R12943159 陳昱揚

2 (b)

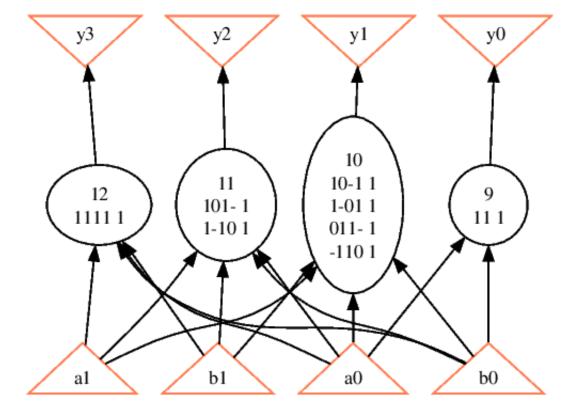
• commands screenshot:

```
UC Berkeley, ABC 1.01 (compiled Sep 11 2023 11:53:08)
     write_dot mul.dot
show_bdd
source -s abc.rc
read lsv/pa1/mul.blif
print_stats
strash
show
collapse
show_bdd -g
abc 01> read lsv/pa1/mul.blif
abc 02> print_stats
                         : i/o = 4/ 4 lat = 0 nd = 4 edge = 14 cube =
                                                                                    8 	ext{ lev} = 1
abc 02> show
abc 02> strash
abc 03> show
abc 03> collapse
abc 04> show_bdd -g
abc 04> quit
```

• abc 02> show screenshot (before strash):

Network structure visualized by ABC Benchmark "mul". Time was Tue Sep 12 15:46:10 2023.

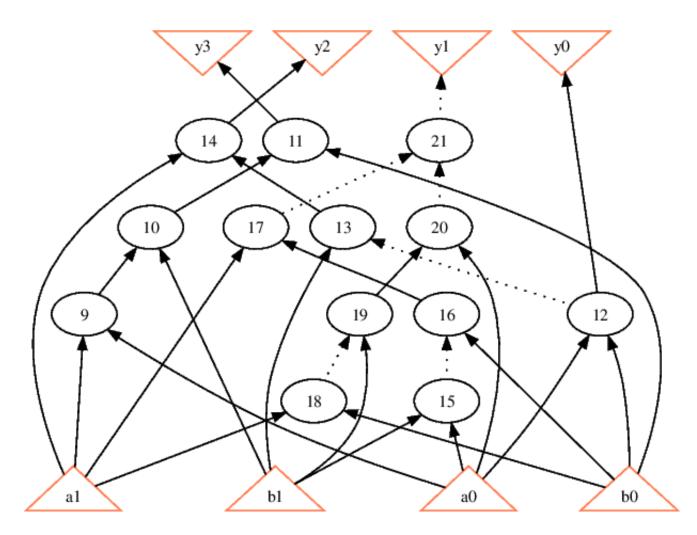
The network contains 4 logic nodes and 0 latches.



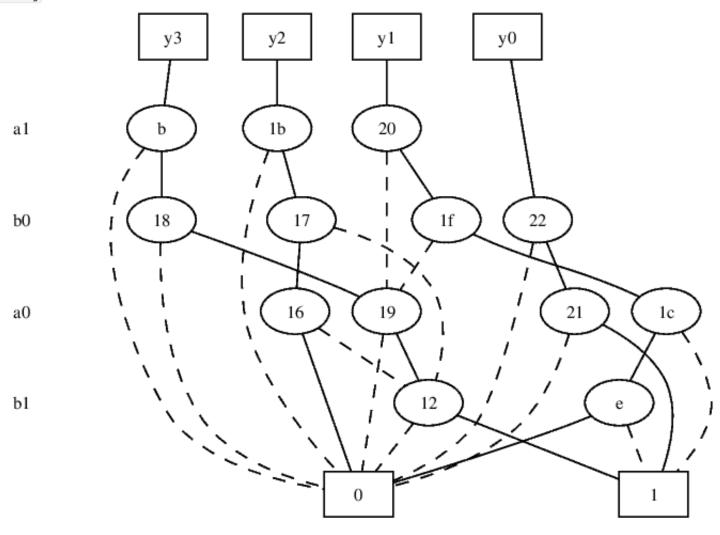
• abc 03> show screenshot (after strash):

Network structure visualized by ABC Benchmark "mul". Time was Tue Sep 12 15:46:49 2023.

