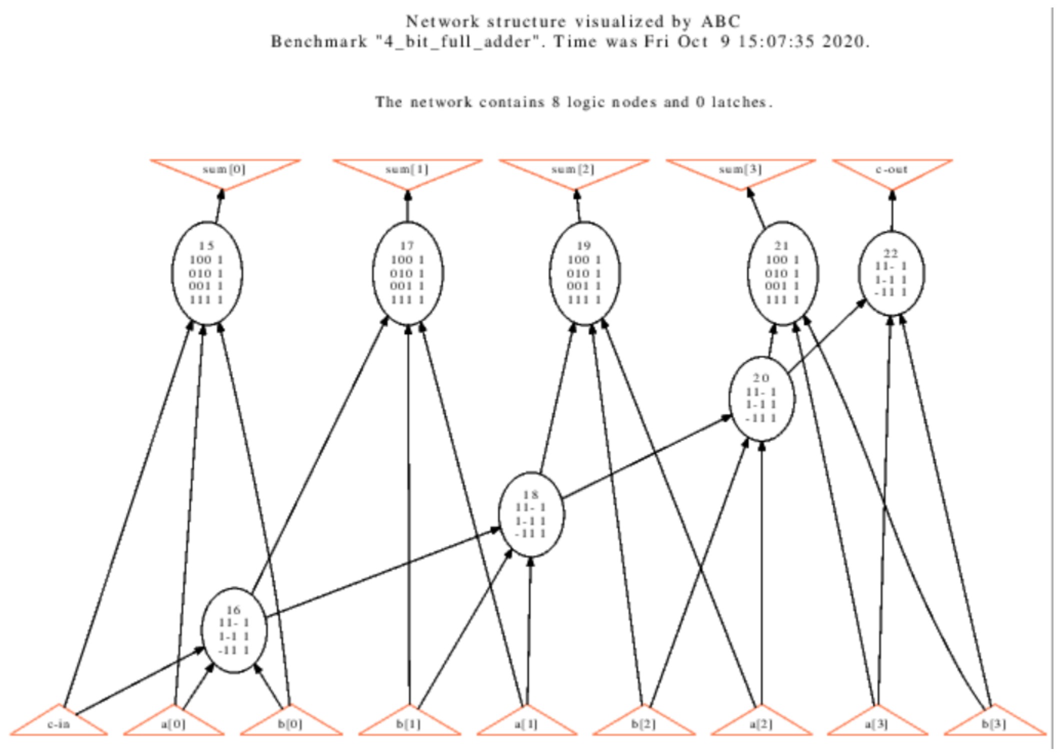
Part 1

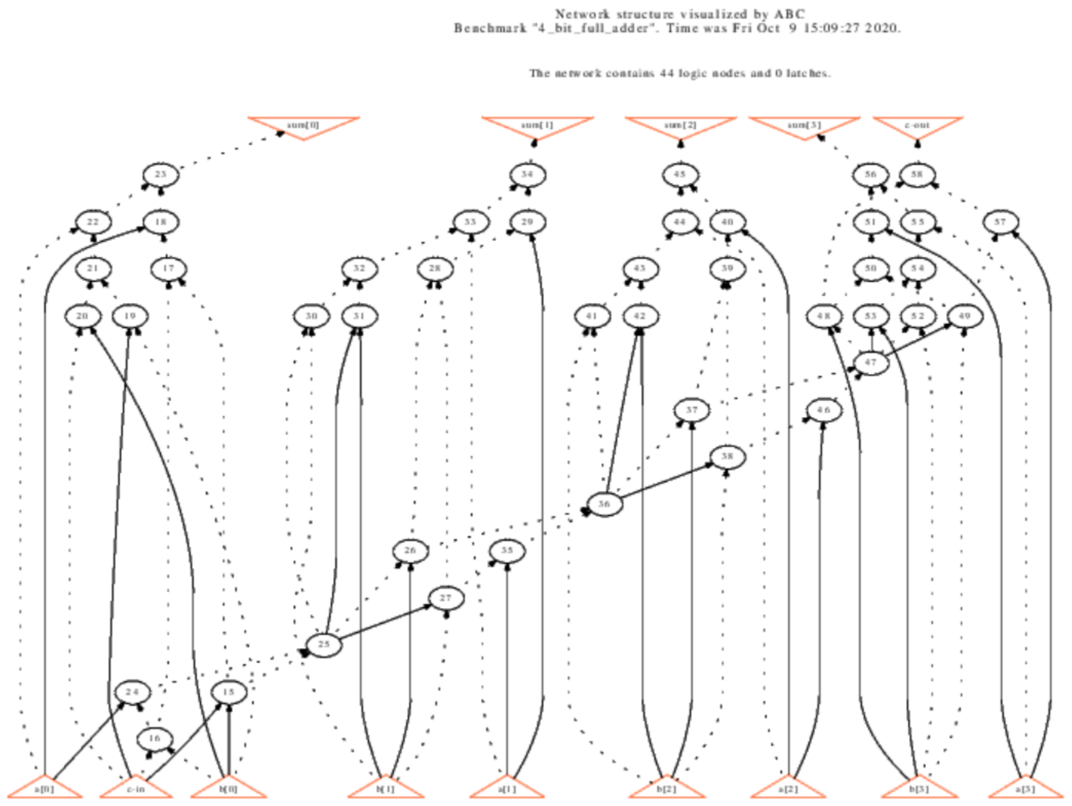
visualize the network structure (command show)



