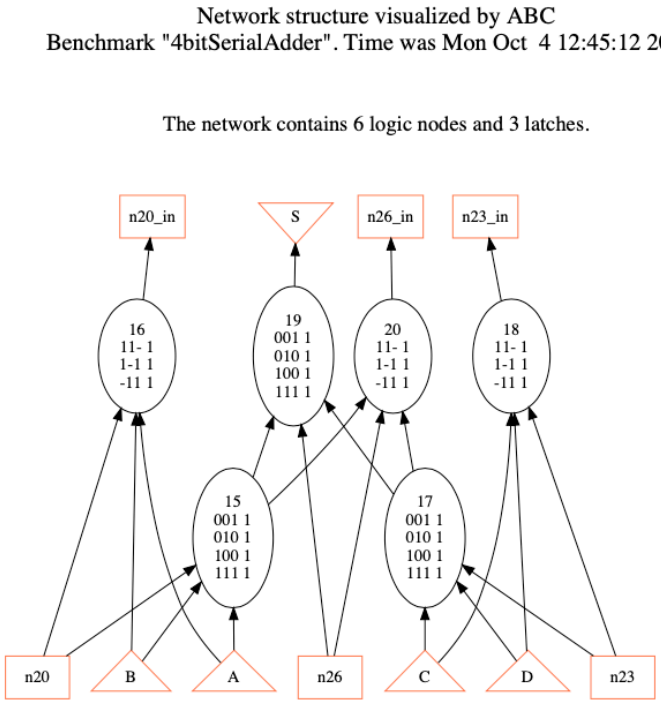
Logic Synthesis & Verification PA1 Report

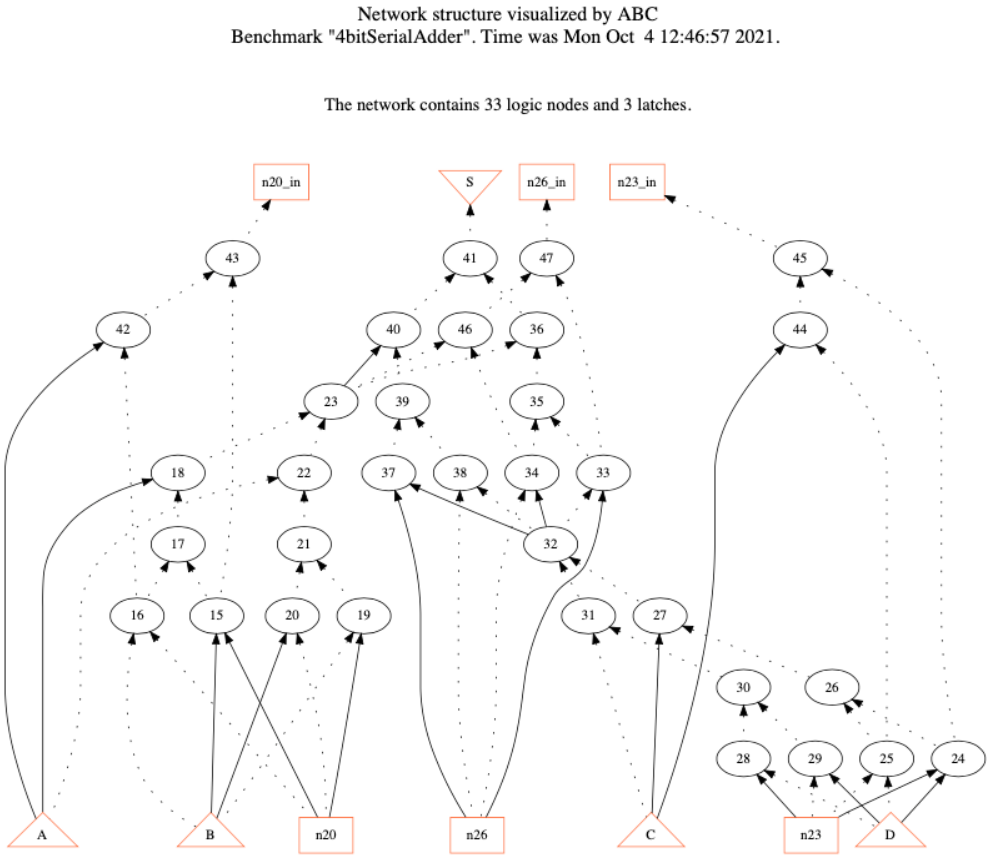
R10943100 林聖亞

1. [Using ABC]

