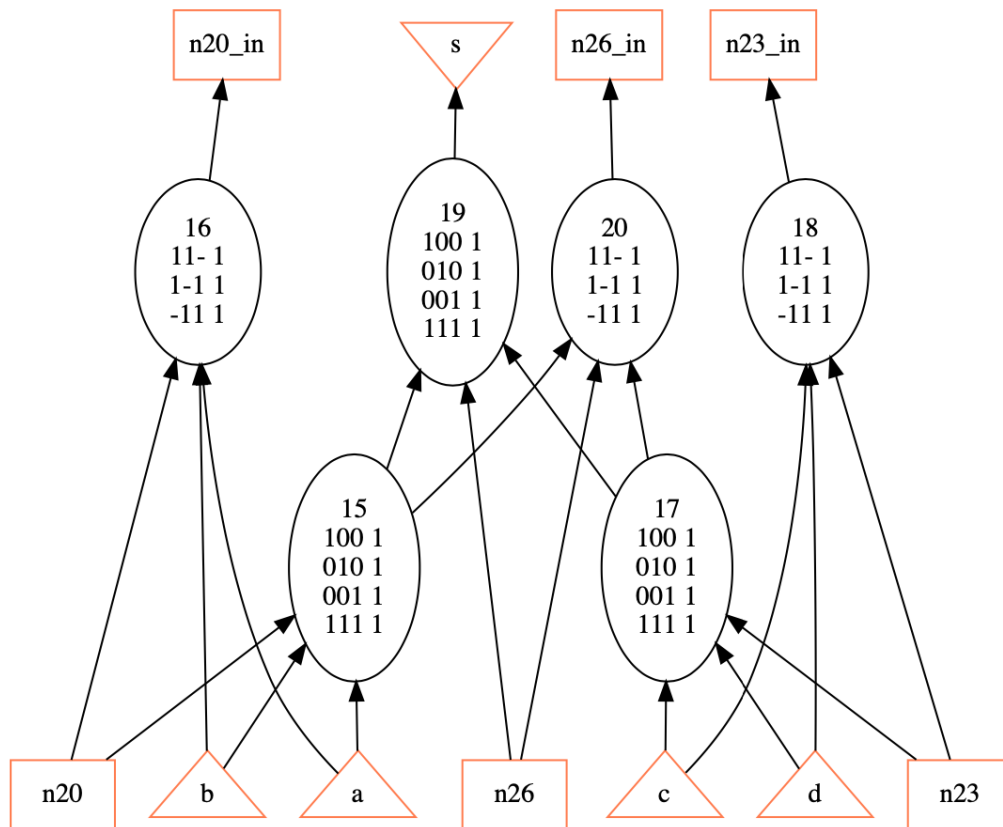


Part 1

Step3

Network structure visualized by ABC
Benchmark "serial_adder". Time was Mon Oct 4 15:15:13 2021.

