

CAD Contest P1

B08901048 Yu-Chen, Chen

B08901061 Yung-Chin, Chen

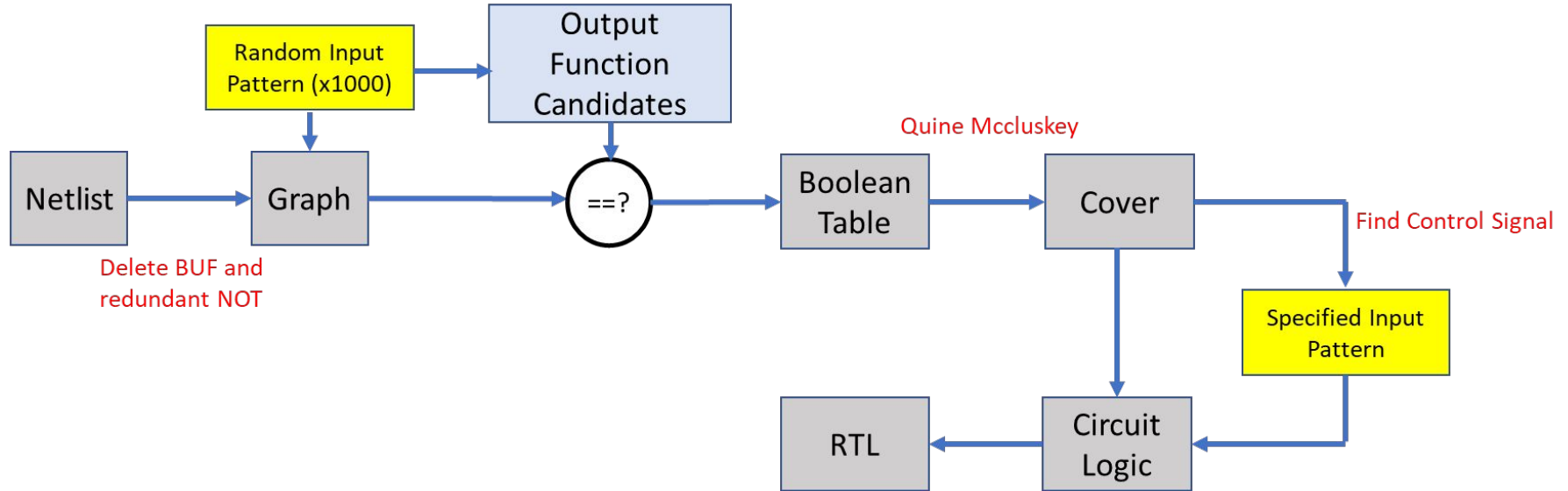
B08901062 Chia-Hsiang, Chang

Instructor: Prof. Jie-Hong Roland Jiang

Presupposition

- Input and output words will not be separated or permuted
- Operations include only +, -, *, if-else, ==, >, <

