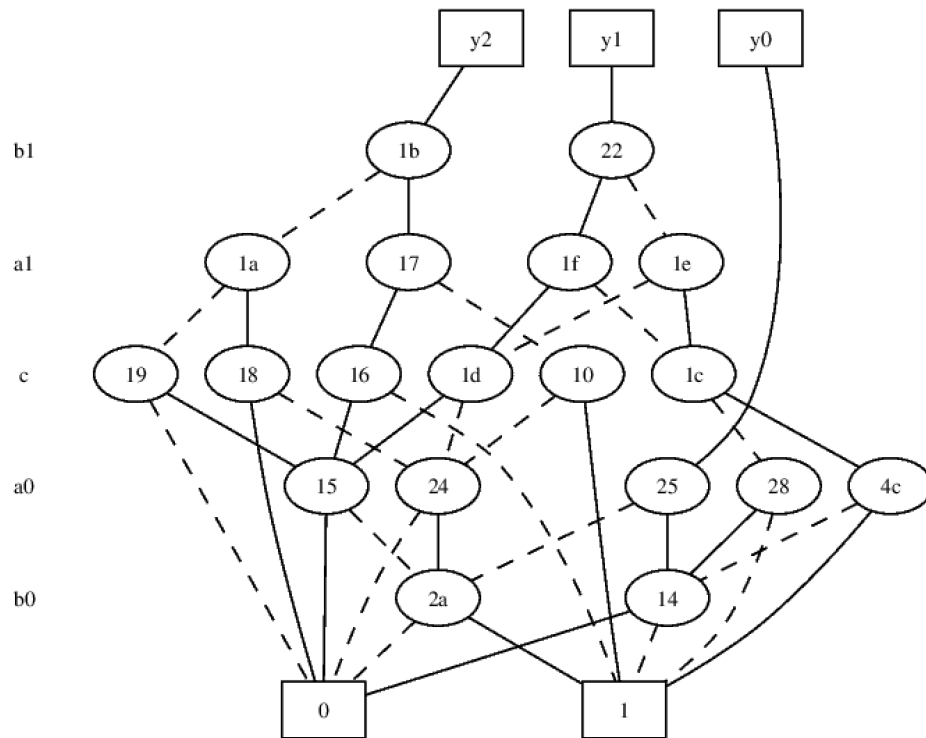
abc 04> strash
abc 05> print_stats
alu : i/o = 5/ 3 lat = 0 and = 32 lev = 6

```

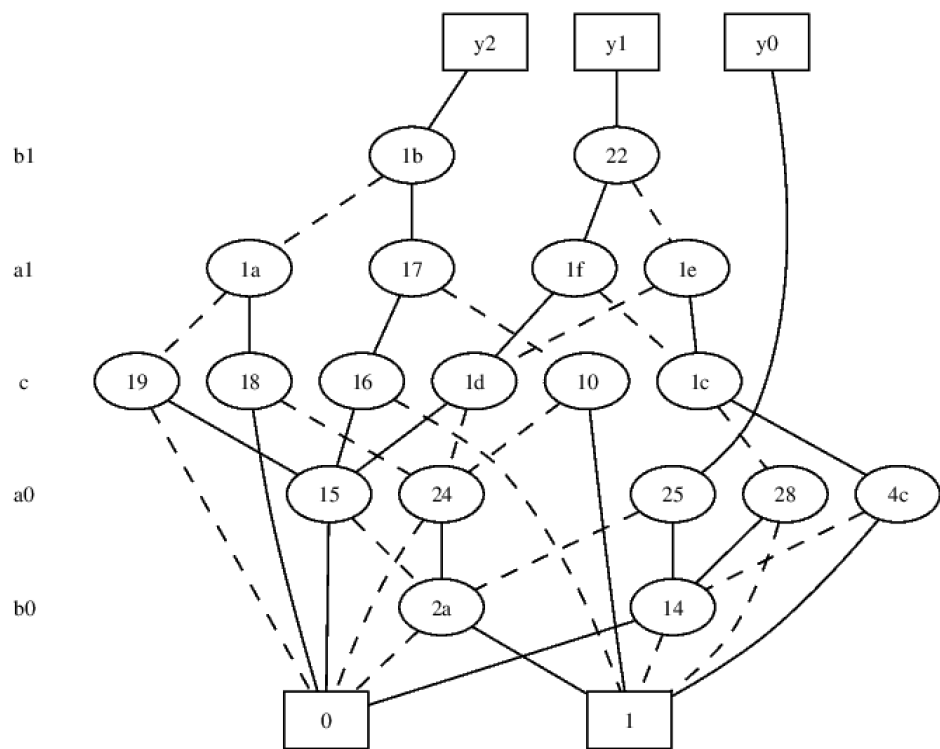
After structural hashing, number of aig nodes is reduced, but logic depth increases from 1 to 6.

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

logic network in BDD:



collapsed BDD:



These two command show the same bdd structure because in this case, outputs are directly written as a function of primary inputs without intermediate variables in the blif file.

- (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 your “alu.blif” to test your command sequence (by first running “strash” to convert it to AIG). Screenshot the results, and put them in your report.

```
abc 01> read_blif lsv/pa1/alu.blif
abc 02> aig
abc 02> strash
abc 03> logic
abc 04> show
```