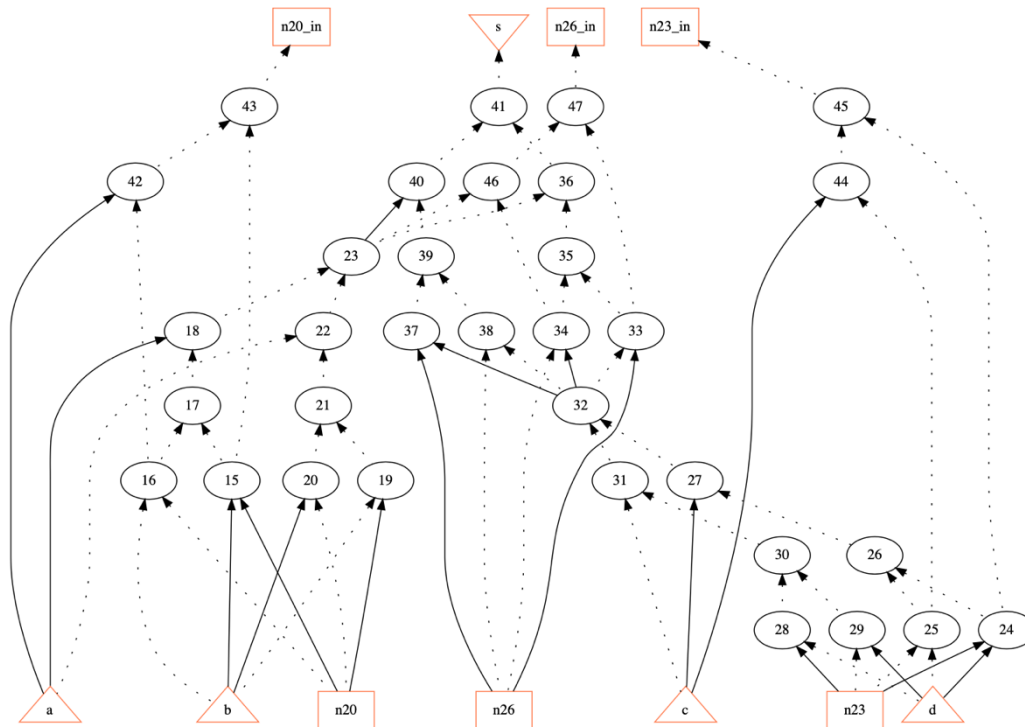
The network contains 6 logic nodes and 3 latches.



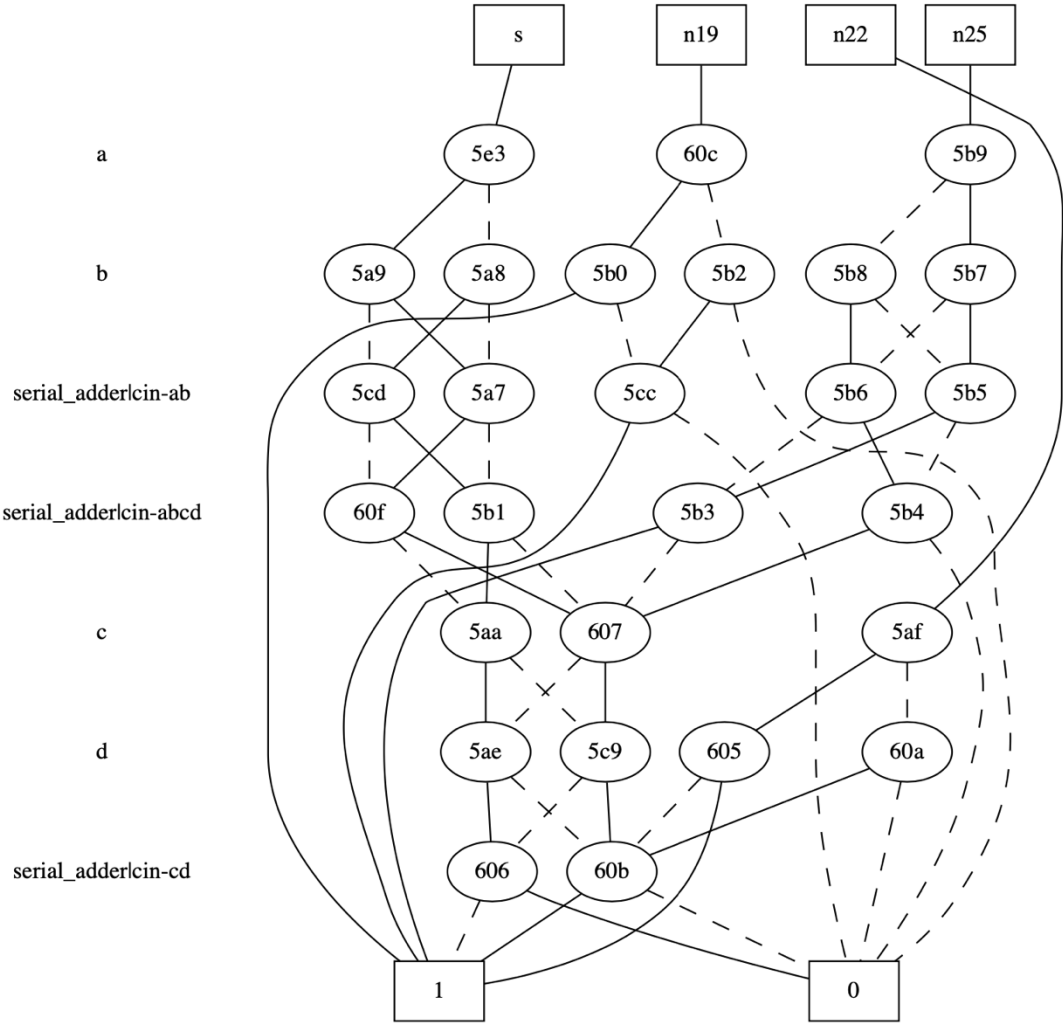
Step5 after 'strash'