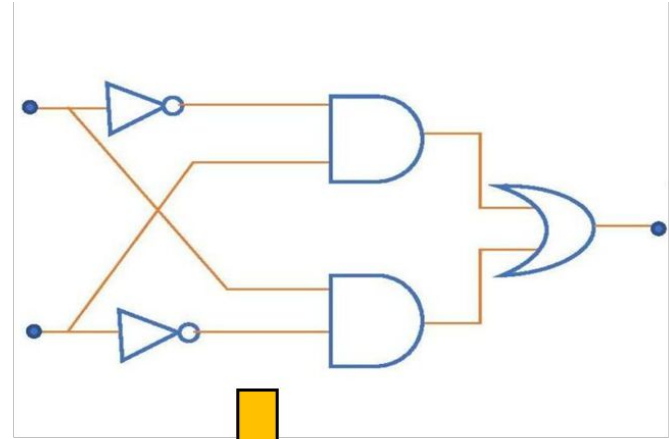
Procedure



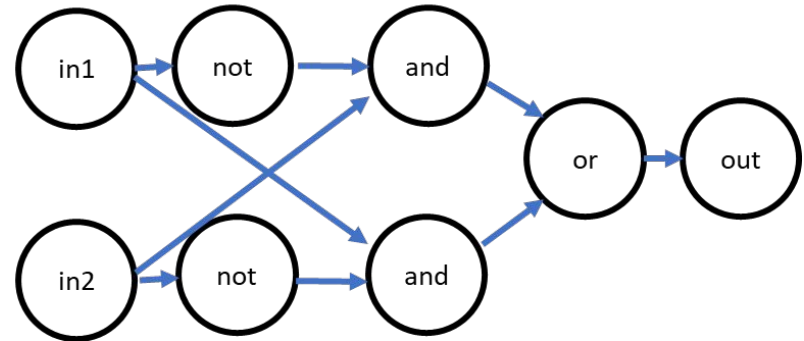
Netlist to Graph

Transfer netlist to graph

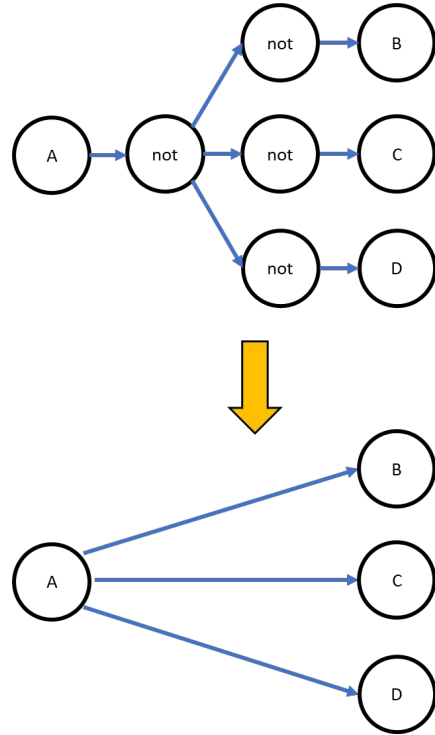
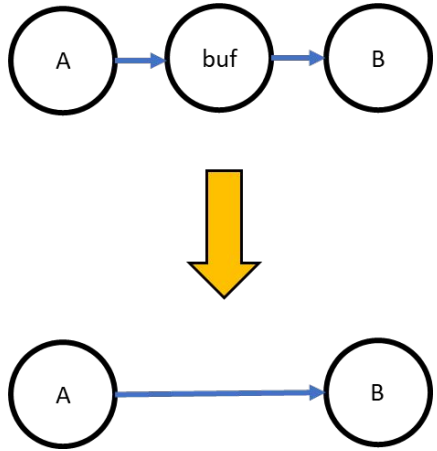
Netlist:



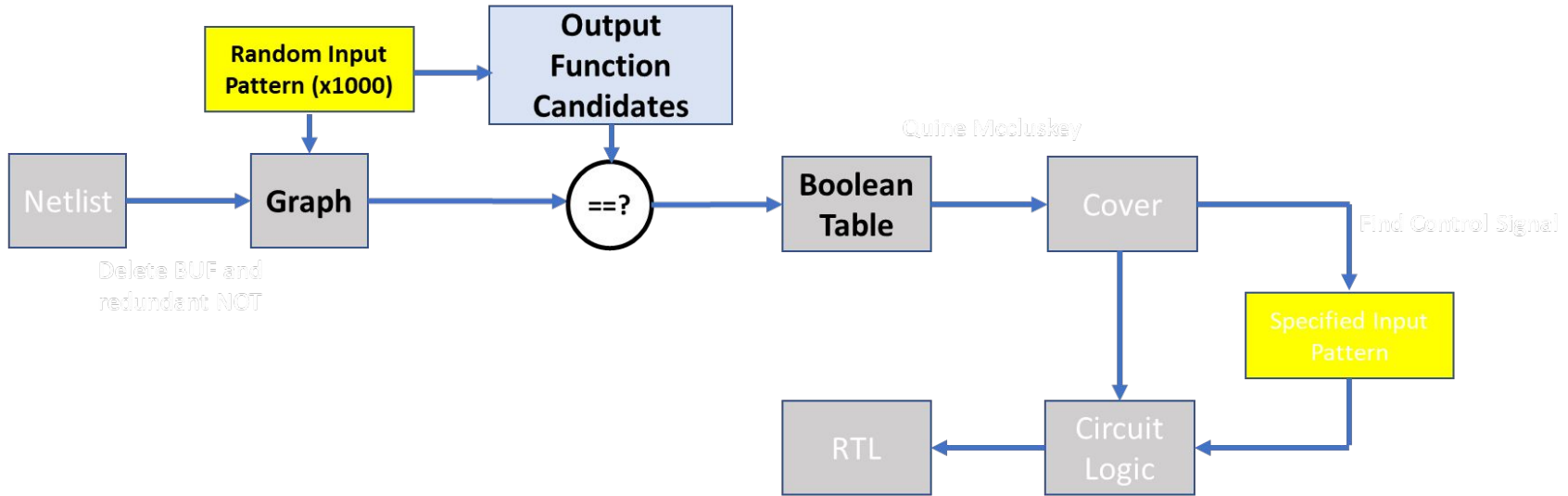
Graph:



