LSV PA1

R09943109 陳冠誠

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. The Description of command "**aig**" and "**strash**":
   甲、**aig** – Converts local functions of the nodes to AIGs.
   乙、**strash** – Transforms the current network into an AIG by one-level structural hashing. The resulting AIG is a logic network composed of two-input AND gates and inverters represented as complemented attributes on the edges. Structural hashing is a purely combinational transformation, which does not modify the number and positions of latches.

   Thus, we know the main difference between **aig** and **strash** is that **strash** uses structural hashing to merge gates with same fanins.

2. The Description of command "**bdd**" and "**collapse**":
   甲、bdd – Converts local functions of the nodes to BDDs.
   乙、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. Collapsing is performed by building global functions using BDDs and is, therefore, limited to relatively small circuits. After collapsing, the node functions are represented using BDDs.

   Thus, we know the main difference between **bdd** and **collapse** is that **collapse** recursively composes the fanin nodes into the fanout nodes in a network. This limits the size of circuit command **collapse** could be used.

(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).

We could use command “**logic**” to convert structurally hashed AIG into SOP.