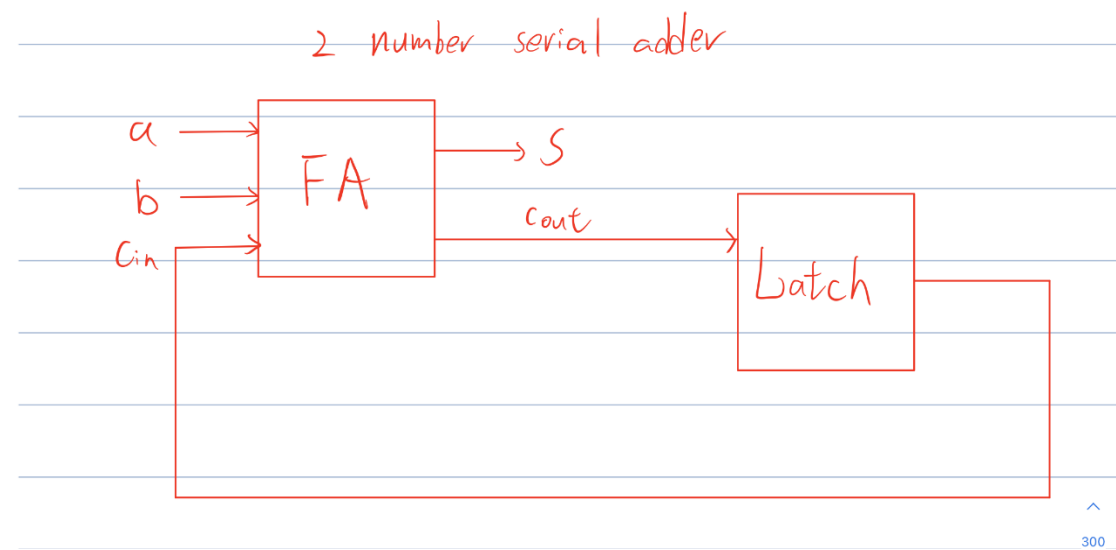
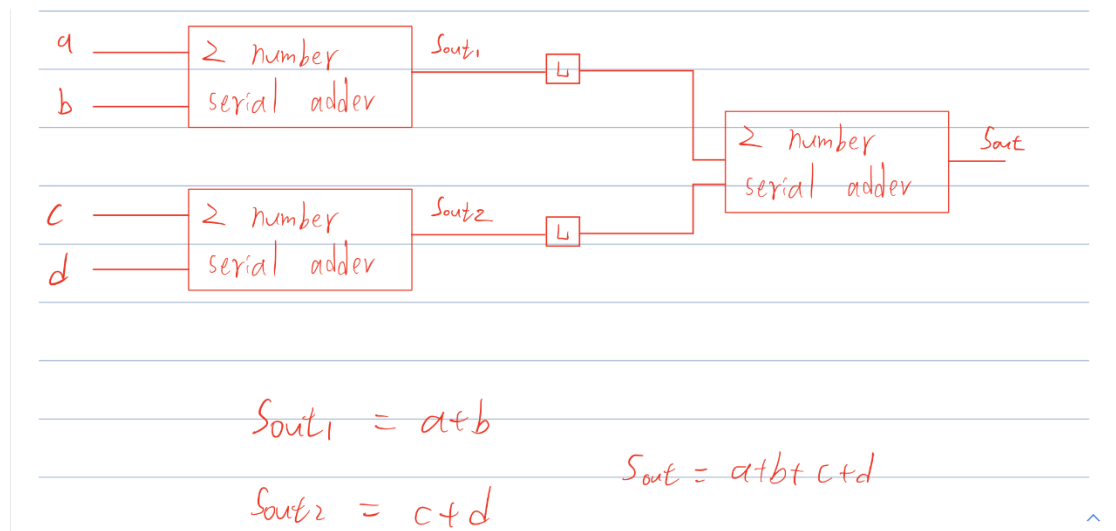


