Logic Synthesis & Verification, Fall 2023

National Taiwan University

Programming Assignment 1

Name: 陳均豪

Id: r12k41016

Problem 2. [Using ABC]

Solution:

(a) two-bit unsigned multiplier Y = A * B, where A = (a1 a0) and B = (b1 b0) are two-bit un- signed integers, and Y = (y3 y2 y1 y0) is a four-bit unsigned integer. We can find,

$$\begin{array}{cccc}
 & & a_1 & a_0 \\
* & & b_1 & b_0 \\
\hline
 & & a_1 * b_0 & a_0 * b_0 \\
 & a_1 * b_1 & a_0 * b_1 & \\
\hline
c_2 & a_1 * b_1 + c_1 & a_1 * b_0 + a_0 * b_1 & a_0 * b_0
\end{array}$$

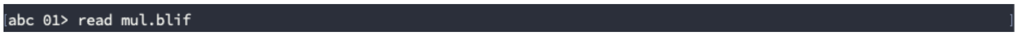
Then, we can easily write the blif with the following formula,

$$\begin{cases}
y_0 = a_0 \wedge b_0 \\
y_1 = (a_0 \wedge b_1) \oplus (a_1 \wedge b_0) \\
y_2 = (a_1 \wedge b_1) \oplus ((a_0 \wedge b_1) \wedge (a_1 \wedge b_0)) \\
y_3 = (a_1 \wedge b_1) \wedge ((a_0 \wedge b_1) \wedge (a_1 \wedge b_0))
\end{cases}$$

Or, we can write down the truth table to contstruct the blif

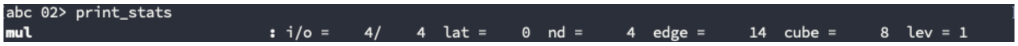
a1	a0	b1	b0	y3	y2	y1	y0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

(b) 1. **read mul.blif**



```
abc 01> read mul.blif
```

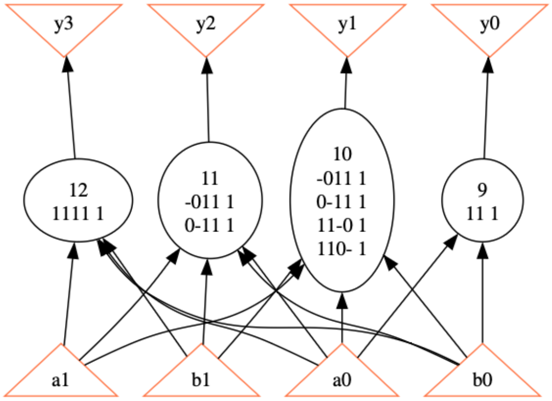
2. **print_stats**



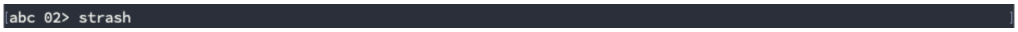
```
abc 02> print_stats
mul                          : i/o =    4/    4  lat =    0  nd =     4  edge =     14  cube =     8  lev = 1
```

3. **show**

The network contains 4 logic nodes and 0 latches.



