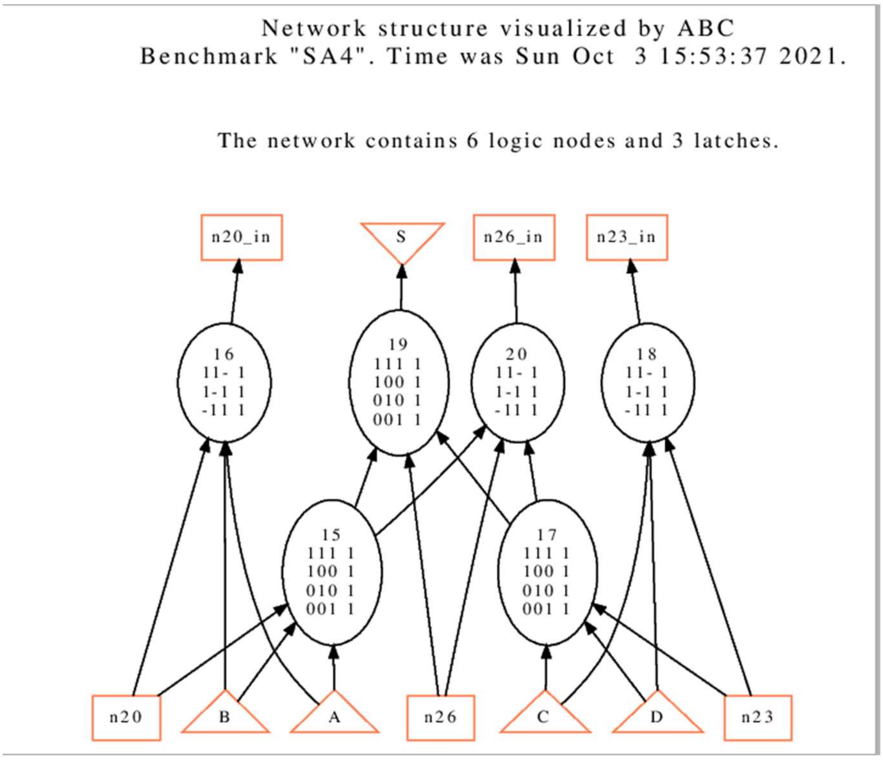
Logic Synthesis & Verification, Fall 2021

