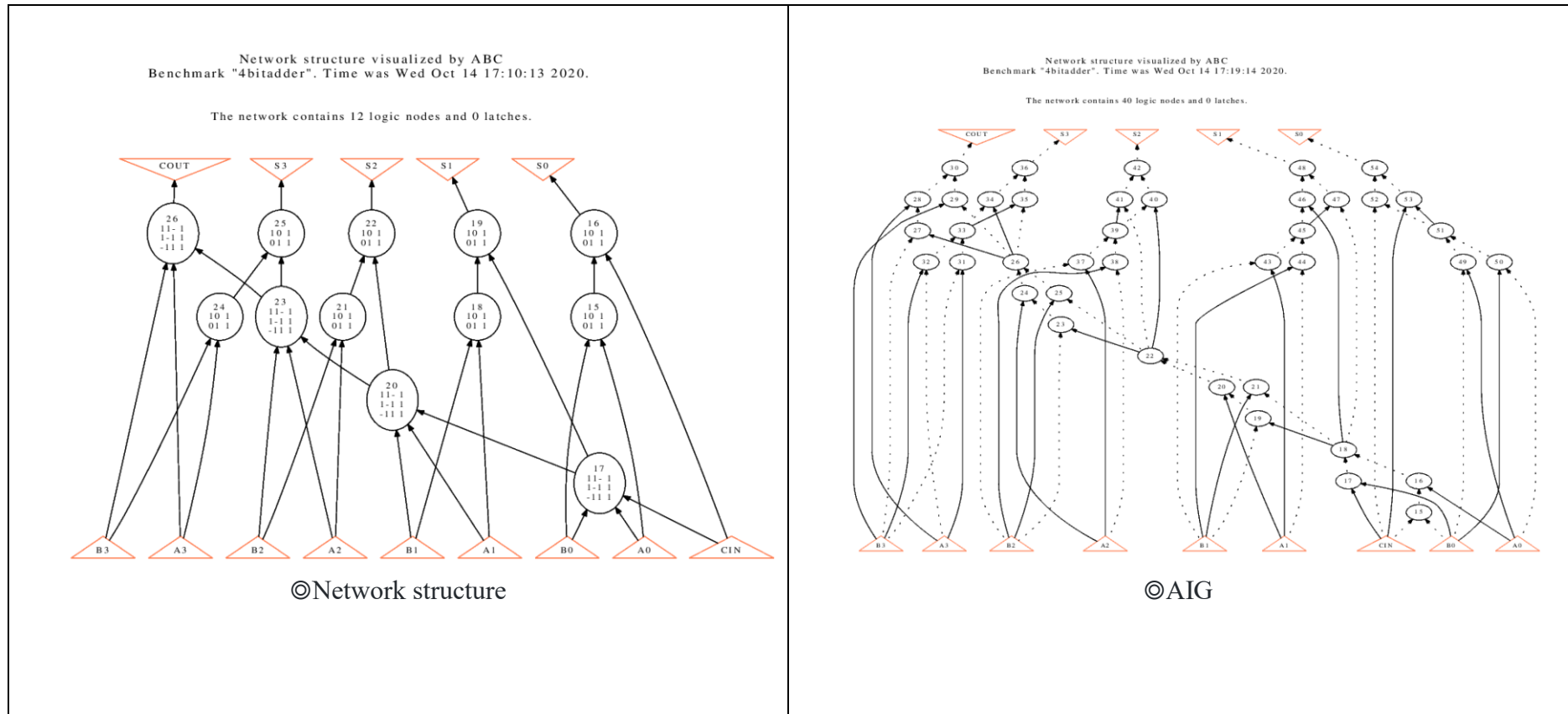


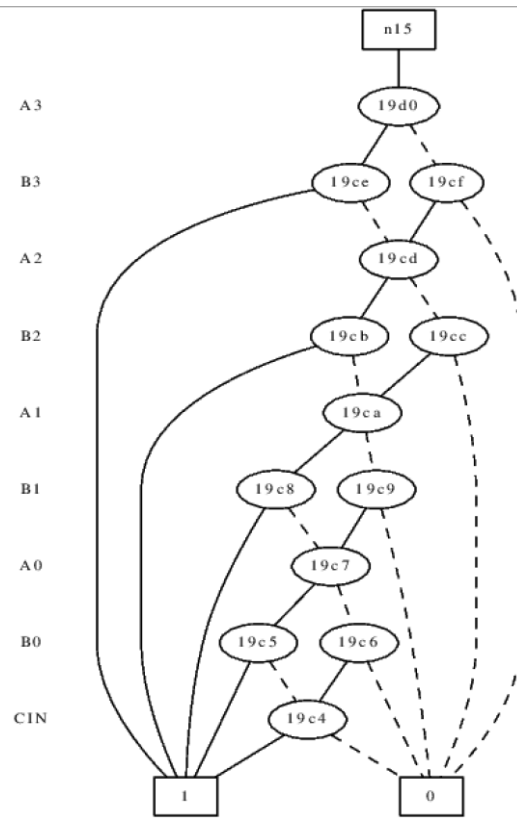
1. [Using ABC] (10%)