Network structure visualized by ABC
Benchmark "4_bit_full_adder". Time was Fri Oct 9 15:07:35 2020.

The network contains 8 logic nodes and 0 latches.

visualize the AIG (command show)

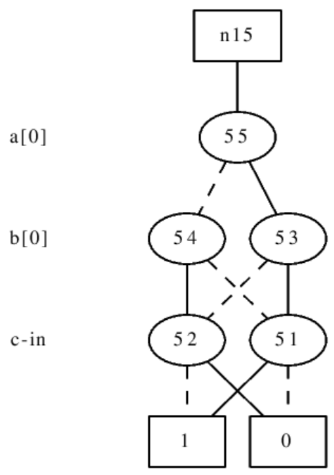


Network structure visualized by ABC
Benchmark "4_bit_full_adder". Time was Fri Oct 9 15:09:27 2020.

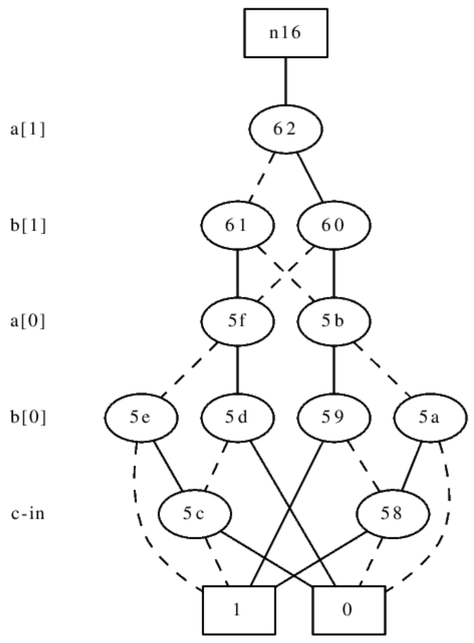
The network contains 44 logic nodes and 0 latches.

visualize the BDD (command show_bdd)

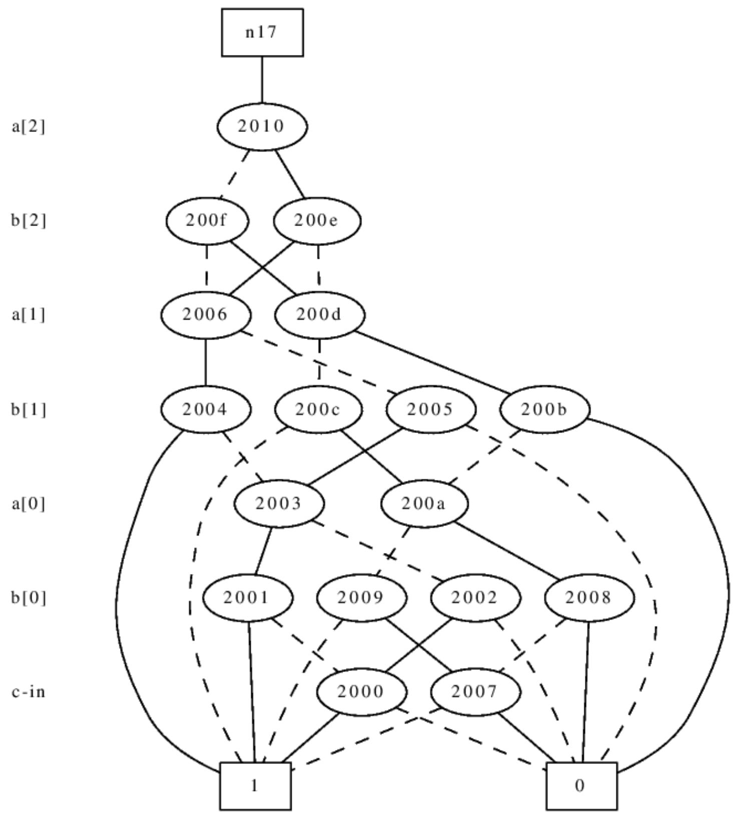
sum[0]