The network contains 13 logic nodes and 0 latches.



• abc 04> show_bdd -g screenshot:

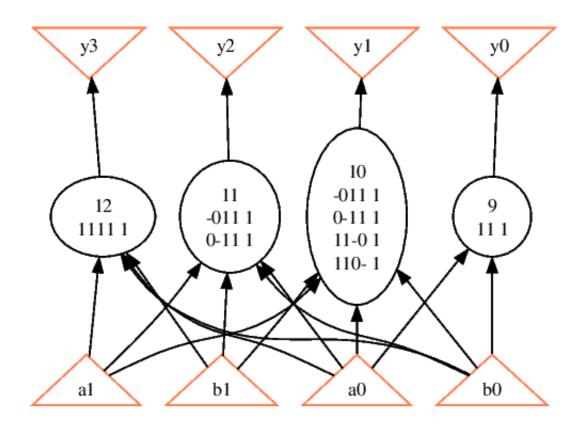


3 (a)

1. AIG after aig:

Network structure visualized by ABC Benchmark "mul". Time was Tue Sep 12 16:03:42 2023.

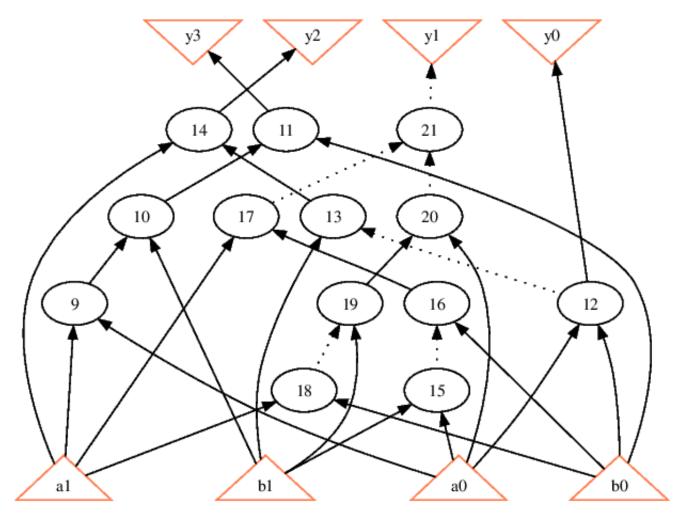
The network contains 4 logic nodes and 0 latches.



AIG after strash:

Network structure visualized by ABC Benchmark "mul". Time was Tue Sep 12 16:04:15 2023.

The network contains 13 logic nodes and 0 latches.