Delete BUF and redundant NOT gate



Equivalence Checking



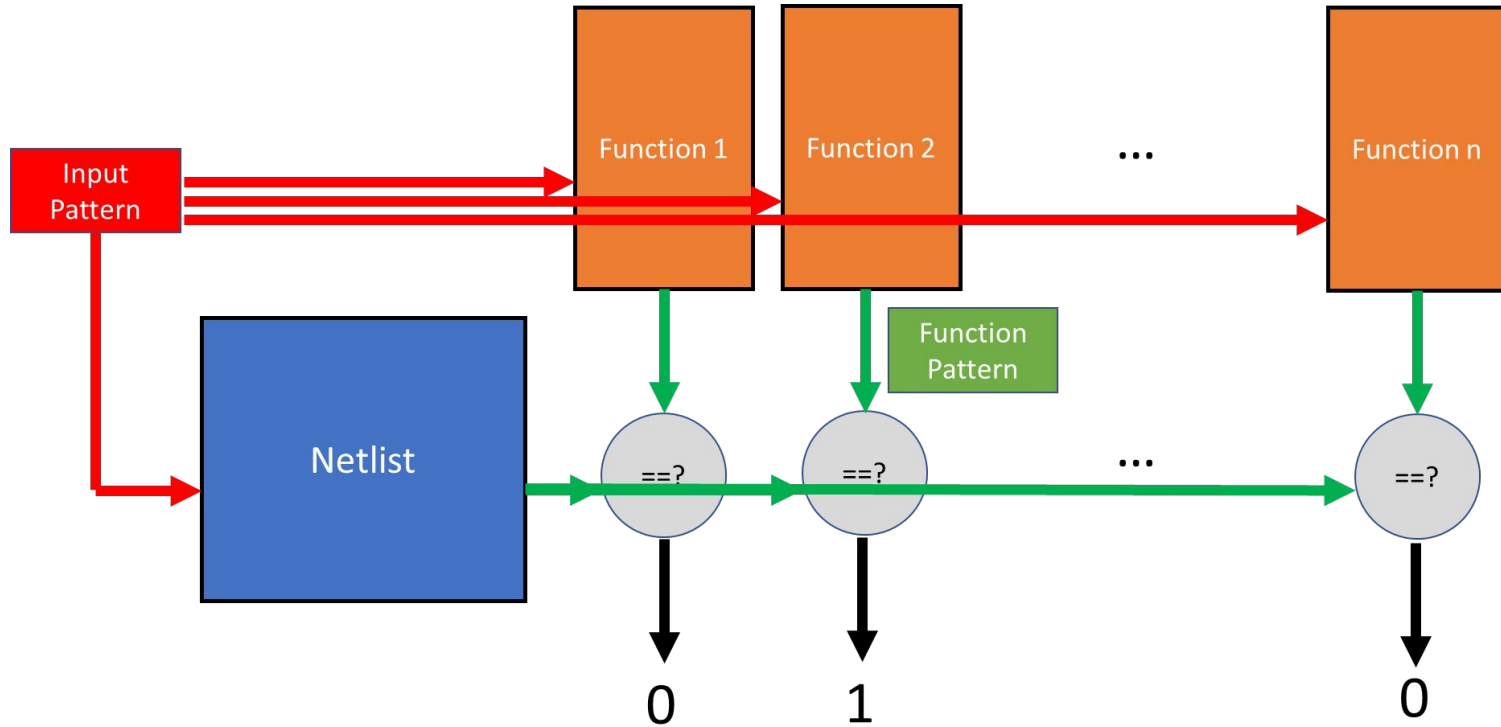
Output Function Assumption

Take a three input netlist for example:

$$F = S \cdot \{a, b, c, ab, bc, ca, a^2, b^2, c^2, abc, a^3, b^3, c^3\} + \text{constant}$$