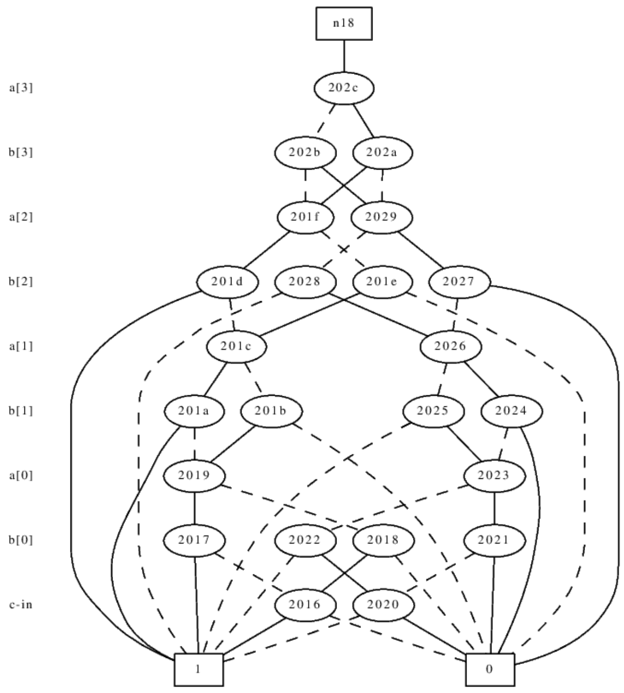
sum[1]



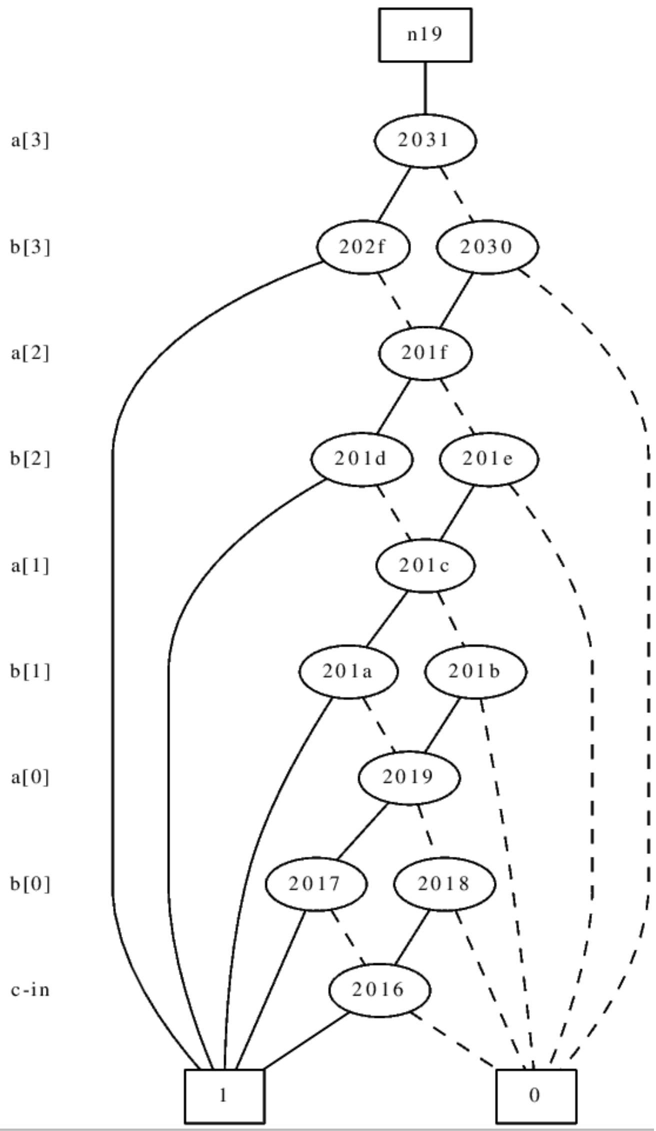
sum[2]


n17
a[2]
2010
b[2]
200f
200e
a[1]
2006
200d
b[1]
2004
200c
2005
200b
a[0]
2003
200a
b[0]
2001
2009
2002
2008
c-in
2000
2007
1
0