→ 、 Four number serial adder circuit

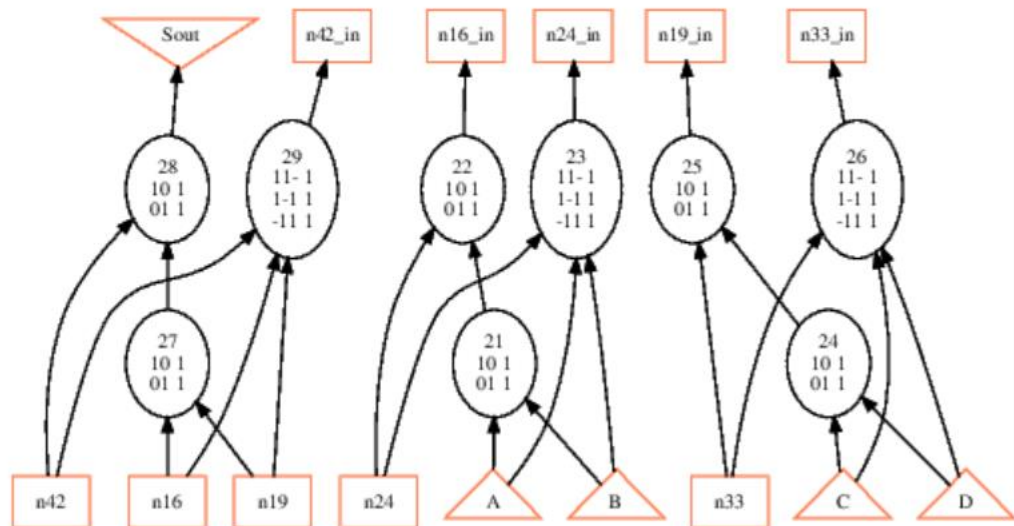


二、 Using abc

1. Command 'show'

Network structure visualized by ABC
Benchmark "four_number_serial_adder". Time was Wed Oct 6 18:33:02 2021.

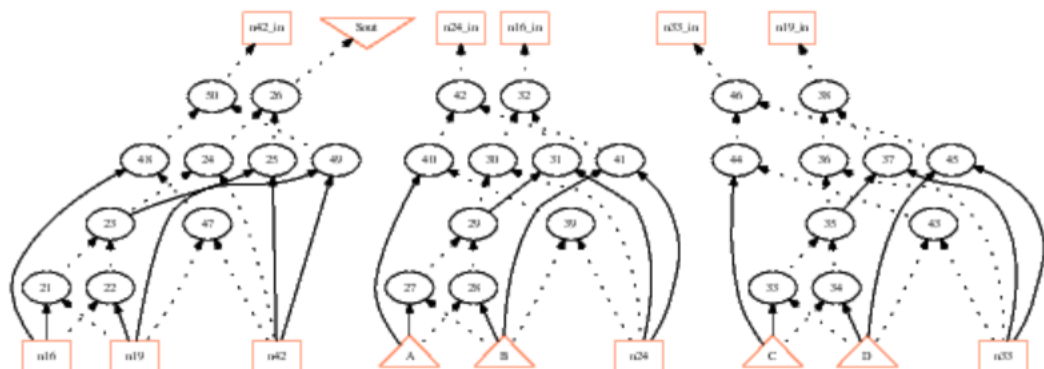
The network contains 9 logic nodes and 5 latches.



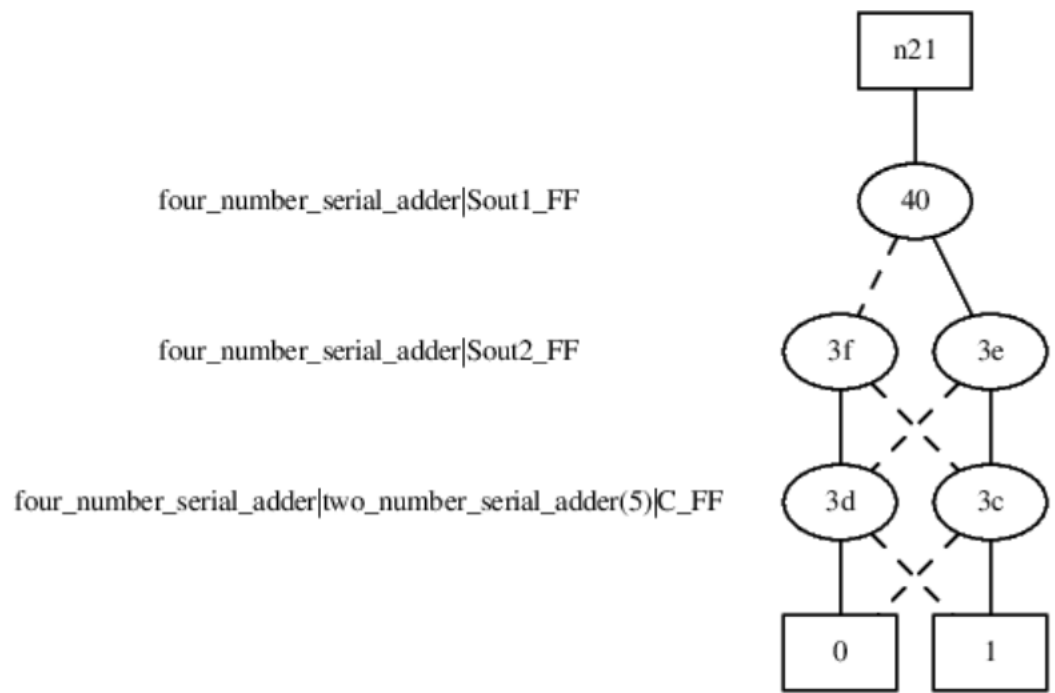
2. Command 'strash' & Command 'show'

