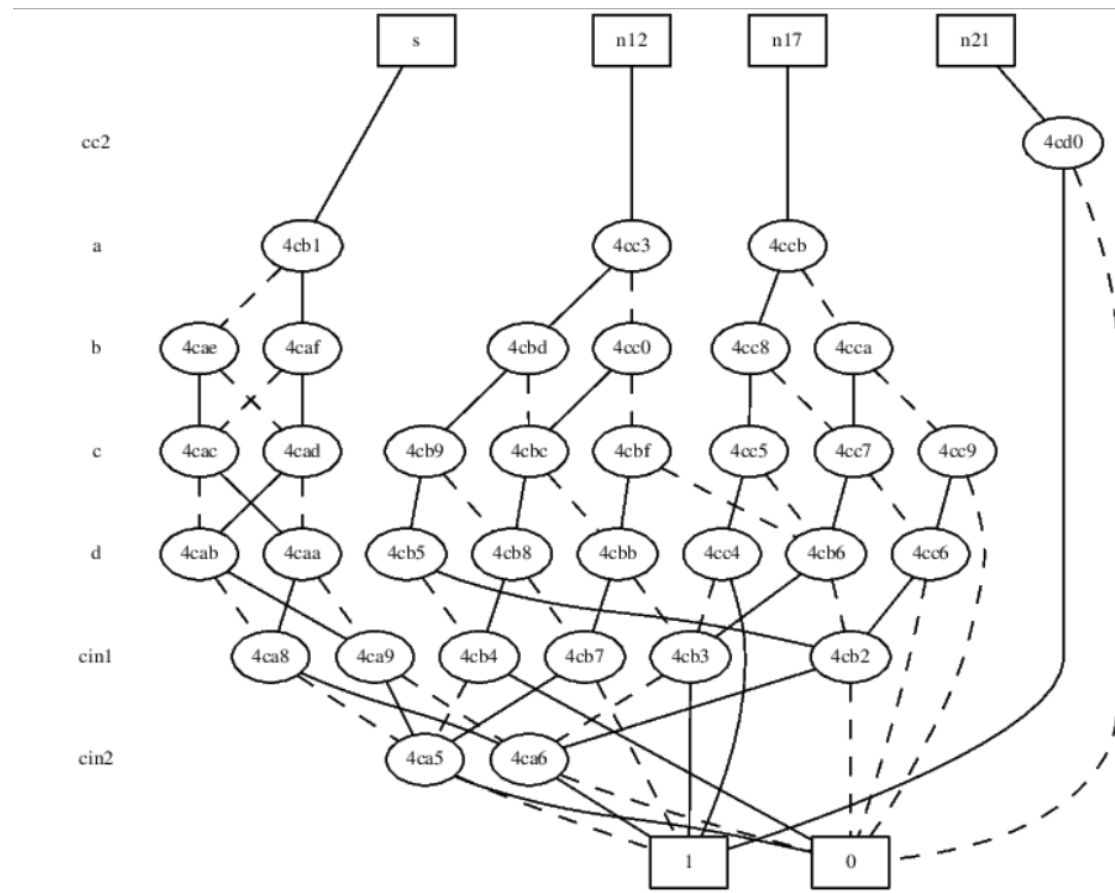


LSV PA1

PART I

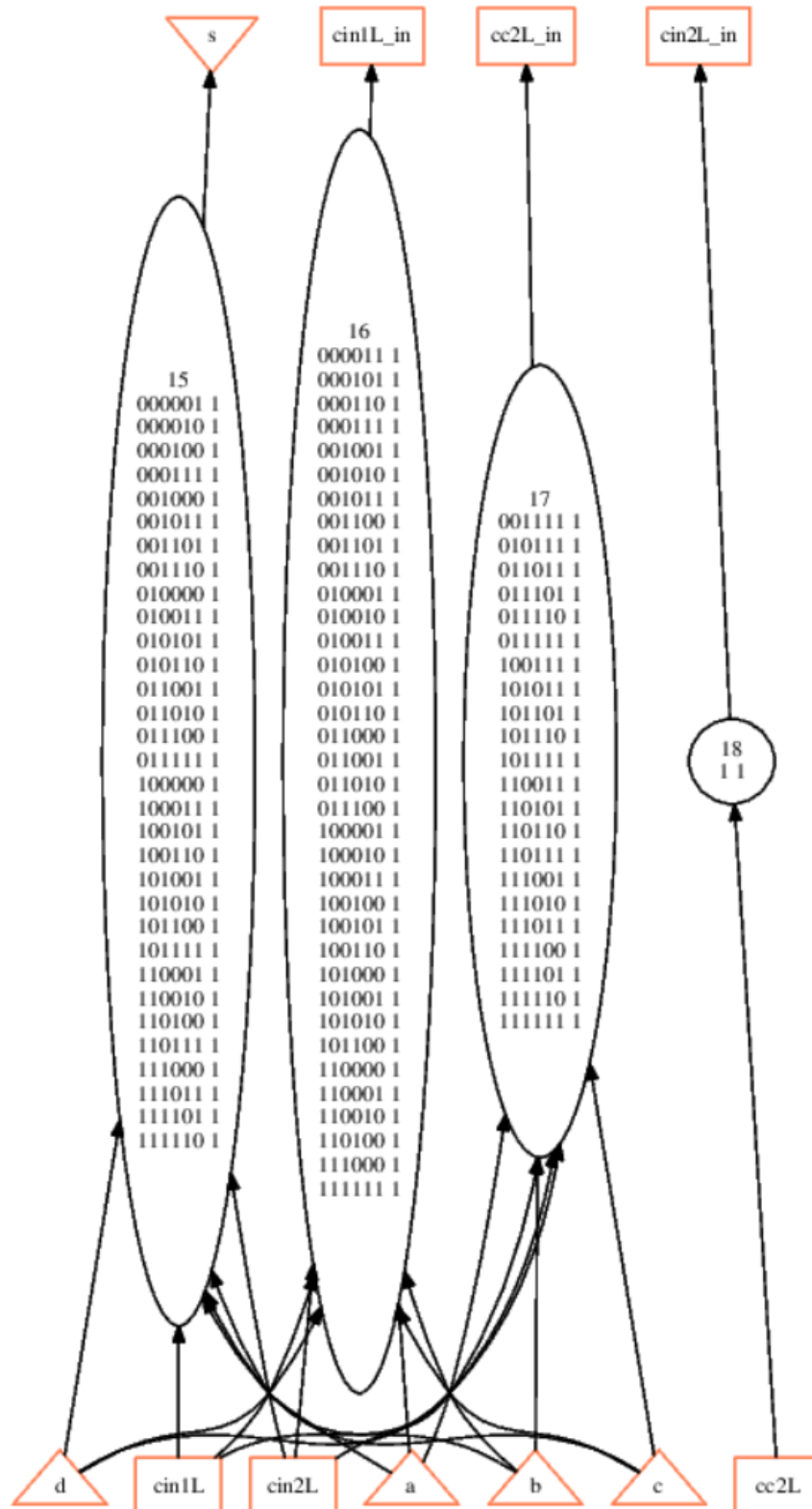
BDD:



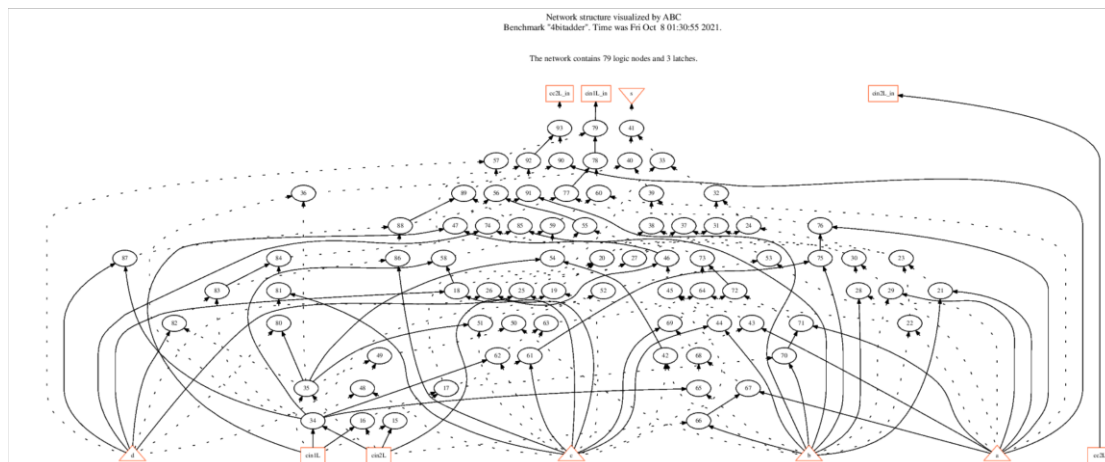
BLIF:

Network structure visualized by ABC
 Benchmark "4bitadder". Time was Fri Oct 8 01:28:38 2021.

The network contains 4 logic nodes and 3 latches.



AIG:



PART II

(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**)

A: For logic network AIG, it simply combines the onset terms into a more general form compared to the original .blif. Ex: 0011 1 and 0010 1 would be combined into 001- 1. The **strash** command however transform blif into a aig graph with one-level structural hashing, checking if there is any AND gate with same inputs identical to the new AND gate. Also the number of latches will not change while the number of nodes increases 20x (4 -> 79) of the structural hashed AIG.

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

A: For the graph of bdd, it is identical to the command aig, as for the collapse command, we can see a BDD graph while the left column is the nodes and the bottom are logic 0 and 1.

(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).

A: Two answers I'm not sure which is better. Command **logic** : from "usage" in abc.c , it says "transforms an AIG into a logic network with SOPs". The other is command **renode**: "transforms the AIG into a logic network with larger nodes, while minimizing the number of FF literals of the node SOPs." Seems like that **renode** is better compared to **logic**.