Network structure visualized by ABC
Benchmark "serial_adder". Time was Mon Oct 4 15:23:37 2021.

The network contains 33 logic nodes and 3 latches.



Step7: after 'collapse' and 'show_bdd -g'



Part 2

(a) 1. logic network in AIG vs. structurally hashed AIG

If we read 'serial_adder.blif' file and input the command 'print_stats'.(Fig. 1)

```
i/o = 4/ 1 lat = 3 nd = 6 edge = 18 cube = 21 lev = 2
```

Fig. 1 read .blif file and print_stats

Initially, logic nodes are represented as cubes (SOP form). However, if we input the command 'aig', they will be transformed from SOPs into AIGs, we can verify by input the command 'print_stats' again. (Fig. 2)

```
i/o = 4/ 1 lat = 3 nd = 6 edge = 18 aig = 39 lev = 2
```

Fig. 2 print_stats after 'aig'

Then, when we input the command 'strash' and 'print_stats' (Fig. 3), we will find that not only local functions of the nodes turn to AIGs, but the whole current network are transformed into an AIG.

```
i/o = 4/ 1 lat = 3 and = 33 lev = 8
```

Fig. 3 print_stats after 'strash'

Using the command 'show' (Fig.4 and Fig. 5), can also show the difference between 'aig' and 'strash'.

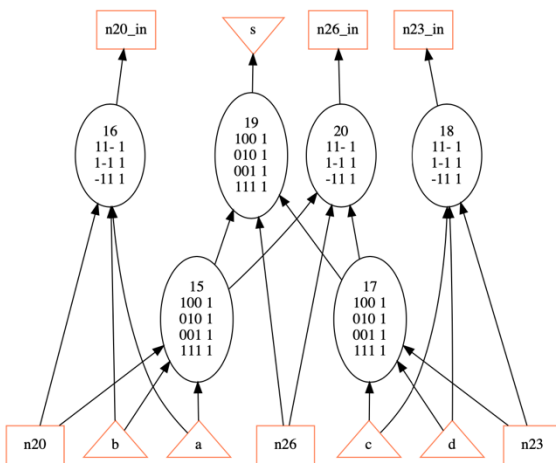


Fig. 4 show after 'aig'

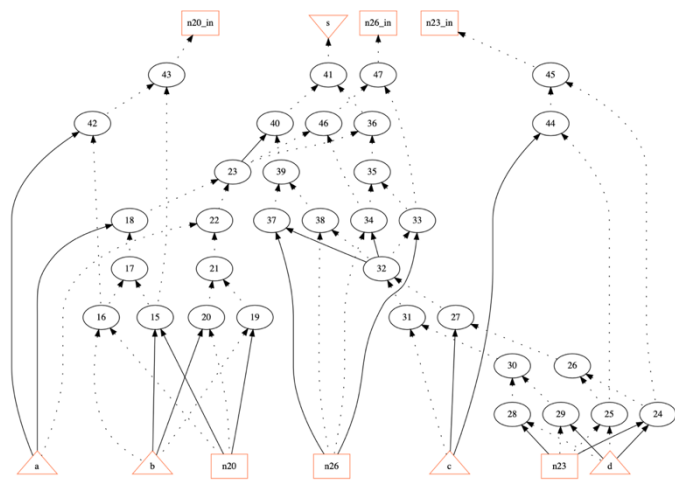


Fig. 5 show after 'strash'

