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. logic network in AIG (by command `aig`) vs. structurally hashed AIG (by command `strash`)
2. logic network in BDD (by command `bdd`) vs. collapsed BDD (by command `collapse`)

Four-bit adder	
command	print_stats
`aig`	i/o= 9/5, lat= 0, nd= 52, edge= 104, aig= 52, lev= 16
`strash`	i/o= 9/5, lat= 0, and= 52, lev= 16

aig: For a logic network, convert the local function of the nodes to AIGs.

strash: Transforms the current network into an AIG by one-level structural hashing.

Four-bit adder	
command	print_stats
`bdd`	i/o= 9/5, lat= 0, nd= 52, edge= 104, bdd= 104, lev= 16
`collapse`	i/o= 9/5, lat= 0, nd= 5, edge= 33, bdd= 43, lev= 1

bdd: For a logic network, converts local functions of the nodes to BDDs.

collapse: Builds global functions using BDDs resulting in a network, in which each primary output is produced by a node, whose fanins are primary inputs.

(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).

>> logic