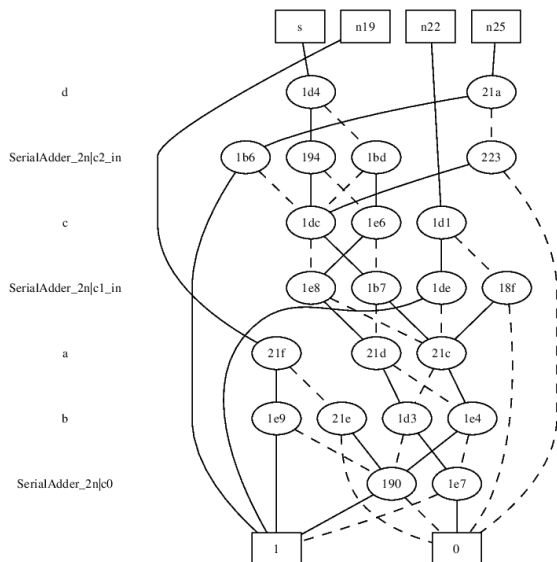
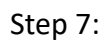
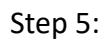


Step 3:



Part 2

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

1. aig vs. strash

aig – Converts local functions of the nodes to AIGs (representation data structure).

strash – Transforms the current network into an global AIG by one-level structural hashing. Structural hashing is a purely combinational transformation, which does not modify the number and positions of latches.

```
abc 09> read /home/frankchiang/course/lsv/LSV-PA/lsv_fall_2021/pa1/sa4.blif
Hierarchy reader flattened 3 instances of logic boxes and left 0 black boxes.
abc 10> ps
SerialAdder_2n      : i/o = 4/ 1 lat = 3 nd = 6 edge = 18 cube = 21 lev = 3
abc 10> aig
SerialAdder_2n      : i/o = 4/ 1 lat = 3 nd = 6 edge = 18 aig = 39 lev = 3
abc 10> strash
abc 11> ps
SerialAdder_2n      : i/o = 4/ 1 lat = 3 and = 33 lev = 8
```

2. bdd vs. collapse

bdd – Converts local functions of the nodes to BDDs.

collapse – Transforms the current network into an global BDD. It 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.

```
abc 11> read /home/frankchiang/course/lsv/LSV-PA/lsv_fall_2021/pa1/sa4.blif
Hierarchy reader flattened 3 instances of logic boxes and left 0 black boxes.
abc 12> ps
SerialAdder_2n      : i/o = 4/ 1 lat = 3 nd = 6 edge = 18 cube = 21 lev = 3
abc 12> bdd
SerialAdder_2n      : i/o = 4/ 1 lat = 3 nd = 6 edge = 18 bdd = 21 lev = 3
abc 12> collapse
abc 13> ps
SerialAdder_2n      : i/o = 4/ 1 lat = 3 nd = 4 edge = 22 bdd = 25 lev = 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).

\$ logic

\$ sop