S is a vector whose elements take on values of $\{-1, 0, 1\}$

Equivalence Checking



Quine McCluskey Procedure

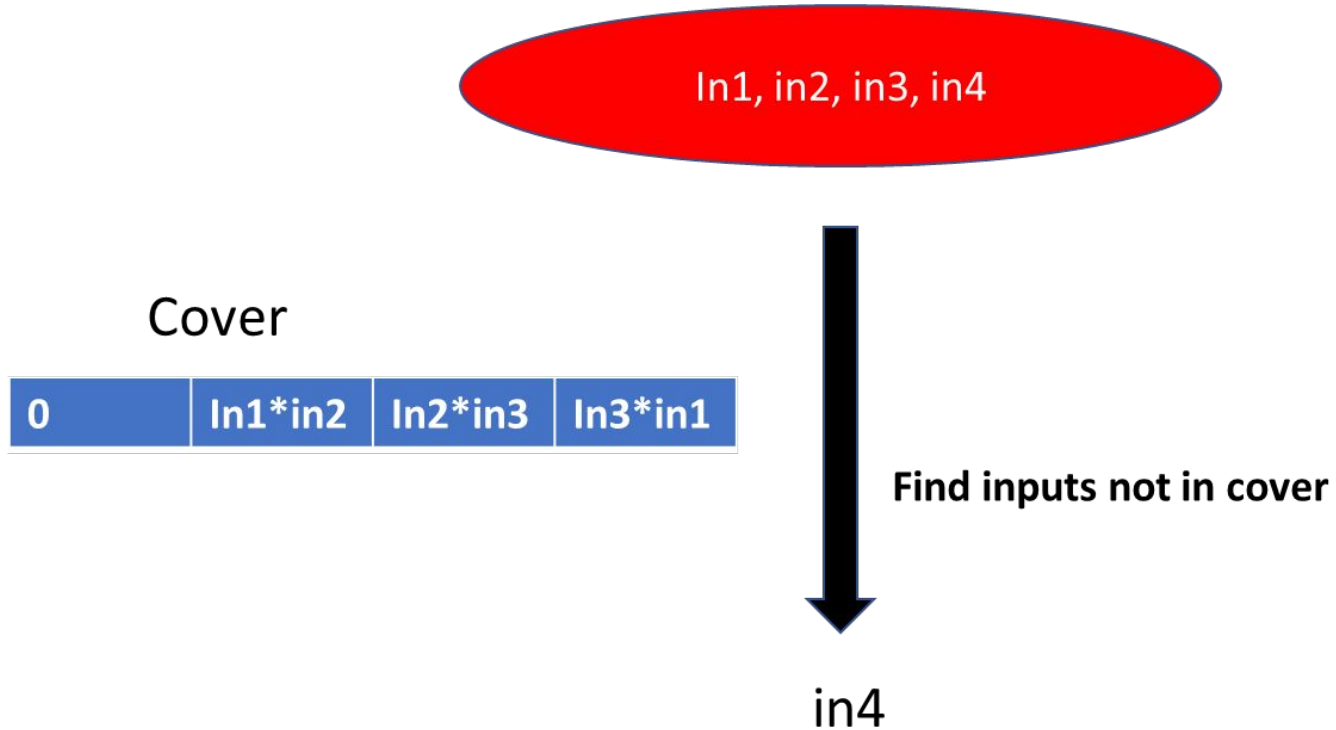
Find a cover from the table

