

VLSI Testing PA2
B08505039 工海四 林楷崑

1. In `podem.cpp`, there are three TODO :

1.1. Function *find_pi_assignment* is used to determine initial PI through objective. According to Fig.9. in the Goel's paper, if the objective is not fed by a PI, first we should determine the type of gate that drives the objective to find the next objective net.

From Fig.9.4/9.5:

NAND/OR with Objective = 1 -> take `easiest_control`

AND/NOR with Objective = 1 -> take `hardest_control`

Second, we should determine the gate to find the next objective level.

From Fig.9.7/9.8:

NAND/NOR -> next objective level should be complement with the current
otherwise -> next objective level can be the same

Last, repeating this function until we find the PI.

1.2. Function *trace_unknown_path* searches for X-path and it return TRUE if X-path exists. We should check If the wire is an output and its value is unknown, then return TRUE. Otherwise, checking all its fanout to trace again. If all conditions not satisfied, return FALSE.

1.3. For function *set_uniquely_implied_value* fault excitation part, according to the hint, we should check if the fault can be excited by backward implication. Thus, for SA0, we should use `backward_imply` with value 1 and for SA1, we should use `backward_imply` with value 0.

2. The complexity of *trace_unknown_path* could be exponential of n since in *trace_unknown_path*, it would call itself if there is need to check fanout.

circuit number	number of gates	number of total faults	number of detected faults	number of undetected faults	fault coverage	number of test vector	run time
C432	245	1110	149	961	13.42%	20	0.1s
C499	554	2390	2280	110	95.40%	74	0.3s
C880	545	2104	1254	850	59.60%	62	0.5s
C1355	554	2726	1702	1024	62.44%	63	3.2s
C2670	1785	6520	6278	242	96.29%	156	2.4s
C3540	2082	7910	2424	5486	30.64%	95	26s
C6288	4800	17376	17109	267	98.46%	47	5.3s
C7552	5679	19456	19145	311	98.40%	267	33.6s