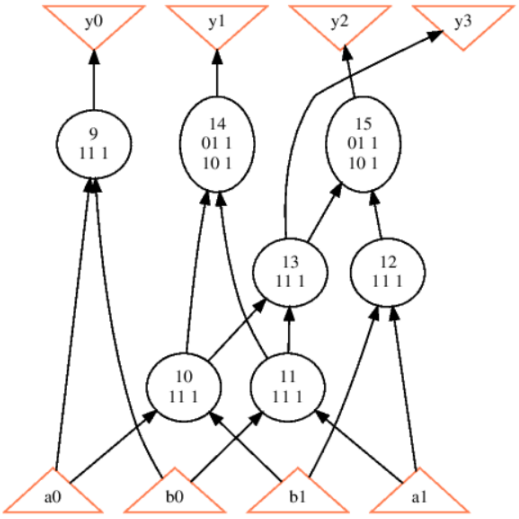
2.(b)

(1)read and print_stats

```
abc 01> read mul.blif
abc 02> print_stats
2bitmultiplier              : i/o =    4/    4  lat =    0  nd =     7  edge =     14  cube =     9  lev = 3
```

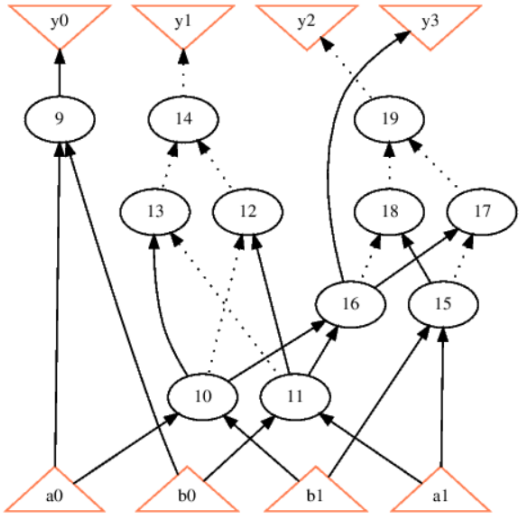
(2)show

The network contains 7 logic nodes and 0 latches.

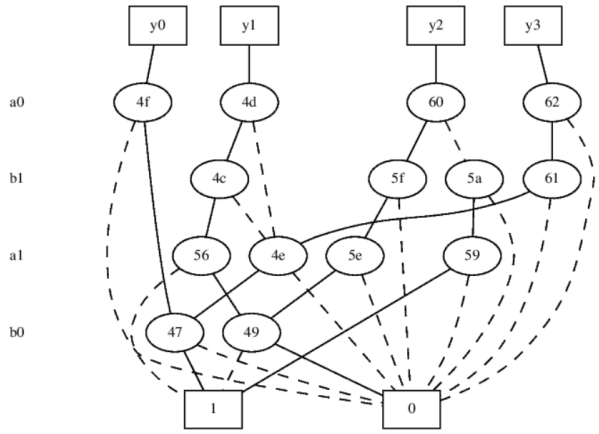


(3)strash and show

The network contains 11 logic nodes and 0 latches.



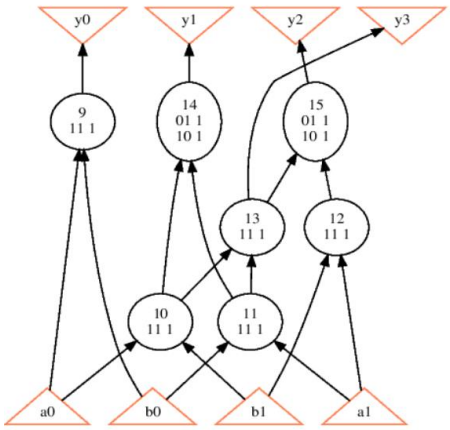
(4)collapse and show_bdd -g



3.(a)

1.use command “aig”

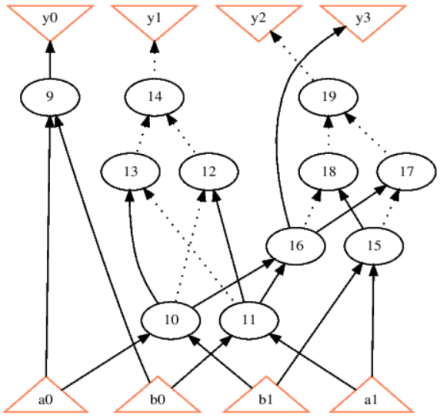
```
abc 01> read mul.blif
abc 02> aig
abc 02> print_stats
2bitmultiplier             : i/o =    4/    4  lat =    0  nd =     7  edge =     14  aig  =    11  lev = 3
```



Use command “strash”

```
abc 01> read mul.blif
abc 02> strash
abc 03> print_stats
2bitmultiplier             : i/o =    4/    4  lat =    0  and =     11  lev =  4
```

