

邏輯合成與驗證 HW1

R13922191 呂廷洋

2.

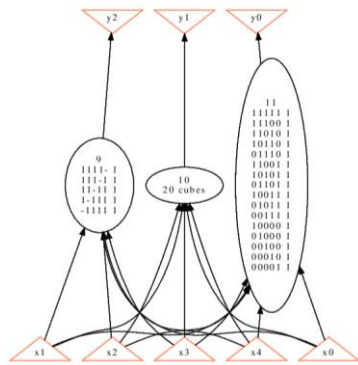
(b)

2.print_stats

```
abc 01> read comp.blif
abc 02> print_stats
comp : i/o = 5/ 3 lat = 0 nd = 3 edge = 15 cube = 41 lev = 1
```

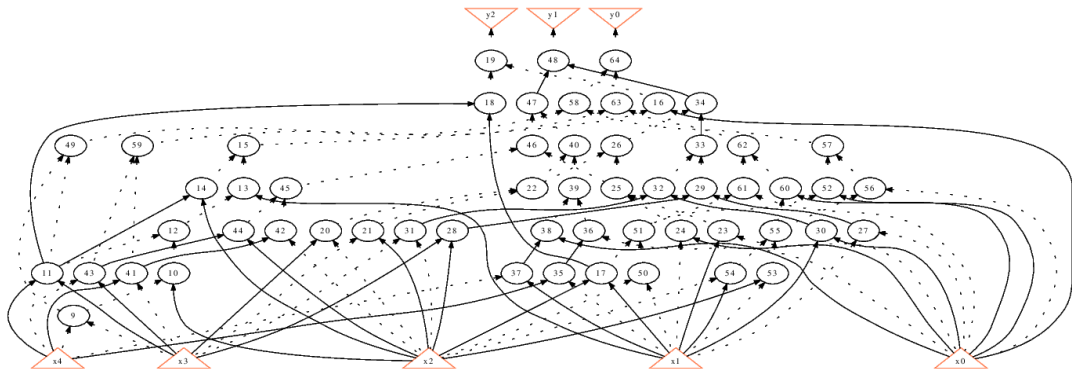
3.show (logic network)

The network contains 3 logic nodes and 0 latches.

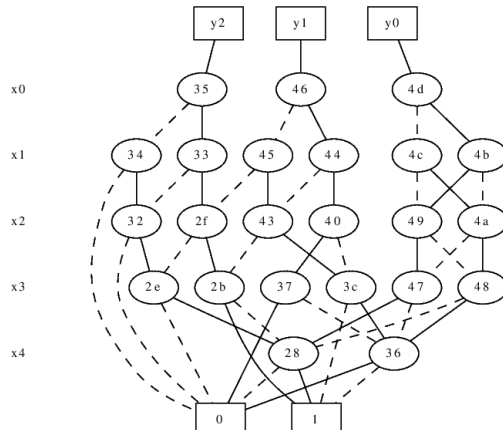


5.show (AIG)

The network contains 56 logic nodes and 0 latches.



7.show_bdd -g (BDD)

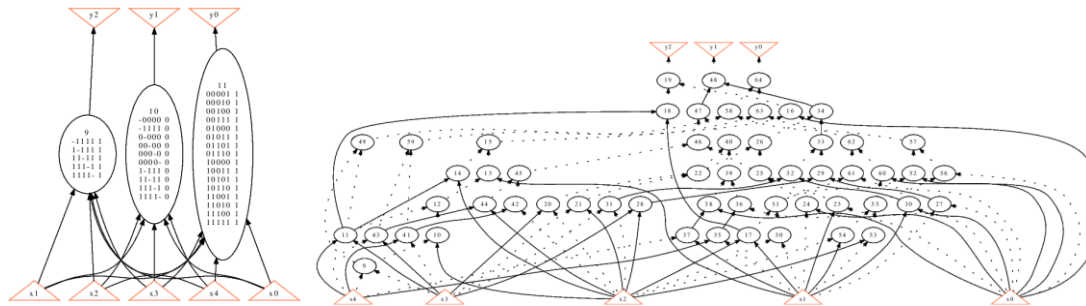


3.

(a)

1. AIG vs. structurally hashed AIG

For “aig”, the logic network has 3 nodes, and each node is formed by one or more aig. While “strash” contain only two-input ANDs and inverters with much more levels.

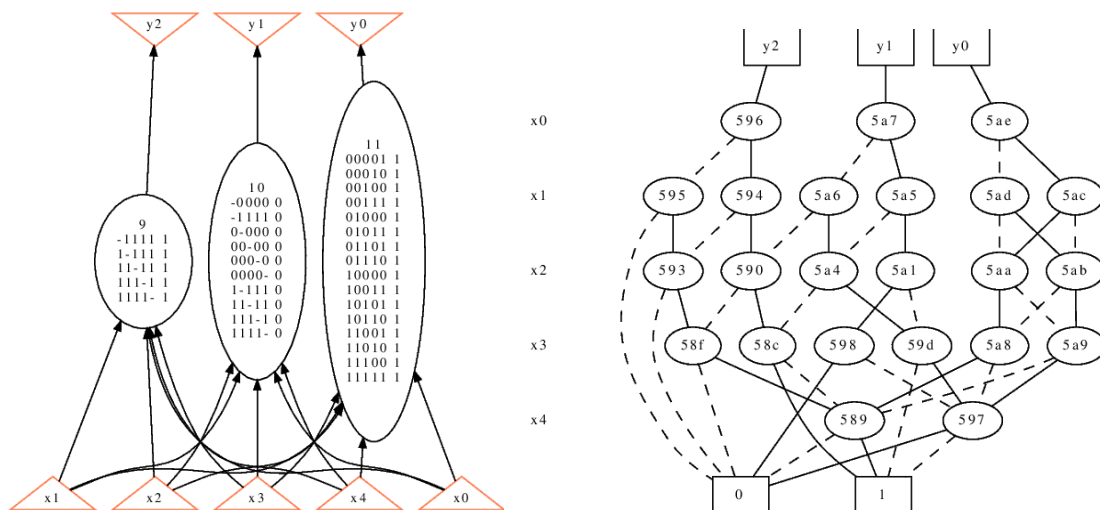


```
abc 01> read comp.blif
abc 02> aig
abc 02> print_stats
comp                               : i/o =   5/   3 lat =   0 nd =   3 edge =   15 aig =   61 lev = 1
abc 02> strash
abc 03> print_stats
comp                               : i/o =   5/   3 lat =   0 and =   56 lev = 7
```

2. BDD vs. collapsed BDD

Collapse BDD supposed to be more simple than BDD. That's because it merges the repeated nodes, without these redundant nodes, collapse BDD looks much smaller and has lesser levels.

However, when I use command “bdd”, the diagram doesn't exist any redundant node, thus “collapse” makes no change to the diagram.



```
abc 03> bdd
abc 03> print_stats
comp                               : i/o =   5/   3 lat =   0 nd =   3 edge =   15 bdd =   21 lev = 1
abc 03> collapse
abc 04> print_stats
comp                               : i/o =   5/   3 lat =   0 nd =   3 edge =   15 bdd =   21 lev = 1
```

(b)

Command “logic” & “sop” can convert an structurally hashed AIG into a logic network expressed in SOP.