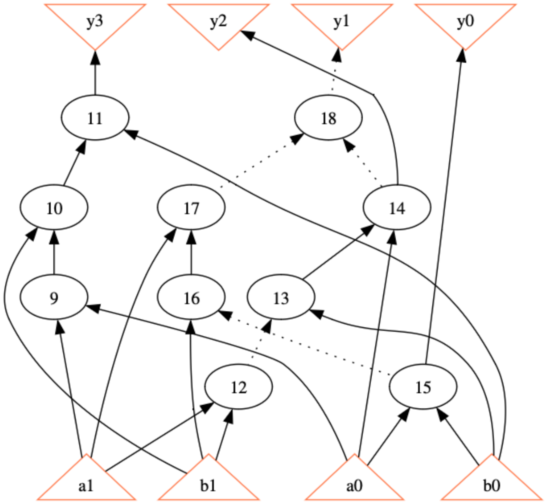
4. **strash**



```
abc 02> strash
```

5. **show**

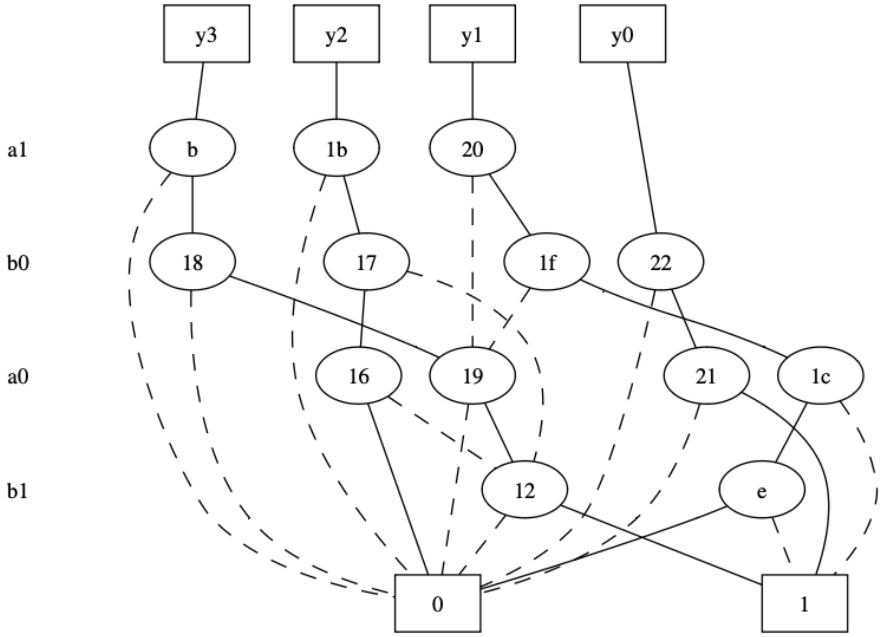
The network contains 10 logic nodes and 0 latches.



6. **collapse**

```
abc 03> collapse
```

7. **show_bdd -g**



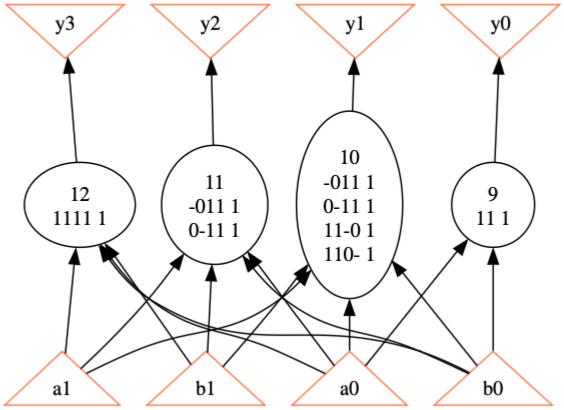
◇

Problem 3. [ABC Boolean Function Representations]

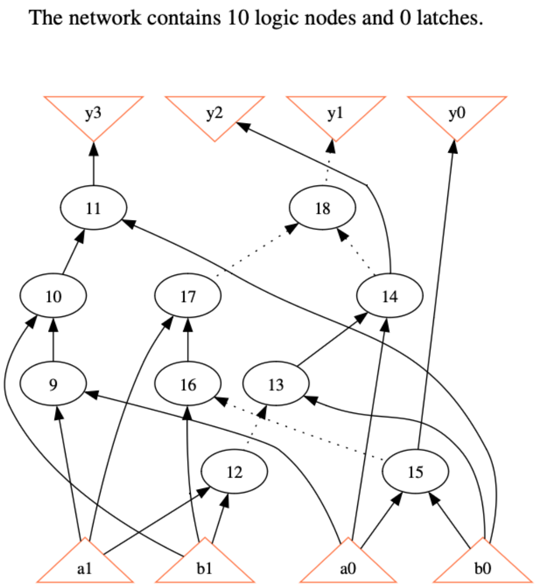
Solution:

(a) 1. Command **"aig"**

The network contains 4 logic nodes and 0 latches.