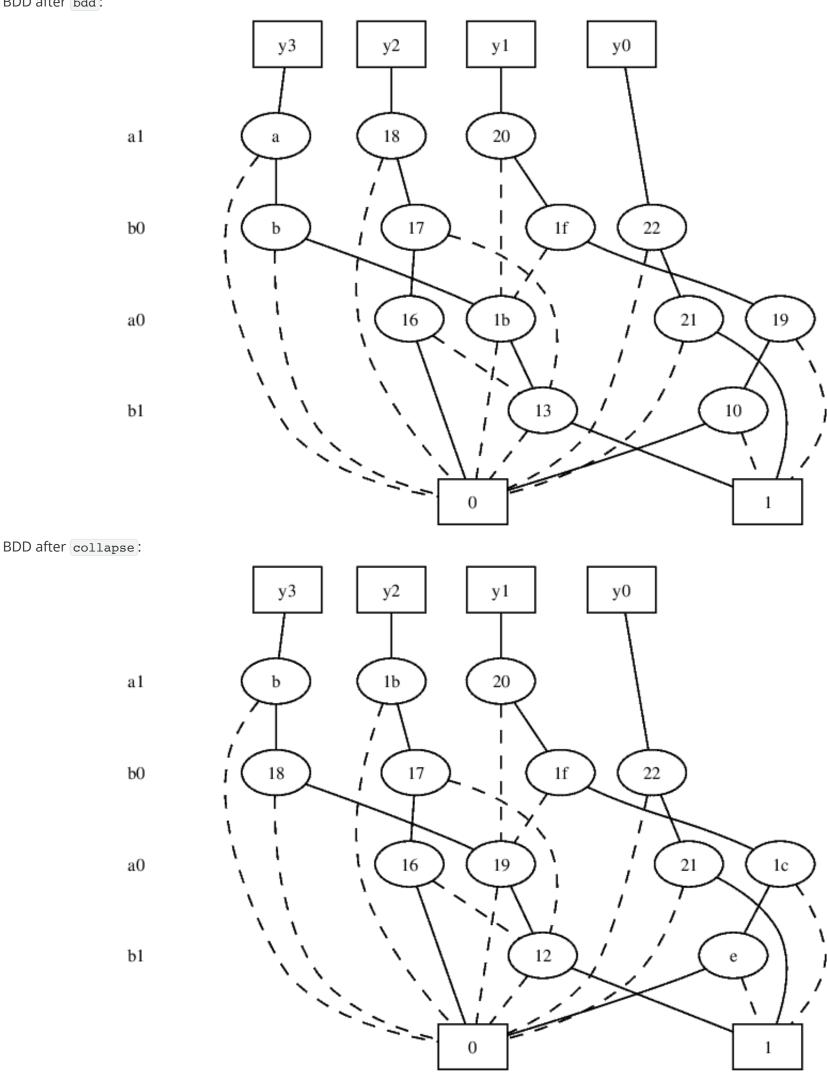


Observation: The AIG graph after aig is still a boolean network, so I suppose that Abc cannot show an AIG graph before structural hashing. To compare the AIG using aig and strash, I turned to observe the information generated by print_stats. The results can be seen below:

```
abc 01> read lsv/pa1/mul.blif
abc 02> aig
abc 02> print_stats
                                                                                                    14 	 lev = 1
mul
                              i/o =
                                               4 lat =
                                                           0 \quad nd =
                                                                       4 \text{ edge} =
                                                                                     14 aig =
abc 02> read lsv/pa1/mul.blif
abc 03> strash
abc 04> print_stats
                                                           0 and =
                              : i/o =
                                         4/
                                               4 lat =
                                                                        13 lev = 4
mul
```

Most of the information is different, but both of the print_stats have the information of level. The levels of aig and strash are different, so I suppose that strash does modification to AIG. However, I do not have the enough information to analyze whether strash had simplified the AIG or not.

2. BDD after bdd:



Observation: The BDD graphs are the same between using bdd and collapse. However, when I print out the information of BDD using print_stats and lsv_print_nodes, I found that the structure of BDD had actually been changed. The information can be seen below:

```
abc 01> read lsv/pa1/mul.blif
abc 02> bdd
abc 02> print_stats
                             i/o = 4/
mul
                                           4 lat =
                                                          0 \text{ nd} =
                                                                     4 edge =
                                                                                   14 bdd =
                                                                                                 17 	 lev = 1
abc 02> lsv_print_nodes
Object Id = 9, name = y0
 Fanin-0: Id = 2, name = a0
 Fanin-1: Id = 4, name = b0
Object Id = 10, name = y1
 Fanin-0: Id = 1, name = a1
 Fanin-1: Id = 2, name = a0
 Fanin-2: Id = 3, name = b1
 Fanin-3: Id = 4, name = b0
Object Id = 11, name = y2
 Fanin-0: Id = 1, name = a1
 Fanin-1: Id = 2, name = a0
 Fanin-2: Id = 3, name = b1
 Fanin-3: Id = 4, name = b0
Object Id = 12, name = y3
 Fanin-0: Id = 1, name = a1
 Fanin-1: Id = 2, name = a0
 Fanin-2: Id = 3, name = b1
 Fanin-3: Id = 4, name = b0
abc 01> read lsv/pa1/mul.blif
abc 02> collapse
abc 03> print_stats
                             : i/o =
                                              4 lat =
                                                          0 nd =
                                                                     4 edge =
                                                                                   14 bdd =
                                                                                                 14 	 lev = 1
mul
abc 03> lsv_print_nodes
Object Id = 9, name = n9
 Fanin-0: Id = 1, name = a1
 Fanin-1: Id = 4, name = b0
 Fanin-2: Id = 2, name = a0
 Fanin-3: Id = 3, name = b1
Object Id = 10, name = n10
 Fanin-0: Id = 1, name = a1
  Fanin-1: Id = 4, name = b0
 Fanin-2: Id = 2, name = a0
 Fanin-3: Id = 3, name = b1
Object Id = 11, name = n11
 Fanin-0: Id = 1, name = a1
 Fanin-1: Id = 4, name = b0
 Fanin-2: Id = 2, name = a0
 Fanin-3: Id = 3, name = b1
Object Id = 12, name = n12
 Fanin-0: Id = 4, name = b0
 Fanin-1: Id = 2, name = a0
```