Testing Function

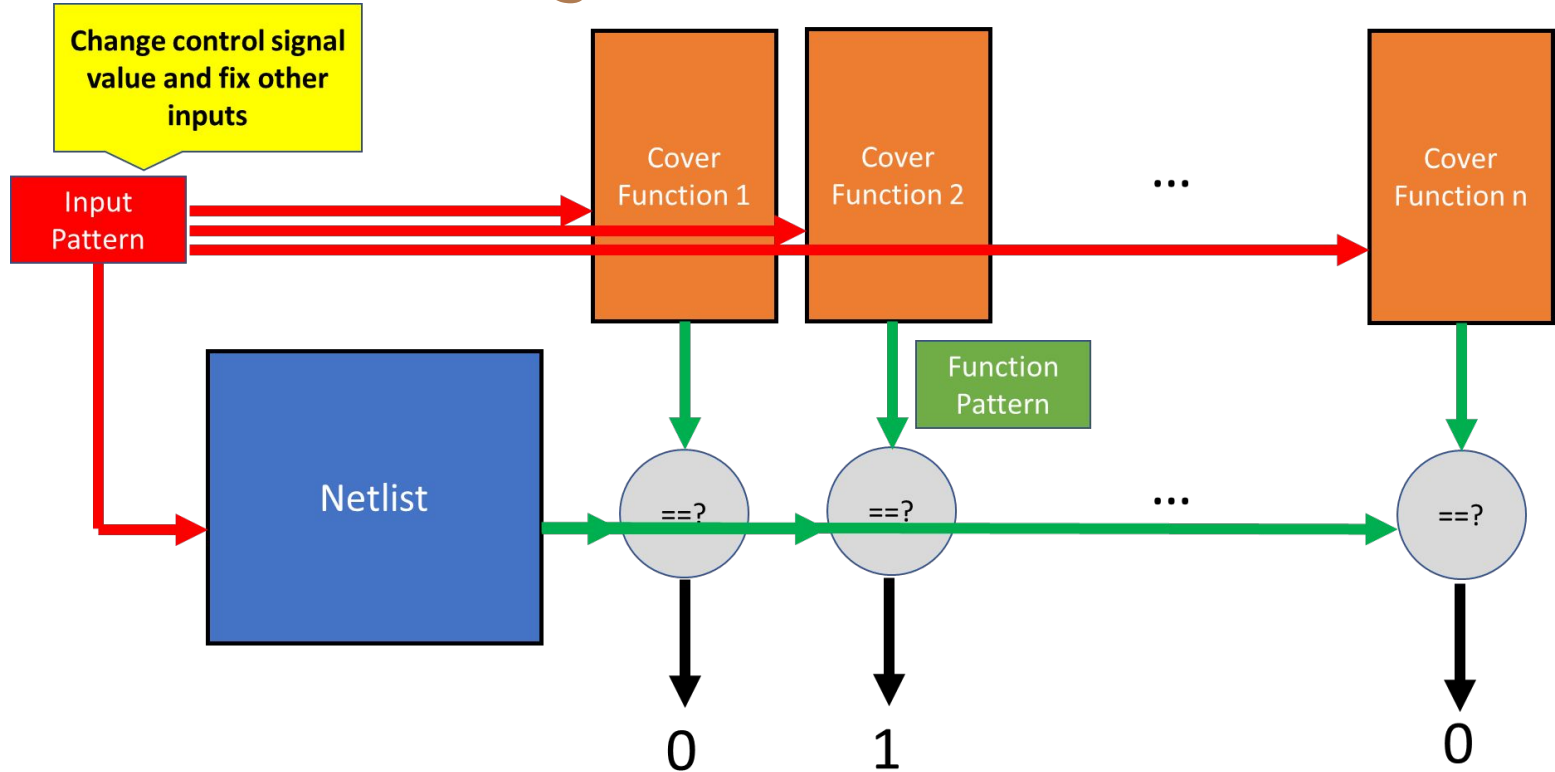
0	$In1*in2$	$In2*in3$	$In3*in1$	$In4$	$In4*in2$
1					
		1			
1				1	
		1			1
	1				
	1				
			1		
		1			
			1		