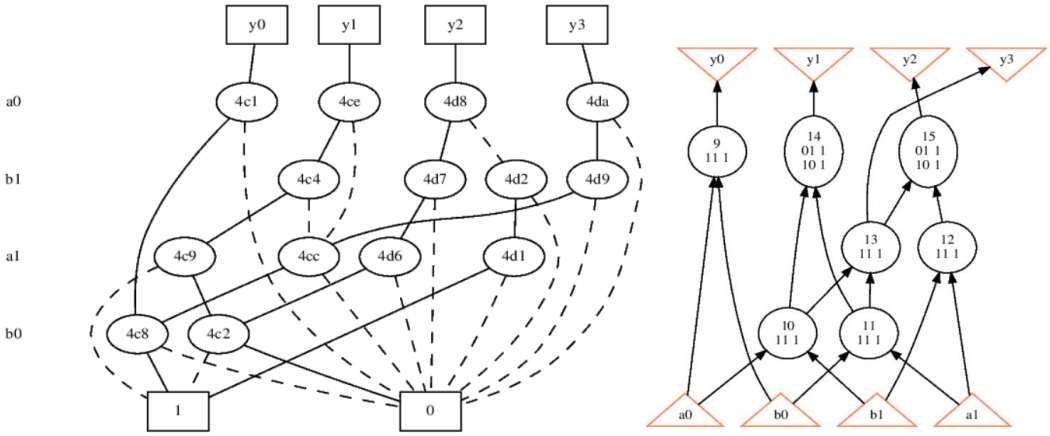
The network contains 11 logic nodes and 0 latches.



in print_stats, we can see the number of aig is 11 after command “aig”, it represents there are 11 and gates including in the total nodes. Because in node 14 and node 15 there are extra two and gates in them and the other nodes don’t have any extra nodes. But when it shows on the screen, it doesn’t show how they interpret in each node. Instead, the command “strash” is for global structural hash aig, so the whole circuit is converted to aig and shown on the screen.

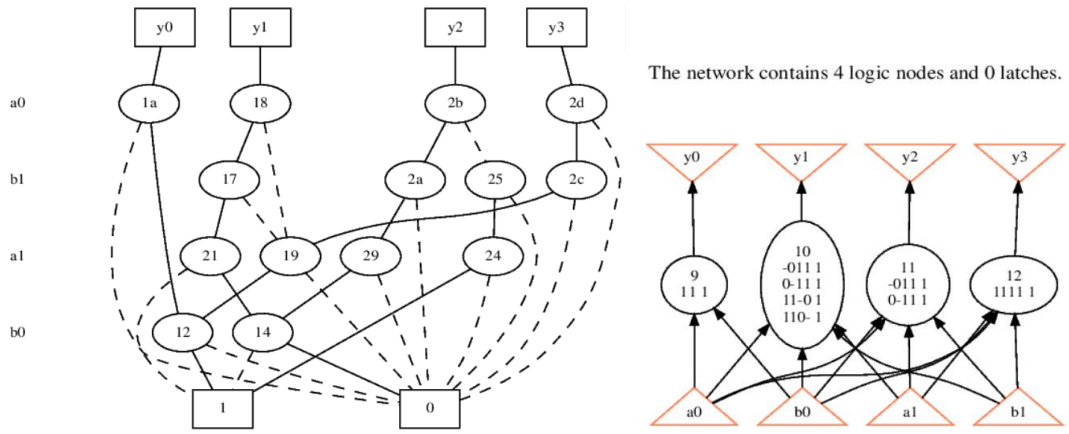
2.use command “bdd”

```
abc 01> read mul.blif
abc 02> bdd
abc 02> print_stats
2bitmultiplier            : i/o =    4/    4  lat =    0  nd =     7  edge =     14  bdd  =    14  lev = 3
```



Use command “collapse”

```
abc 01> read mul.blif
abc 02> collapse
abc 03> print_stats
2bitmultiplier            : i/o =    4/    4  lat =    0  nd =     4  edge =     14  bdd  =    16  lev = 1
```



We can see the level and the number of nodes doesn’t change after the command “bdd” because it only use the way of bdd to interpret each local nodes. And the command “collapse” change the whole network to the way of bdd, so we can see nodes are changed from input to output directly.

3.(b) after read and strash, we can get structurally hashed AIG. As long as typing command “renode” and then “sop”, it will be convert it to a logic network with

node function expressed in sum-of-products.

```
abc 01> read mul.blif
abc 02> strash
abc 03> renode
abc 04> sop
abc 04> print_stats
2bitmultiplier             : i/o =    4/    4  lat =    0  nd =     6  edge =     14  cube =     8  lev = 2
```

The network contains 6 logic nodes and 0 latches.