(a) Use [BLIF manual](#) to create a BLIF file representing a four-bit adder.

```
4bitadder.blif
1.model 4bitadder
2.inputs A3 A2 A1 A0 B3 B2 B1 B0 CIN
3.outputs COUT S3 S2 S1 S0
4.subckt fulladder a=A0 b=B0 cin=CIN s=S0 cout=CARRY1
5.subckt fulladder a=A1 b=B1 cin=CARRY1 s=S1 cout=CARRY2
6.subckt fulladder a=A2 b=B2 cin=CARRY2 s=S2 cout=CARRY3
7.subckt fulladder a=A3 b=B3 cin=CARRY3 s=S3 cout=COUT
8.end
9
10.model fulladder
11.inputs a b cin
12.outputs s cout
13.names a b k
14 10 1
15 01 1
16.names k cin s
17 10 1
18 01 1
19.names a b cin cout
20 11- 1
21 1-1 1
22 -11 1
23.end
```

(b) Results of show and show_bdd.:





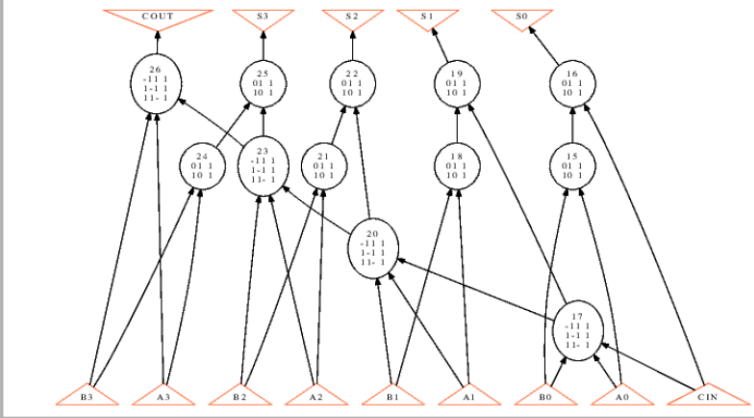
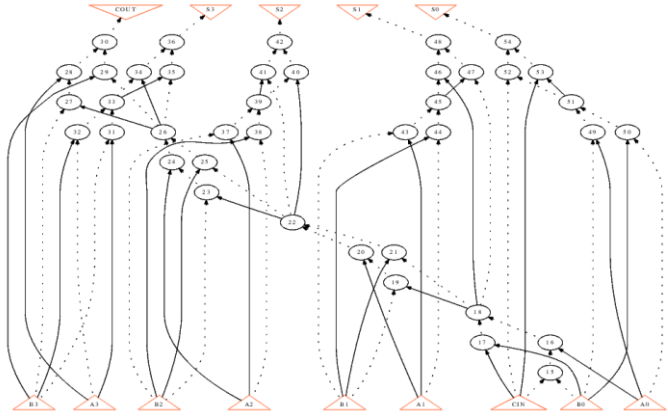
©BDD

2. [ABC Boolean Function Representations] (10%)

In ABC there are different ways to represent Boolean functions.

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

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

AIG (aig)(Status and Result)	Structurally hashed AIG (strash)(Status and Result)
<p>Network structure visualized by ABC Benchmark "4bitadder". Time was Wed Oct 14 17:48:21 2020.</p> <p>The network contains 12 logic nodes and 0 latches.</p>  <p>i/o = 9/ 5 lat = 0 nd = 12 edge = 28 aig = 40 lev = 4</p> <p>Converts node functions to AIG (just convert node to AIG)</p>	<p>Network structure visualized by ABC Benchmark "4bitadder". Time was Wed Oct 14 17:52:29 2020.</p> <p>The network contains 40 logic nodes and 0 latches.</p>  <p>i/o = 9/ 5 lat = 0 and = 40 lev = 12</p> <p>Transforms combinational logic into an AIG (Convert all logic network to AIG)</p>

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

BDD (bdd)(Status and Result) (only shows the first PO)	Collapsed BDD (collapse)(Status and Result)
<p>Diagram illustrating the BDD (bdd) conversion of a 4-bit adder circuit. The circuit has inputs 1 and 0, and outputs COUT, A3, B3, and 4bitadder CARRY3. The BDD structure shows nodes 1c, 1a, 1b, and 19, with edges connecting them to the outputs.</p> <p>i/o = 9/ 5 lat = 0 nd = 12 edge = 28 bdd = 32 lev = 4</p> <p>Converts node functions to BDD</p> <p>(just convert node to BDD)</p>	<p>Diagram illustrating the collapsed BDD (collapse) conversion of a 4-bit adder circuit. The circuit has inputs 1 and 0, and outputs COUT, A3, B3, and 4bitadder CARRY3. The collapsed BDD structure shows nodes 1, 0, 44, 45, 46, 47, 48, 49, 4a, 4b, 4c, 4d, 4e, 4f, 50, and n15, with edges connecting them to the outputs.</p> <p>i/o = 9/ 5 lat = 0 nd = 5 edge = 33 bdd = 43 lev = 1</p> <p>Collapses the network by constructing global BDDs</p> <p>(Convert all logic network to BDD)</p>

(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).

Use command **logic** to transforms an AIG into a logic network with SOPs.

```
abc 08> read lsv/pa1/4bitadder.blif
Hierarchy reader flattened 4 instances of logic boxes and left 0 black boxes.
abc 09> strash
abc 10> logic
```