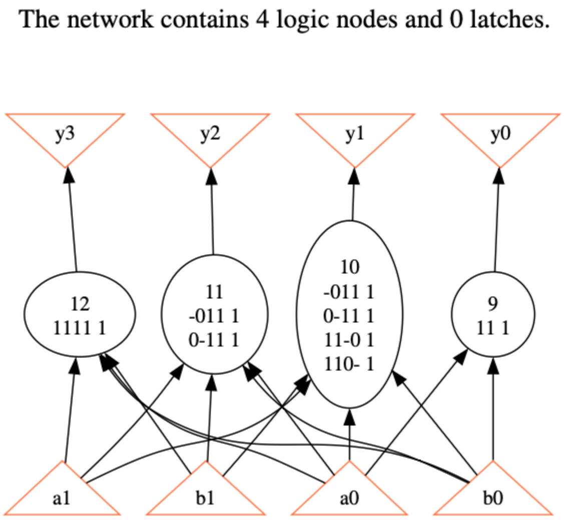


Command **"strash"**


The network contains 10 logic nodes and 0 latches.

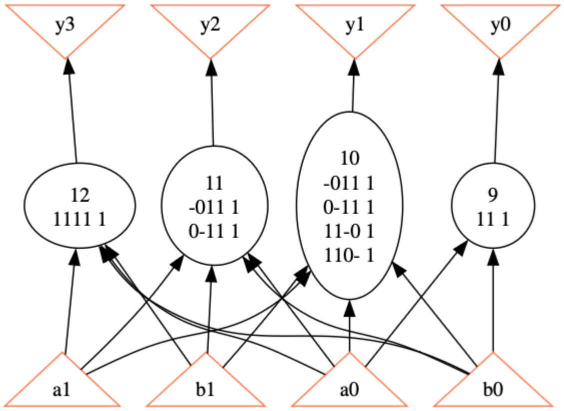
We can see that the figure that **ABC** shows has a different representation. The main difference is that in the command **show** (src/base/abc/abcShow.c:292), it will convert the logic network into "SOP" form to show the network, and **AIG** is the type of logic network, but the **Strashed AIG** is not the logic network in the definition in the ABC.

2. Command **"bdd"**


The network contains 4 logic nodes and 0 latches.

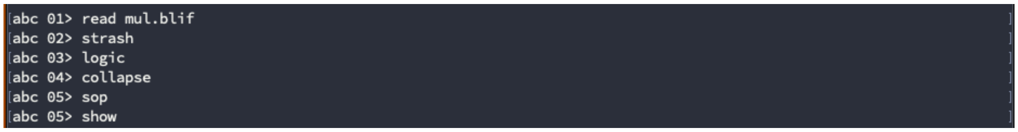
Command **"collapse"**

The network contains 4 logic nodes and 0 latches.



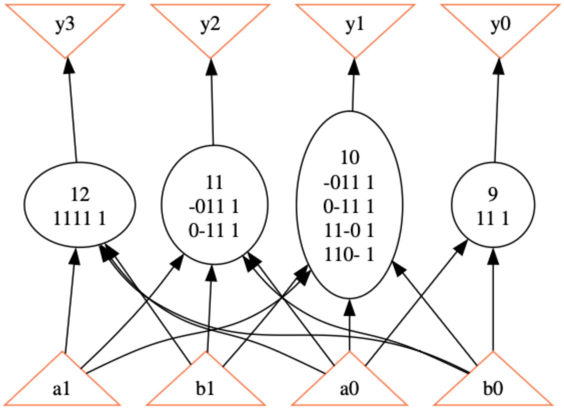
The command ”bdd” and command ”collapse” have the same representation because they are both logic networks. Therefore, the command **show** will convert them into the same ”SOP” form to show the network.

(b) After we get the structurally hashed AIG, we need to convert it into logic network first, using the command **”logic”**. Then, we can use the command **”collapse”** to obtain the same result as the previous result. Finally, we can use the command **”sop”** to convert a network into SOP form.



```
abc 01> read mul.blif
abc 02> strash
abc 03> logic
abc 04> collapse
abc 05> sop
abc 05> show
```

The network contains 4 logic nodes and 0 latches.



◇