From ABC help message can also found the difference between 'aig' and 'strash'

aig : converts node functions to AIG

strash : transforms combinational logic into an AIG

2. logic network in BDD vs. collapsed BDD

If we read .blif file and print_stats will show Fig. 6 below.

```
i/o = 4/ 1 lat = 3 nd = 6 edge = 18 cube = 21 lev = 2
```

Fig. 6 read .blif file and print_stats

Then input the command 'bdd' will convert node functions from SOP form to BDD form. (Fig. 7)

```
i/o = 4/ 1 lat = 3 nd = 6 edge = 18 bdd = 21 lev = 2
```

Fig. 7 print_stats after 'bdd'

Then if we input the command 'collapse', it will collapse the network by constructing global BDD. We can verify it by first input the command 'show' after input the command 'bdd' and after input the command 'collapse'. (Fig. 8 and Fig. 9), and can also find the difference by the command 'show_bdd'. (Fig. 10 and Fig. 11)

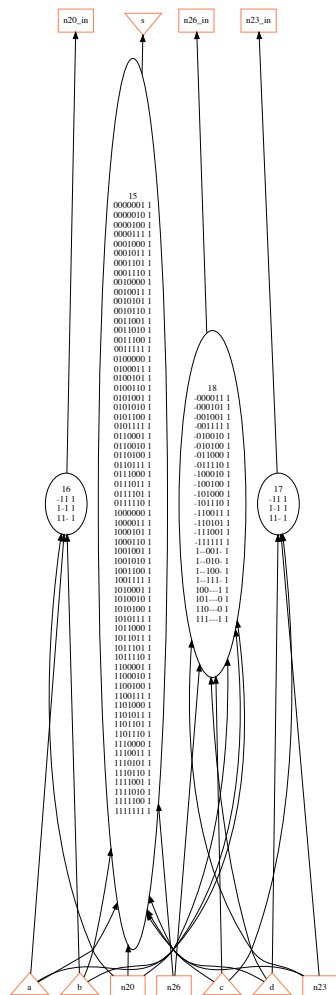


Fig. 8 show after 'collapse'

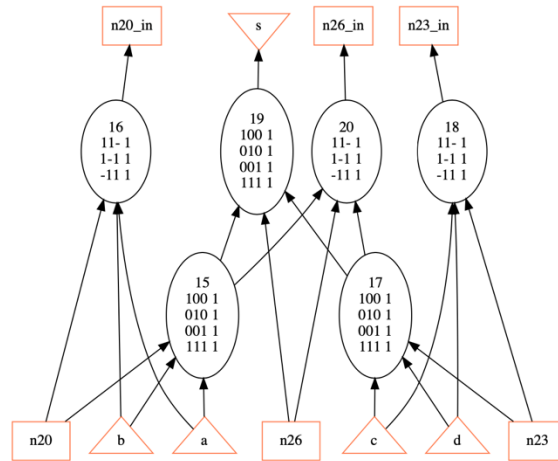


Fig. 9 show after 'bdd'