Programming Assignment 1

Part 1, 2 Report

Part1

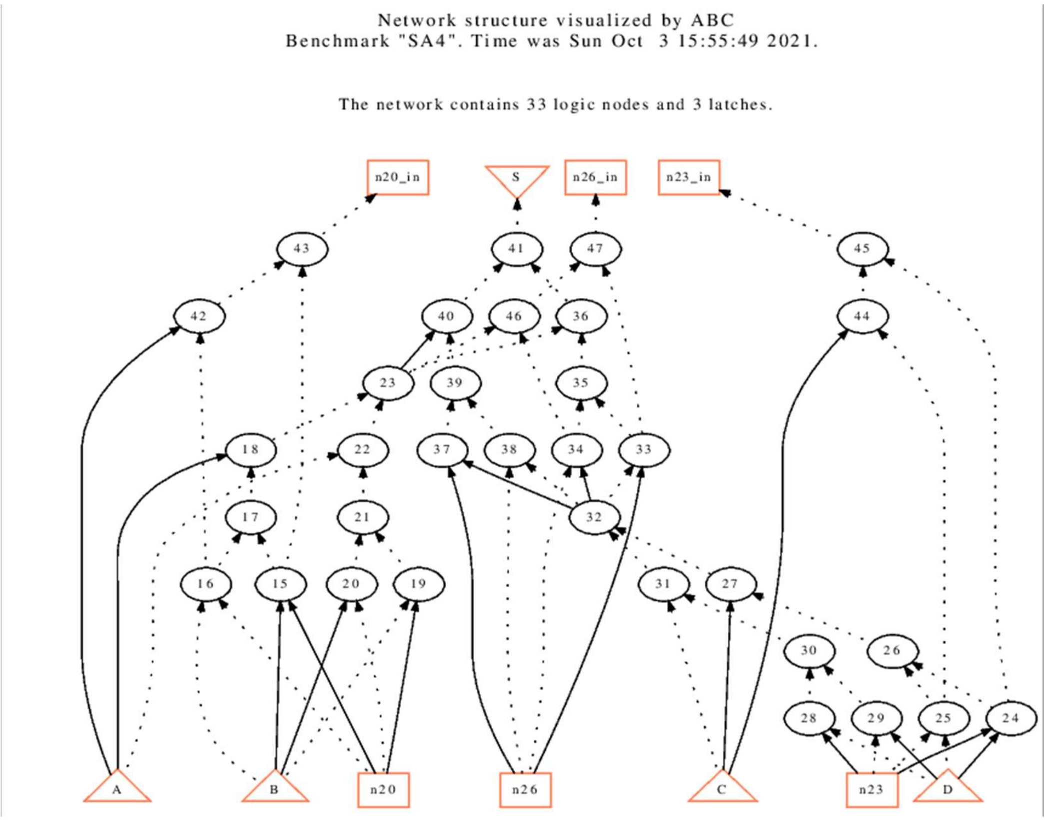
After Step 3 …



Network structure visualized by ABC
Benchmark "SA4". Time was Sun Oct 3 15:53:37 2021.

The network contains 6 logic nodes and 3 latches.

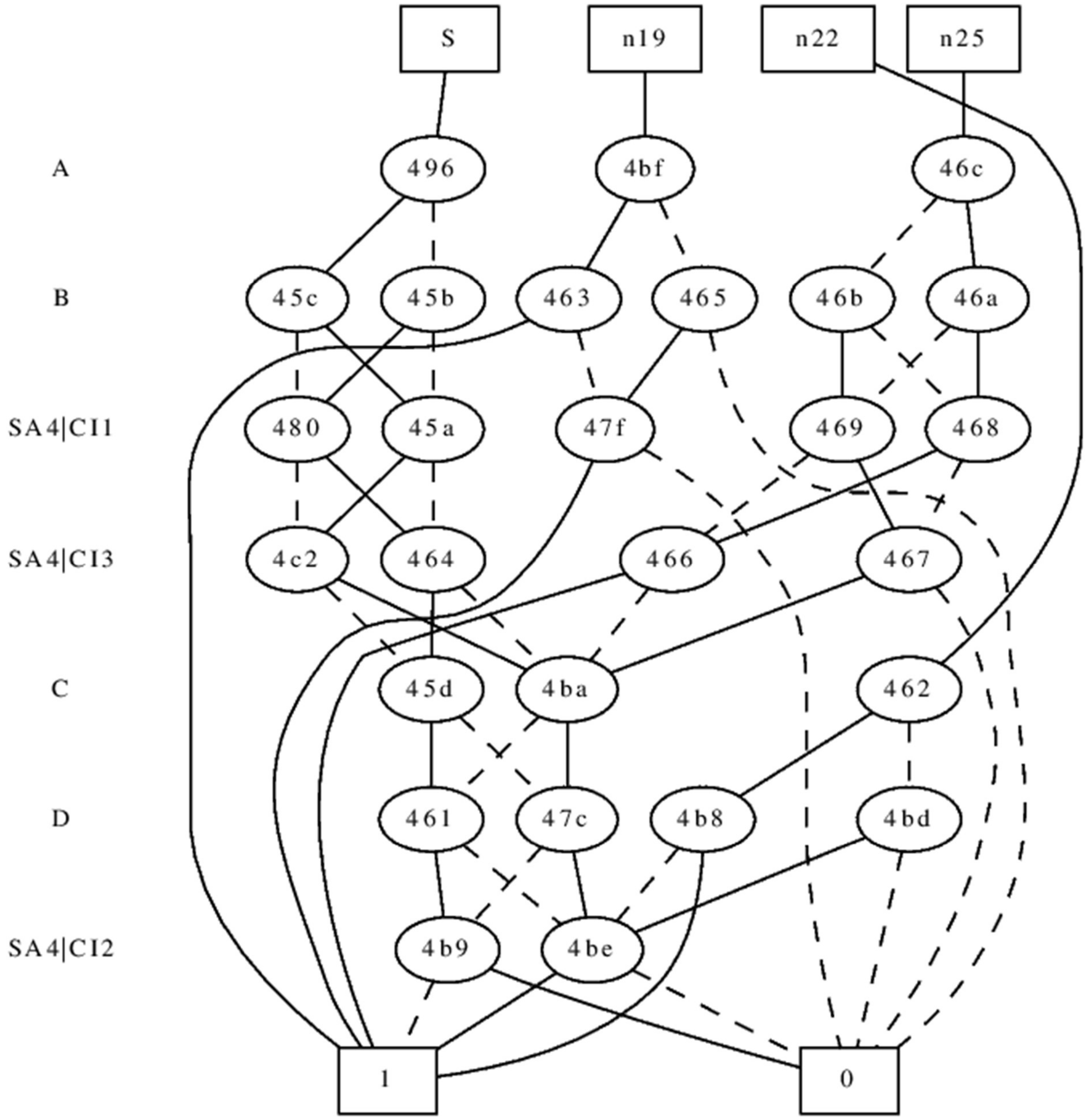
After Step 5 … (strashed AIG)