Network structure visualized by ABC
Benchmark "four_number_serial_adder". Time was Wed Oct 6 18:33:53 2021.

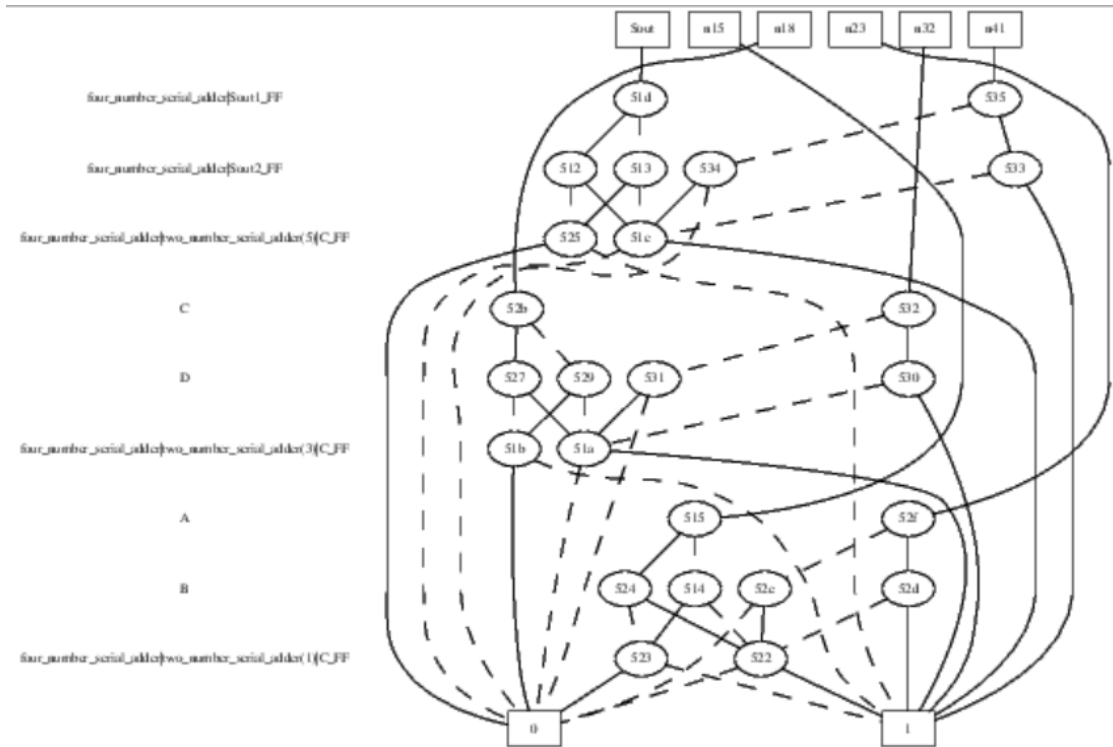
The network contains 30 logic nodes and 5 latches.



3. Command 'collapse' & Command 'show_bdd'



4. Command 'collapse' & Command 'show_bdd -g'



三、ABC Boolean Function Representations

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

1. logic network in AIG vs. structurally hashed AIG

aig : Converts local functions of the nodes to AIGs.

Strash : Transforms the current network into an AIG by one-level structural hashing. The resulting AIG is a logic network composed of two-input AND gates and inverters represented as complemented attributes on the edges. Structural hashing is a purely combinational transformation, which does not modify the number and positions of latches.

2. logic network in BDD vs. collapsed BDD

bdd : Converts local functions of the nodes to BDDs.

Collapse : Recursively composes the fanin nodes into the fanout nodes resulting in a network, in which each CO is produced by a node, whose fanins are CIs. Collapsing is performed by building global functions using BDDs and is, therefore, limited to relatively small circuits. After collapsing, the node functions are represented using BDDs.

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

Using command 'logic' can do the converting.

logic : Transforms the AIG into a logic network with the SOP representation of the two-input AND-gates.