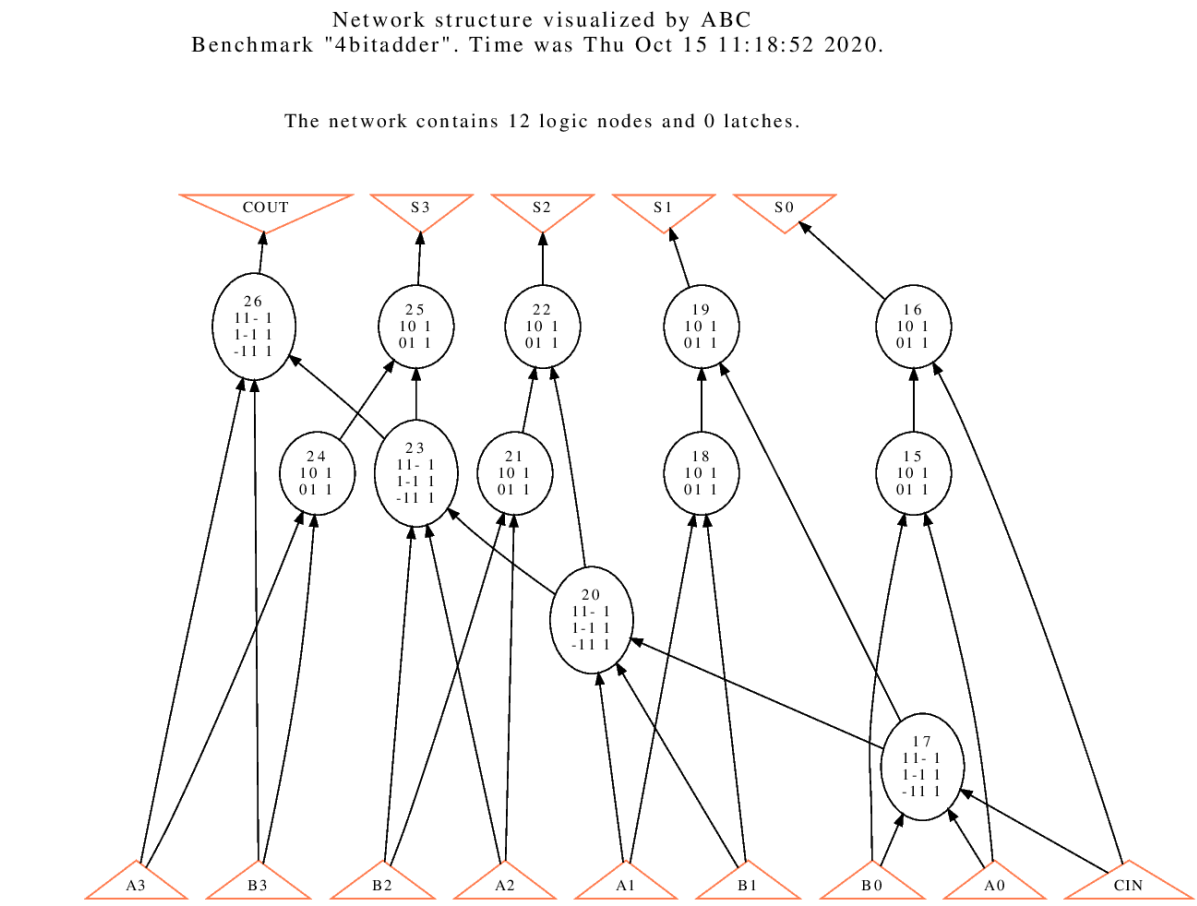
LSV PA1 電機四 黃彥翔

Part 1:

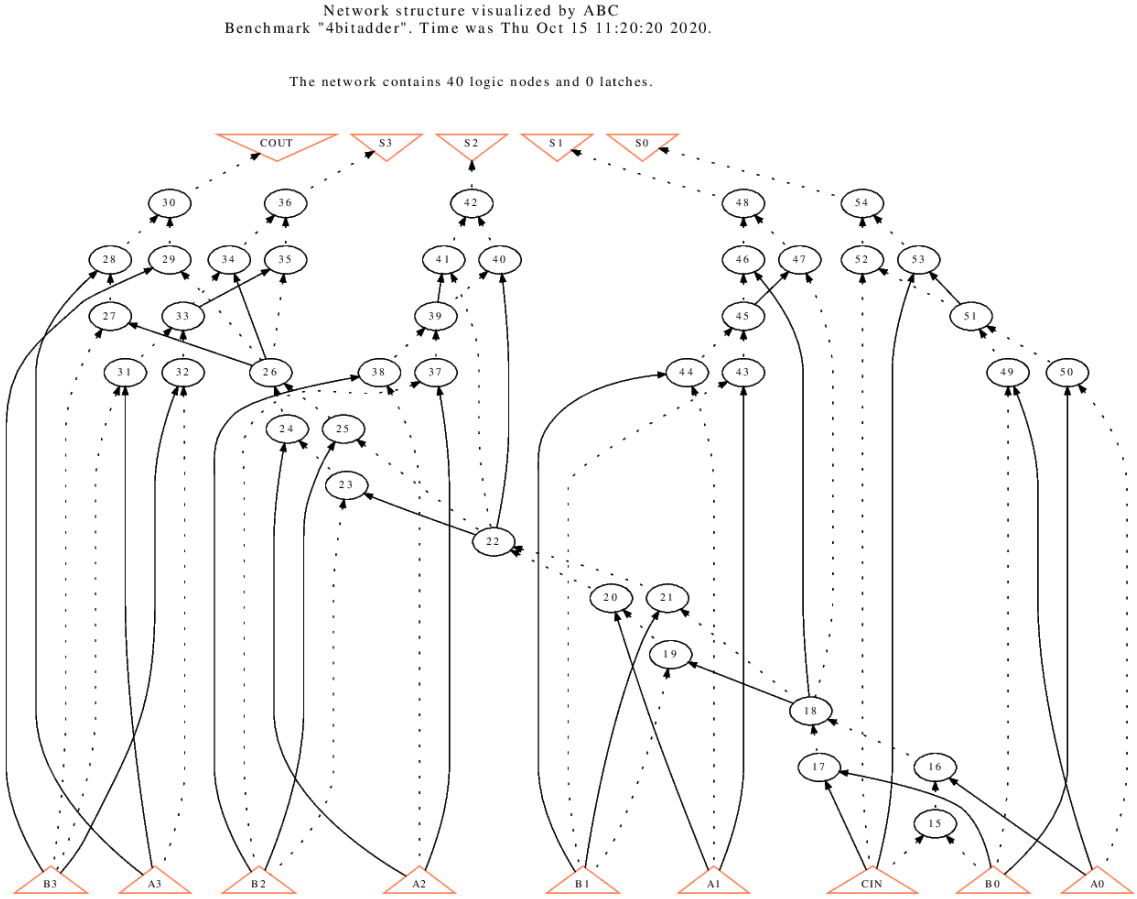
- BLIF檔案:

```
.model 4bitadder
.inputs A3 A2 A1 A0 B3 B2 B1 B0 CIN
.outputs COUT S3 S2 S1 S0
.subckt fulladder a=A0 b=B0 cin=CIN s=S0 cout=CARRY1
.subckt fulladder a=A1 b=B1 cin=CARRY1 s=S1 cout=CARRY2
.subckt fulladder a=A2 b=B2 cin=CARRY2 s=S2 cout=CARRY3
.subckt fulladder a=A3 b=B3 cin=CARRY3 s=S3 cout=COUT
.end
.model fulladder
.inputs a b cin
.outputs s cout
.names a b k
10 1
01 1
.names k cin s
10 1
01 1
.names a b cin cout
11- 1
1-1 1
-11 1
.end
```