Network structure visualized by ABC
Benchmark "SA4". Time was Sun Oct 3 15:55:49 2021.

The network contains 33 logic nodes and 3 latches.

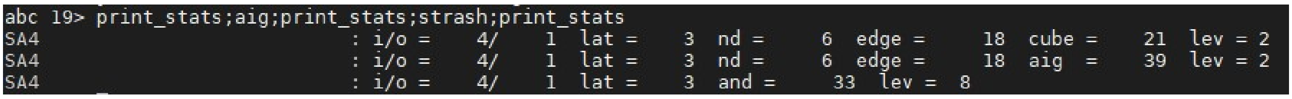
After Step 7 … (collapsed BDD)



Part2

(a) Compare the following differences with the four-number serial adder example.

1. logic network in AIG (by command “aig”) vs. structurally hashed AIG (by command “strash”)



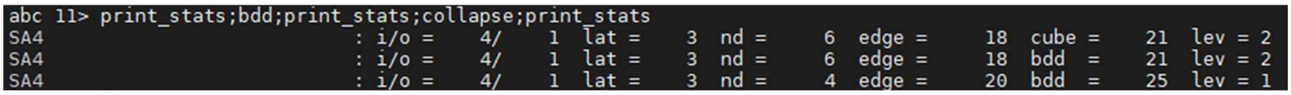
```
abc 19> print_stats;aig;print_stats;strash;print_stats
SA4                          : i/o =    4/    1  lat =    3  nd =     6  edge =     18  cube =    21  lev = 2
SA4                          : i/o =    4/    1  lat =    3  nd =     6  edge =     18  aig  =    39  lev = 2
SA4                          : i/o =    4/    1  lat =    3  and =     33  lev =  8
```

Fig 1. Stats of original, aig and strashed networks

The *“aig”* command only converts local functions of the nodes to AIGs, therefore the number of nodes and levels is still the same, only the expressions of node functions are converted from *cube* to *aig*.

The *“strash”* command transforms the current network into an AIG by one-level structural hashing, the network is reconstructed with the *and* gate and inverters represented as complemented attributes on the edges, therefore the number of aigs and levels is changed.

2. logic network in BDD (by command "bdd") vs. collapsed BDD (by command "collapse")



```
abc 11> print_stats;bdd;print_stats;collapse;print_stats
SA4                           : i/o =    4/    1  lat =    3  nd =     6  edge =     18  cube =    21  lev = 2
SA4                           : i/o =    4/    1  lat =    3  nd =     6  edge =     18  bdd  =    21  lev = 2
SA4                           : i/o =    4/    1  lat =    3  nd =     4  edge =     20  bdd  =    25  lev = 1
```

Fig 2. Stats of original, bdd and collapsed networks

The *"bdd"* command only converts local functions of the nodes to BDDs, therefore the number of nodes and levels is still the same, only the expressions of node functions are converted from *cube* to *bdd*.

The *"collapse"* command recursively composes the fanin nodes into the fanout nodes, the network is collapsed into one level and the number of nodes is also reduced.