The number of bdd is fewer when using collapse, and the order of inputs are different between using bdd and collapse as well. As for the reason why the BDD graphs are the same, I went further to investigate the source code of show_bdd. I found that if the _g switch is added, ABC will do structure hashing first and then show it out. I believe it is the reason why the graph of show_bdd _g is not changed even after collapse. A part of source code of show_bdd is shown below:

```
if ( fGlobal )
{
    Abc_Ntk_t * pTemp = Abc_NtkIsStrash(pNtk) ? pNtk : Abc_NtkStrash(pNtk, 0, 0, 0);
    Abc_NtkShowBdd( pTemp, fCompl, fReorder );
    if ( pTemp != pNtk )
        Abc_NtkDelete( pTemp );
    return 0;
}
```

3 (b)

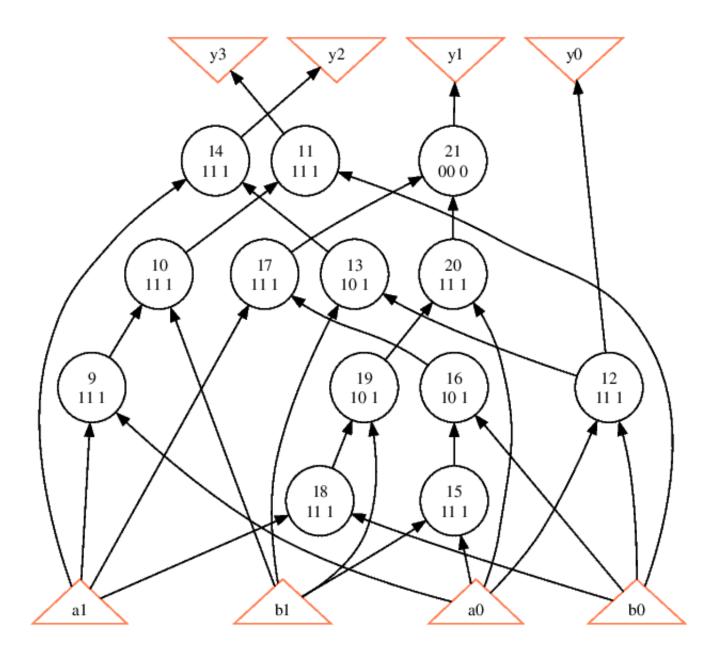
The command to convert a structurally hashed AIG to a logic network with node function expressed in SOP is logic. To generate this form of 2-bit unsigned multiplier, the following sequence of ABC commands should be executed:

```
read lsv/pal/mul.blif
strash
logic
show
```

And the result is shown below:

Network structure visualized by ABC Benchmark "mul". Time was Tue Sep 12 16:53:13 2023.

The network contains 13 logic nodes and 0 latches.



The nodes of this logic network are all expressed in SOP.