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
                                                                93 929,90
                                                O1, A0
                                                        6,60
                                                                0000
      .model mul
                                                        0 0
                                                    0
                                                                0000
      .inputs a1 a0 b1 b0
                                                        0
                                                    (')
                                                               0 0 0 0
                                                 0
      .outputs y3 y2 y1 y0
                                                           0
                                                    0
                                                        1
      .names a1 a0 b1 b0 y3
                                                                00
                                                                    9 0
      1111 1
                                                               0000
                                                        \mathbb{O}
                                                           D
                                                 0
      .names a1 a0 b1 b0 y2
                                                        0 (
                                                                    0 1
                                                 0
                                                    [
                                                                   ( )
                                                               00
      101- 1
                                                 0
      1110 1
                                                                0011
                                                 0)
      .names a1 a0 b1 b0 y1
                                                           0
                                                                0000
      011- 1
                                                    0
                                                        B
      10-1 1
                                                                0 (00
                                                         1
                                                   0
      1101 1
                                                                0110
                                                   0
      1110 1
                                                               0 9 0 0
                                                        0
                                                           0
      .names a1 a0 b1 b0 y0
                                                                0011
                                                        0 (
      01-1 1
                                                               0 (10
                                                        1
                                                           O
      11-1 1
                                                               1001
      .end
```

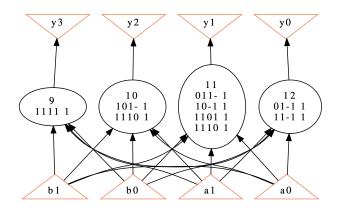
- (b) Perform the following steps to practice using ABC with the two-bit unsigned multiplier example.
- 1. read the BLIF file into ABC (command "read")

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

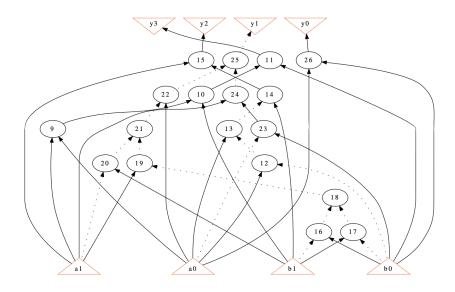


- 4. convert to AIG (command "strash")
- 5. visualize the AIG (command "show")

abc 02> strash abc 03> show

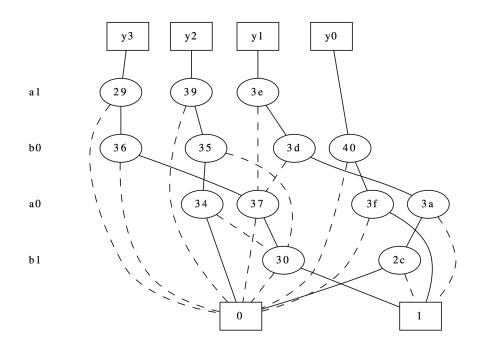
Network structure visualized by ABC Benchmark "mul". Time was Thu Sep 14 14:22:58 2023.

The network contains 18 logic nodes and 0 latches.



- 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	y3 y2 y1 y0 9 1111 1	22 10 11 26 22 10 13 13 23 11 26 23 13 13 23 11 26
finding	Converts local functions of the nodes to AIGs. 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)	collapse
result	y3 y2 y1 y0 9 10 -011 1 0-11 1 11-0 1 11-0 1 11 11-0 1 1 1-0 1 1 1-0 1 1 1-0 1 1 1-0 1 1 1-0 1 1 1-0 1 1 1-0 1 1 1-0 1 1 1 1	a1 29 39 3e y0 3e b0 36 33 3 3a 3
finding	Converts local functions of the nodes to BDDs. - It's still the same as the original 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 Cls. 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.

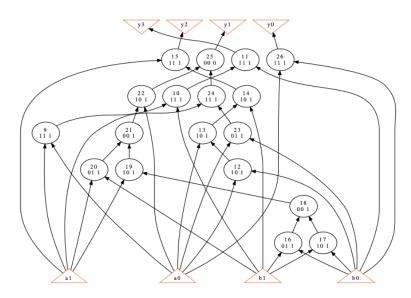
(b) Given a structurally hashed AIG, find a sequence of ABC commands to convert it to a logic network with node function expressed in sum-of-products (SOP). Use the two-bit unsigned multiplier example to test your command sequence, screenshot the results, and put them in your report.

ANS: command "logic"

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

Network structure visualized by ABC Benchmark "mul". Time was Thu Sep 14 17:10:13 2023.

The network contains 18 logic nodes and 0 latches.



Reference

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