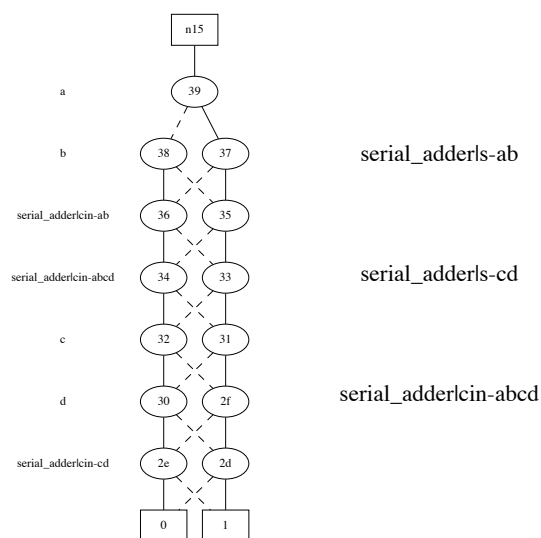


Fig. 10 show_bdd after 'collapse'

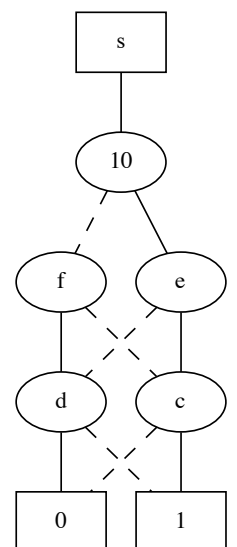


Fig. 11 show_bdd after 'bdd'

From ABC help message can also tell the difference between 'bdd' and 'collapse'

bdd : converts node functions to BDD

collapse : collapses the network by constructing global BDDs

(b) Ans: by command 'logic'

using the command 'logic -h', it will show Fig. 12, and input the command 'show' will

output Fig. 13

```
abc 18> logic -h
```

```
usage: logic [-h]
```

```
      transforms an AIG into a logic network with SOPs
```

```
-h      : print the command usage
```

Fig. 12 the help message of the command 'logic'

The network contains 33 logic nodes and 3 latches.

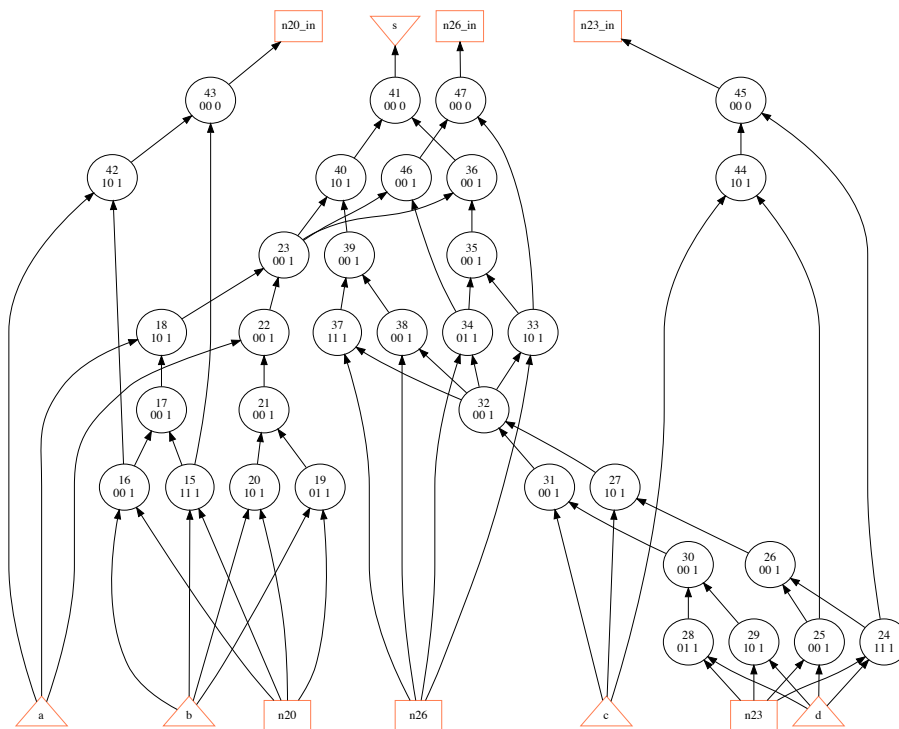


Fig. 13 show after command 'logic'