The "*show_bdd*" command visualizes the network with BDD in the nodes connects directly to the POs, therefore the visualizations of BDD after the "*bdd*" command may have both internal nets and PIs as inputs, while the visualizations of BDD after the "*collapse*" command only have the PIs as inputs.

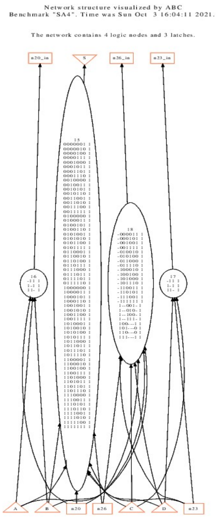

Network structure visualized by ABC
Benchmark "SA4". Time was Sun Oct 3 16:04:11 2021.
The network contains 4 logic nodes and 3 latches.
n20_in
S
n26_in
n23_in
15

Fig 3. Visualization of network after "collapse" command

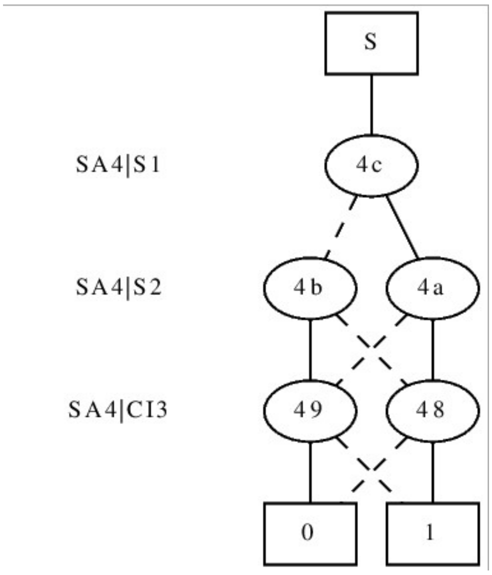


Fig 4. BDD after "bdd" command

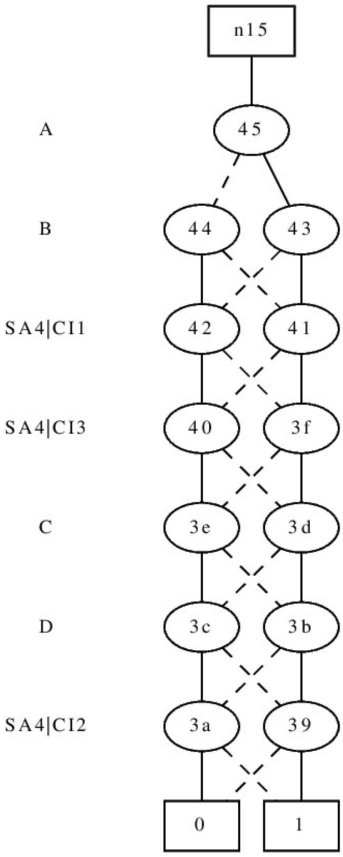


Fig 5. BDD after "collapse" command

(b) Given a structurally hashed AIG, find a sequence of ABC command(s) to covert it to a logic network with node function expressed in sum-of-products (SOP).

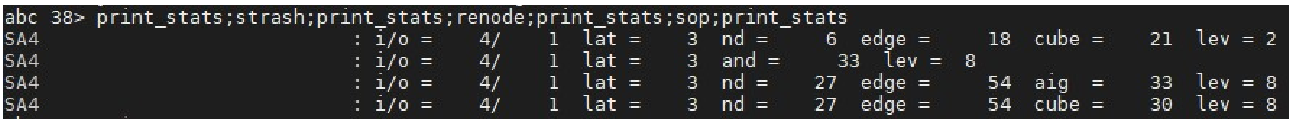
After strash …

abc> renode

#Creates node boundaries in this AIG and collapses the intermediate logic to form larger nodes

abc> sop

# Converts local functions of the nodes to SOPs.



```
abc 38> print_stats;strash;print_stats;renode;print_stats;sop;print_stats
SA4                           : i/o =    4/    1  lat =    3  nd =     6  edge =     18  cube =    21  lev = 2
SA4                           : i/o =    4/    1  lat =    3  and =     33  lev =  8
SA4                           : i/o =    4/    1  lat =    3  nd =    27  edge =     54  aig  =    33  lev = 8
SA4                           : i/o =    4/    1  lat =    3  nd =    27  edge =     54  cube =    30  lev = 8
```