sum[3]


n18
a[3]
202c
b[3]
202b
202a
a[2]
201f
2029
b[2]
201d
2028
201e
2027
a[1]
201c
2026
b[1]
201a
201b
2025
2024
a[0]
2019
2023
b[0]
2017
2022
2018
2021
c-in
2016
2020
1
0

c-out

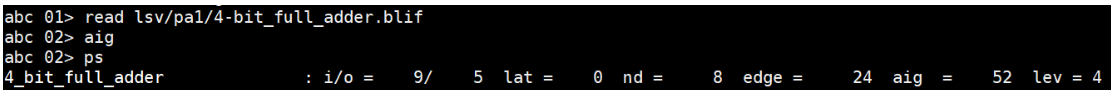

n19
a[3]
2031
b[3]
202f
2030
a[2]
201f
b[2]
201d
201e
a[1]
201c
b[1]
201a
201b
a[0]
2019
b[0]
2017
2018
c-in
2016
1
0

Part 2

(a)

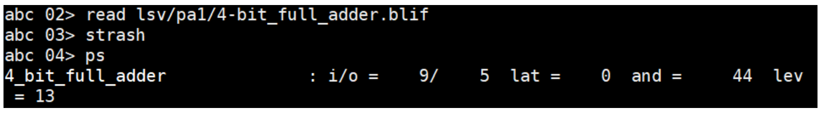
1.

aig locally converts the representation of each node of the logic network to AIG.



```
abc 01> read lsv/pa1/4-bit_full_adder.blif
abc 02> aig
abc 02> ps
4_bit_full_adder              : i/o =    9/    5  lat =    0  nd =     8  edge =     24  aig  =    52  lev = 4
```

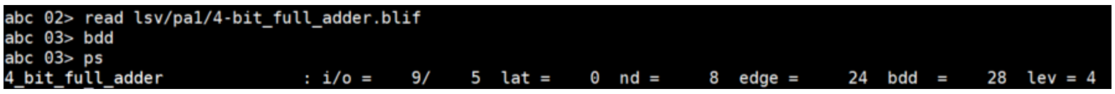
strash globally transforms combinational logic into AIG.



```
abc 02> read lsv/pa1/4-bit_full_adder.blif
abc 03> strash
abc 04> ps
4_bit_full_adder              : i/o =    9/    5  lat =    0  and =     44  lev
 = 13
```

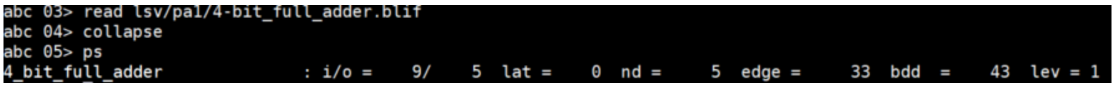
2.

bdd locally converts the representation of each node of the logic network to BDD.



```
abc 02> read lsv/pa1/4-bit_full_adder.blif
abc 03> bdd
abc 03> ps
4_bit_full_adder              : i/o =    9/    5  lat =    0  nd =     8  edge =     24  bdd  =    28  lev = 4
```

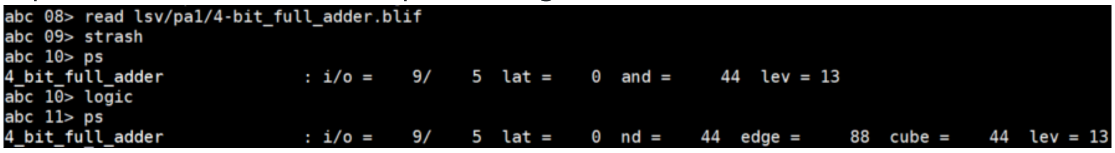
collapse collapses the network by constructing global BDDs.



```
abc 03> read lsv/pa1/4-bit_full_adder.blif
abc 04> collapse
abc 05> ps
4_bit_full_adder              : i/o =    9/    5  lat =    0  nd =     5  edge =     33  bdd  =    43  lev = 1
```

(b)

logic can convert a structurally hashed AIG to a logic network with the SOP representation of the two-input AND-gates.



```
abc 08> read lsv/pa1/4-bit_full_adder.blif
abc 09> strash
abc 10> ps
4_bit_full_adder              : i/o =    9/    5  lat =    0  and =     44  lev = 13
abc 10> logic
abc 11> ps
4_bit_full_adder              : i/o =    9/    5  lat =    0  nd =    44  edge =     88  cube =    44  lev = 13
```