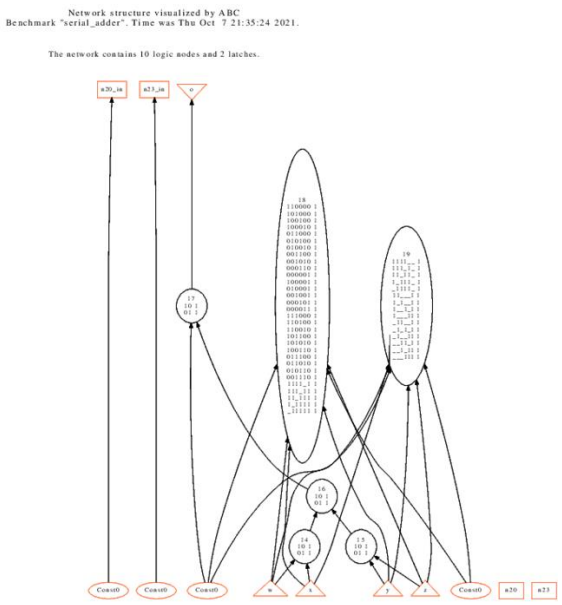
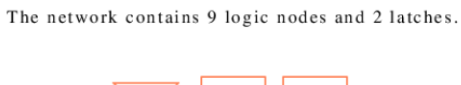
HW1

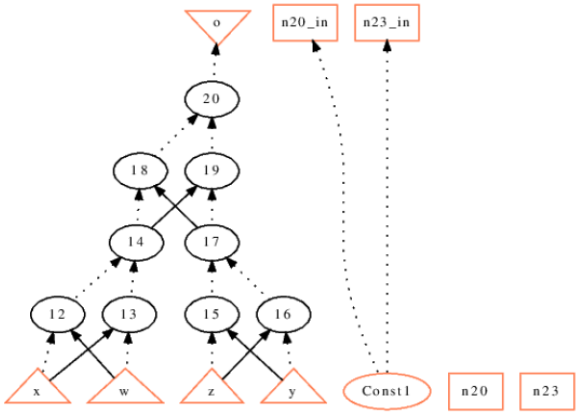
1.

After step3

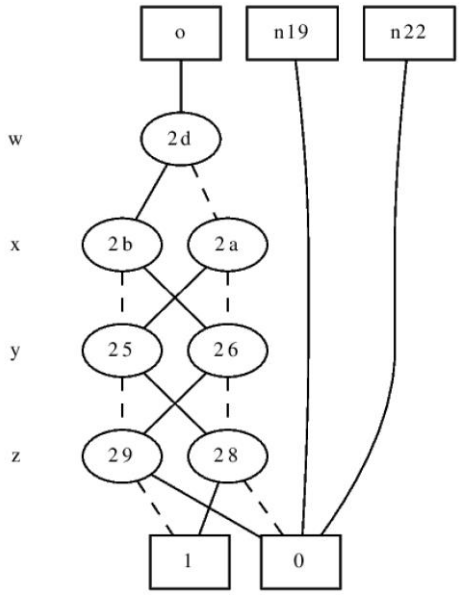


After step5





After step7



2.

(a)

Strash vs aig

```
abc 02> print_stats
serial_adder               : i/o =    4/    1  lat =    2  nd =    10  edge =    20  aig  =    66  lev = 3
abc 02> strash
abc 03> print_stats
serial_adder               : i/o =    4/    1  lat =    2  and =     9  lev =  4
abc 03>
```

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.

(One-level strashing: When a new AND-gate is added, checks is performed for a node with the same fanins (up to permutation of gate inputs).)

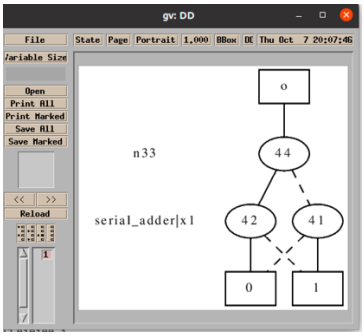
The number of gates and levels are different, while the number of latches remains the same.

bdd vs collapse

```
abc 02> print_stats
serial_adder                : i/o =    4/    1  lat =    2  nd =    10  edge
=     20  bdd  =    35  lev = 3
abc 02> collapse
abc 03> print_stats
serial_adder                : i/o =    4/    1  lat =    2  nd =     3  edge
=      4  bdd  =     4  lev = 1
abc 03>
```

bdd : Converts local functions of the nodes to BDD.

Ex: Single logic node x1 converted to BDD.



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)

renode , sop.