Fig 7. Stats after each abc commands

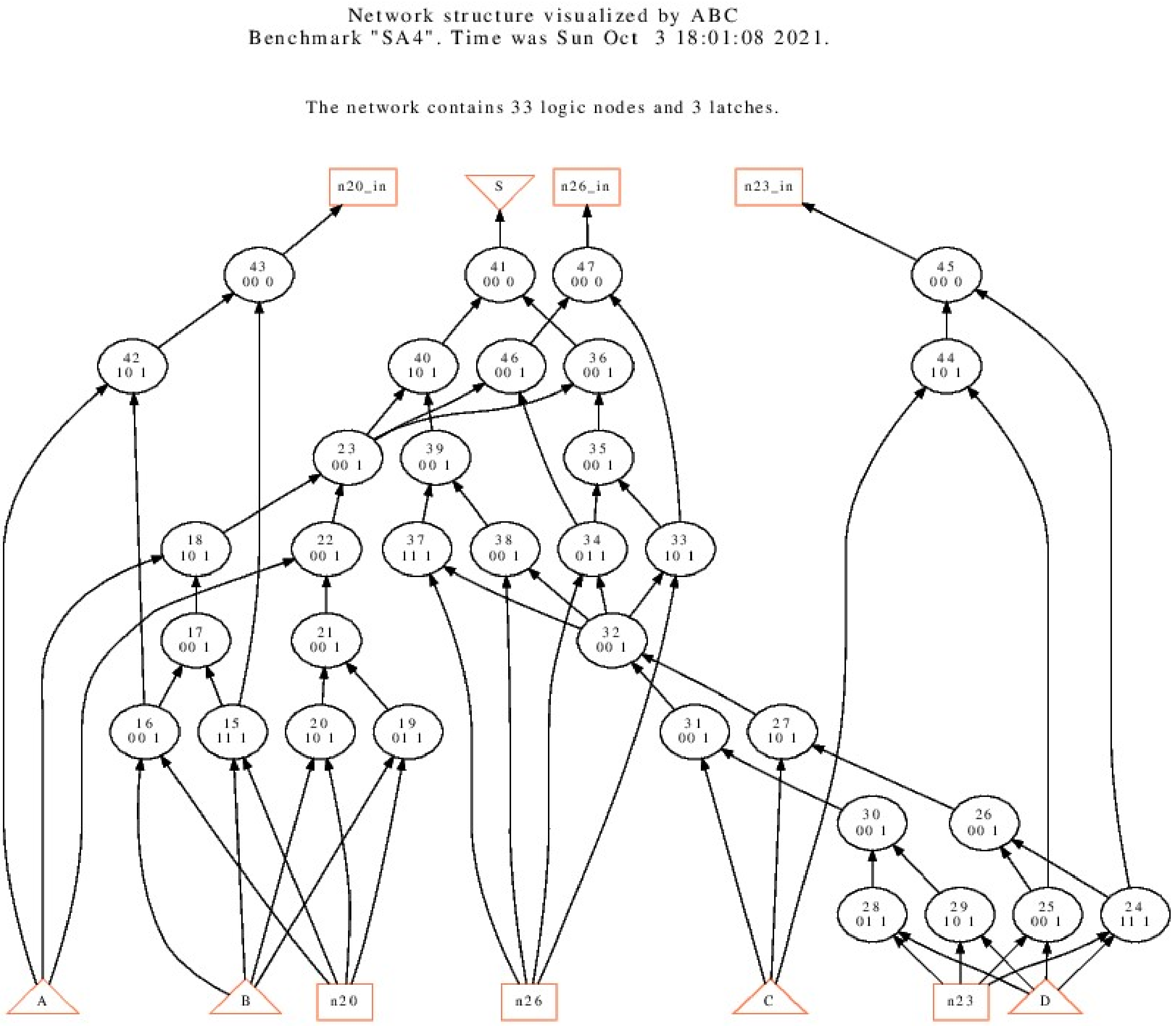


Fig 6. Network after applying abc commands