Random pattern

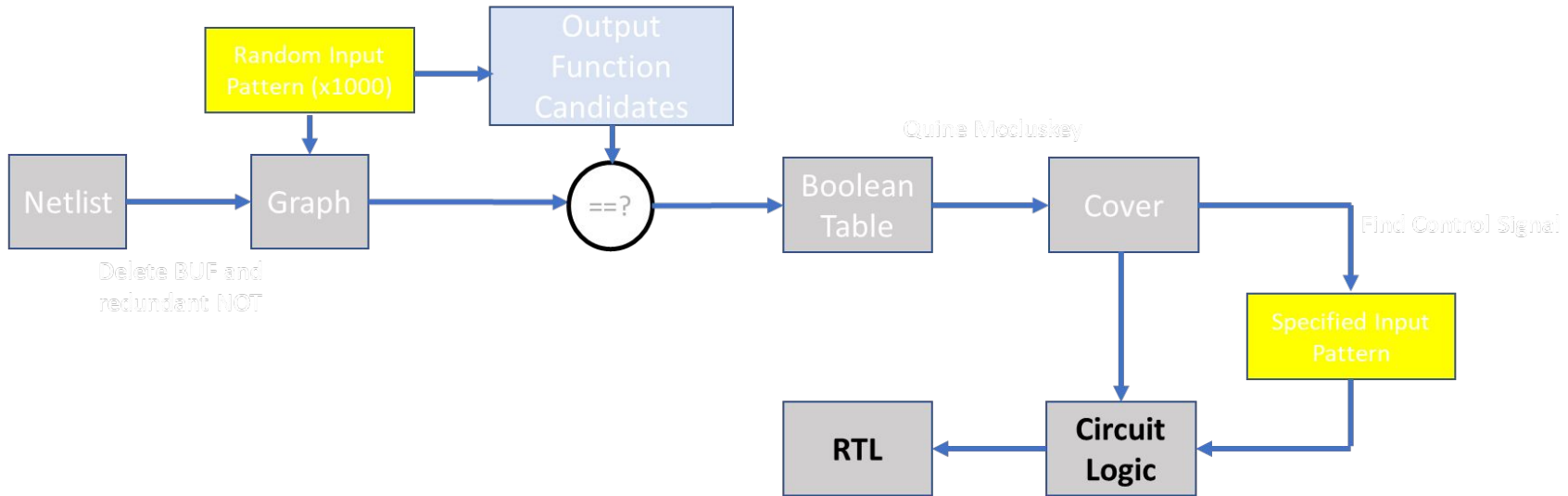
Find Control Signal



Find Control Logic



Generate RTL and Write File



Automation

>> execute .exe

>> input filename

→ Get verilog file

Results

expected final score: 800

```
module top(in1, in2, in3, out1);  
input wire [18:0] in1;  
input wire [18:0] in2;  
input wire [18:0] in3;  
output wire [19:0] out1;  
assign out1 = in1 + in2 + in3 + 102;  
endmodule
```

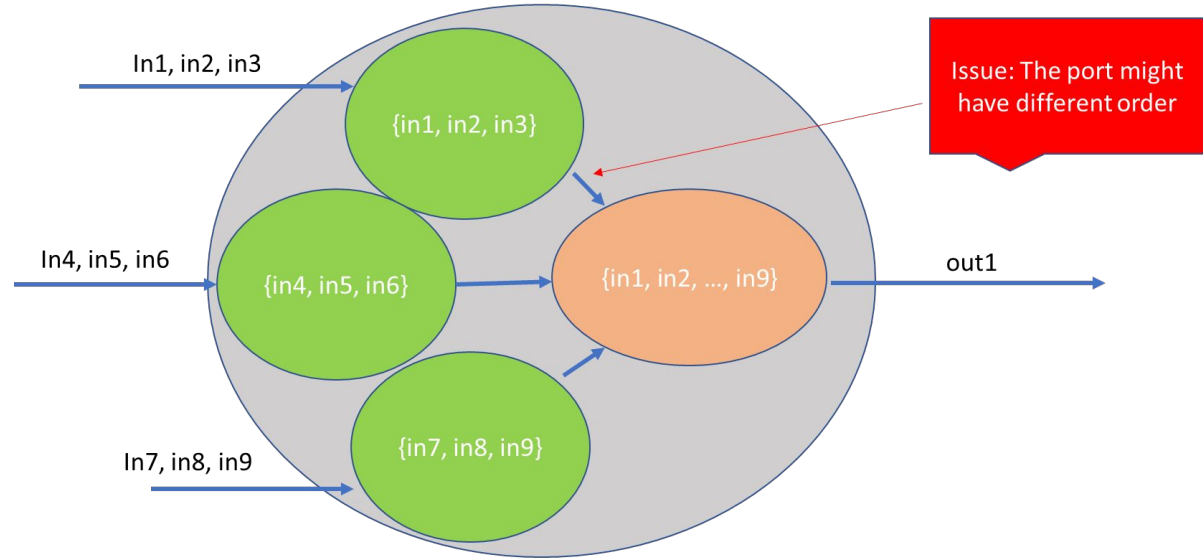
```
module top(in1, in2, in3, out1);  
input wire [31:0] in1;  
input wire [31:0] in2;  
input wire [31:0] in3;  
output wire [64:0] out1;  
assign out1 = in2 + in3 + in1 * in1;  
endmodule
```

Testcases	Reduction rate(%)	Runtime(s)
01	94.2857	2.075
02	96.5517	2.147
03	99.0415	2.211
04	99.862	64.928
05	2.7907	0.355
06	99.9395	4.51
07	99.7540	0.719
08	12.7967	20.111
09	40.4	6.639
10	99.7743	4.915
11	99.8329	4.38
12	41.614	11.756
13	20.623	0.057
14	0	2.300
15	42.6182	1.73
16	20.2497	1.322
17	11.4073	0.066
18	15.625	0.343
19	54.1646	0.179
20	18.8586	0.831

Future Prospects

Problem: Space Complexity blows up when # inputs increases

$$out1 = (in1 + in2 + in3) + (in4 + in5 + in6) + (in7 + in8 + in9)$$



{·} indicates the possible transitive fanin that a gate can possess