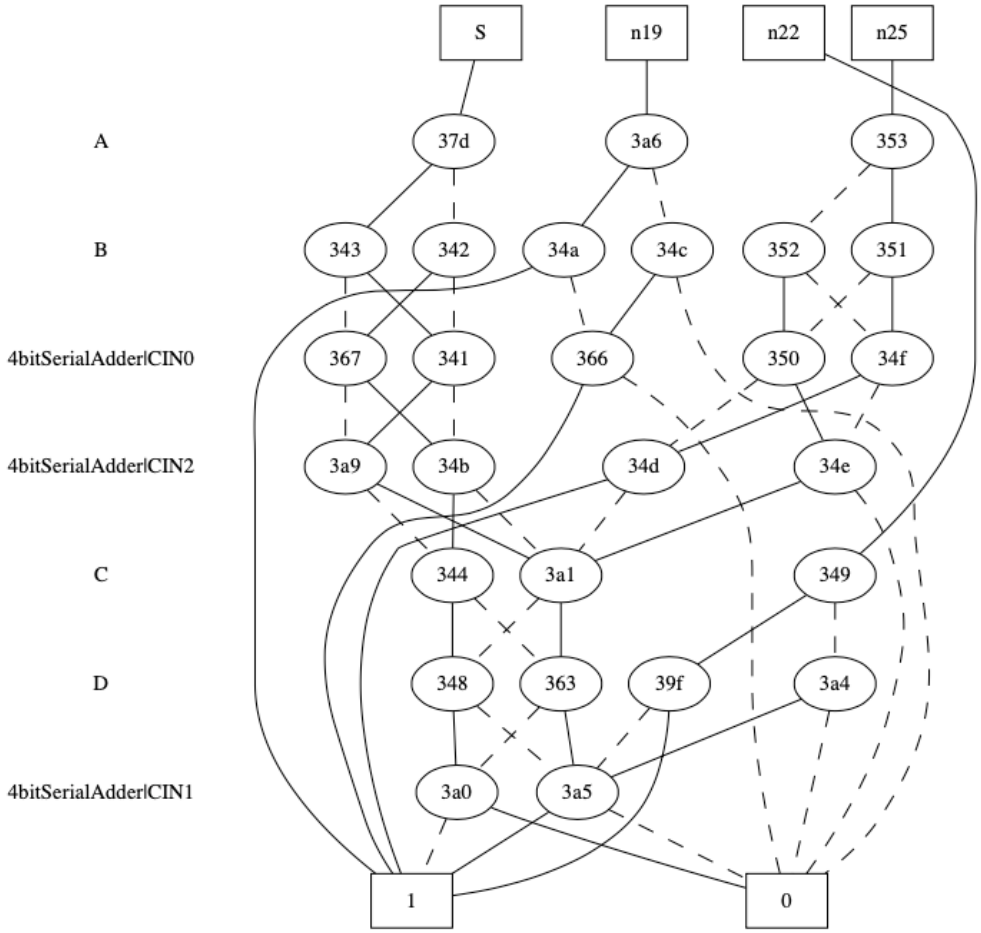
Initial network (result of "show" after step 3):



AIG network (result of "show" after step 5):



BDD representation (result of "`show_bdd -g`" after step 7):



2. [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

   The command "`aig`" converts the representation of the node function to AIG, while the whole network is still a logic network. However, the command "`strash`" converts the entire network into AIG.

   Here's an example with the four-number serial adder:

   "`print_stats`" for original network:

   4bitSerialAdder: i/o = 4/1 lat = 3 nd = 6 edge = 18 cube = 21 lev = 2

   "`print_stats`" for logic network in AIG:

   4bitSerialAdder: i/o = 4/1 lat = 3 nd = 8 edge = 18 aig = 39 lev = 2

   "`print_stats`" for structurally hashed AIG:

   4bitSerialAdder: i/o = 4/1 lat = 3 and = 33 lev = 8

2. logic network in BDD vs. collapsed BDD

   The command "`bdd`" converts the representation of the node function to BDD, while the whole network is still a logic network. However, the command "`collapsed`" converts the entire network into a BDD.

Here's an example with the four-number serial adder:

"`print_stats`" for original network:

```
4bitSerialAdder: i/o = 4/1    lat = 3    nd = 6    edge = 18    cube = 21    lev = 2
```

"`print_stats`" for logic network in BDD:

```
4bitSerialAdder: i/o = 4/1    lat = 3    nd = 6    edge = 18    bdd    = 21    lev = 2
```

"`print_stats`" for collapsed BDD:

```
4bitSerialAdder: i/o = 4/1    lat = 3    nd = 4    edge = 20    bdd    = 25    lev = 1
```

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

```
> logic
> sop
```