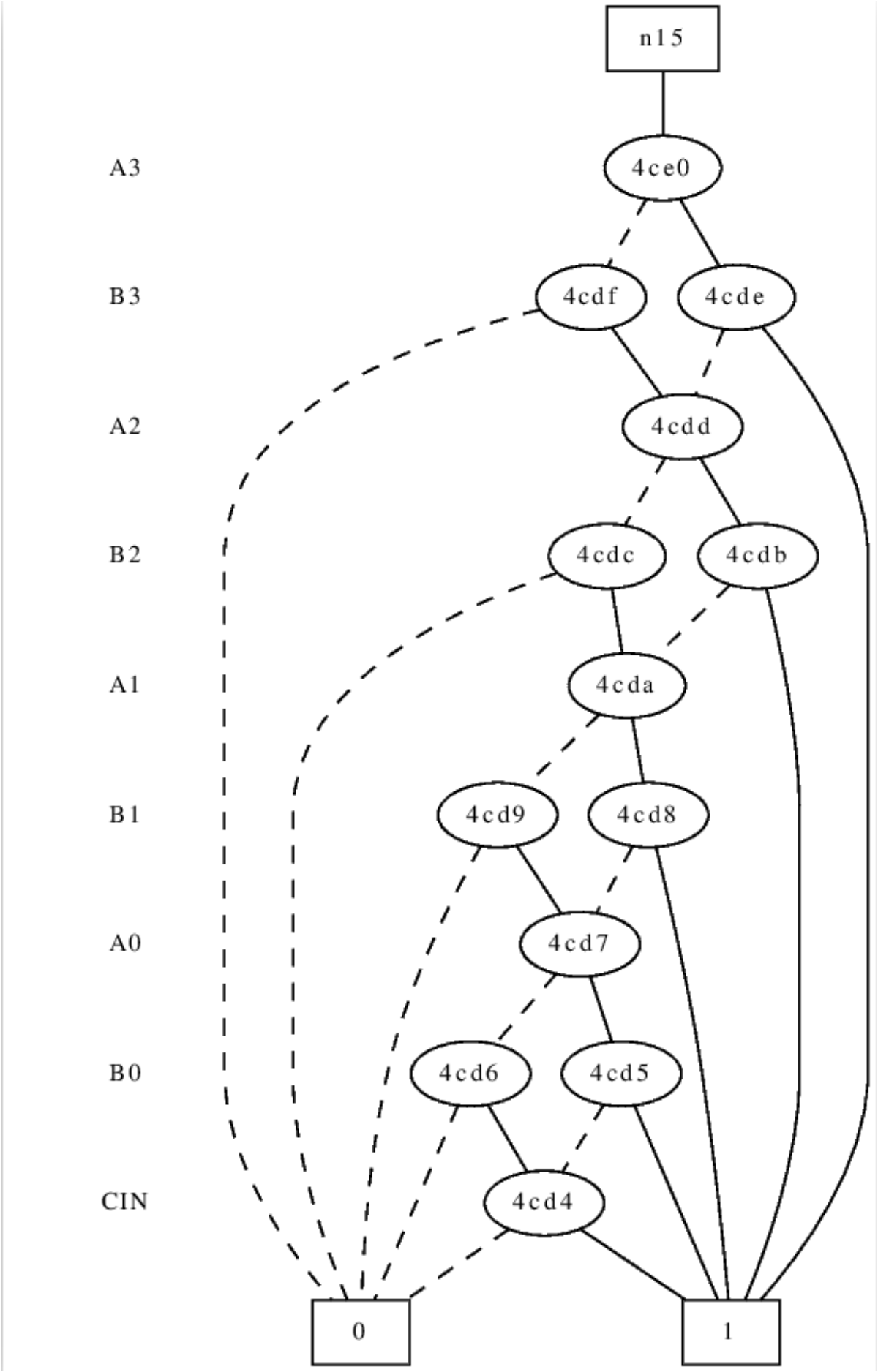
- Results of `show` and `show_bdd`:
  - the first `show`



- the second `show`



- `show_bdd`


n15
A3
4ce0
B3
4cdf
4cde
A2
4cdd
B2
4cdc
4cdb
A1
4cda
B1
4cd9
4cd8
A0
4cd7
B0
4cd6
4cd5
CIN
4cd4
0
1

PART 2:

- Compare the following differences with the four-bit adder example.
  - `print_stats` can easily see the difference:

```
(base) yen-hsiang@yenhsiang-MS-7B89:~/Desktop/LSV-PA$ ./abc
UC Berkeley, ABC 1.01 (compiled Oct 14 2020 22:37:15)
abc 01> read adder.blif
Warning: Constant-0 drivers added to 4 non-driven nets in network "4bitadder":
S3, S2, S1, A4 ...
Hierarchy reader flattened 4 instances of logic boxes and left 0 black boxes.
abc 02> aig
abc 02> print_stats
4bitadder                 : i/o =    9/    5  lat =    0  nd =    16  edge =    28  aig  =    40  lev = 4
abc 02> strash
abc 03> print_stats
4bitadder                 : i/o =    9/    5  lat =    0  and =    19  lev = 10
abc 03> bdd
Error: Converting to BDD is possible only for logic networks.
abc 03>
abc 03> q
(base) yen-hsiang@yenhsiang-MS-7B89:~/Desktop/LSV-PA$ ./abc
UC Berkeley, ABC 1.01 (compiled Oct 14 2020 22:37:15)
abc 01> read adder.blif
Warning: Constant-0 drivers added to 4 non-driven nets in network "4bitadder":
S3, S2, S1, A4 ...
Hierarchy reader flattened 4 instances of logic boxes and left 0 black boxes.
abc 02> bdd
abc 02> print_stats
4bitadder                 : i/o =    9/    5  lat =    0  nd =    16  edge =    28  bdd  =    32  lev = 4
abc 02> collapse
abc 03> print_stats
4bitadder                 : i/o =    9/    5  lat =    0  nd =     5  edge =    11  bdd  =    14  lev = 1
abc 03>
```

  - `command summary` in https://people.eecs.berkeley.edu/~alanmi/abc/ says that
    - **`aig`** – Converts local functions of the nodes to AIGs.
    - **`strash`** – Transforms the current network into an AIG by one-level structural hashing.
    - **`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.

    It means that:

    1. the difference between **`aig`** and **`strash`** :

       `aig` only convert local functions of the nodes to **AIGs**, so the circuit remains a **logic network**.

       However, **`strash`** will convert the entire circuit to **AIGs**.

    2. the difference between **`bdd`** and **`collapse`** :

       `bdd` only convert local functions of the nodes to **BDDs**, so the circuit remains a **logic network**.

       However, **`collapse`** will convert the entire circuit to **BDDs**.

- 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`