Test and Diagnosis Pattern Generation for Dynamic Bridging Faults and Transition Delay Faults

Cheng-Hung Wu, Saint James Lee and Kuen-Jong Lee

National Cheng Kung University
Tainan, Taiwan

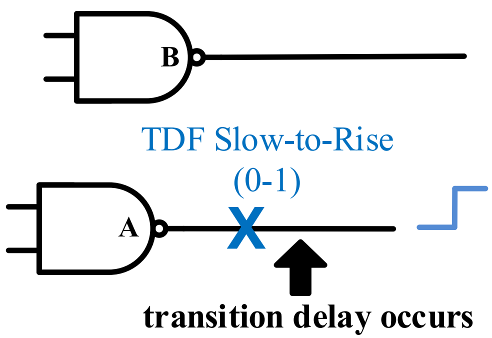
ASPDAC 2016

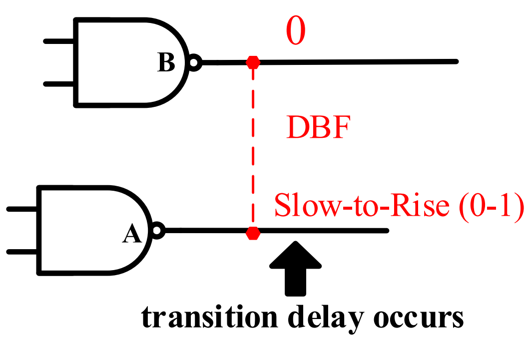
Outline

- Introduction
- Background
  - Dynamic Bridging Faults (DBFs)
  - Dominance
  - DBF Detection
- Diagnosis Pattern Generation
  - Inverse Dynamic Bridging Faults (IDBFs)
  - Fault Pair Classification
  - Test and Diagnosis Pattern Generation Flow
- Experimental Results
- Conclusions

Introduction

- Accurate diagnosis needs to identify both the location and the type of fault.
- Both the transition delay fault (TDF) and the dynamic bridging fault (DBF) have similar transition delay faulty effect.

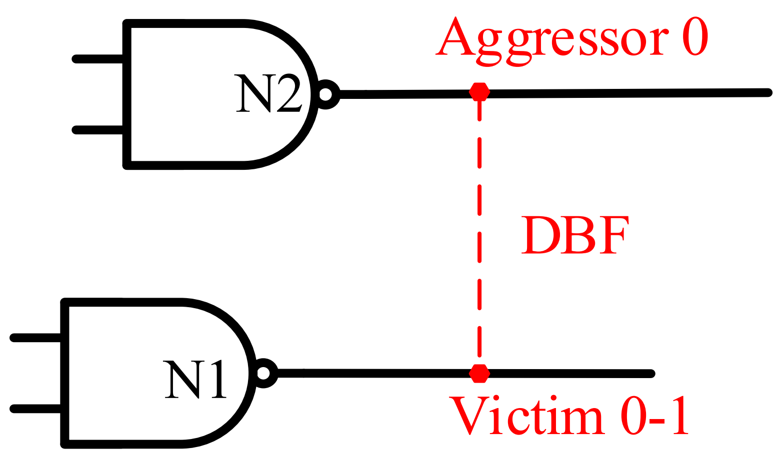




- We need a diagnosis method to distinguish these two similar fault models.

Dynamic Bridging Faults

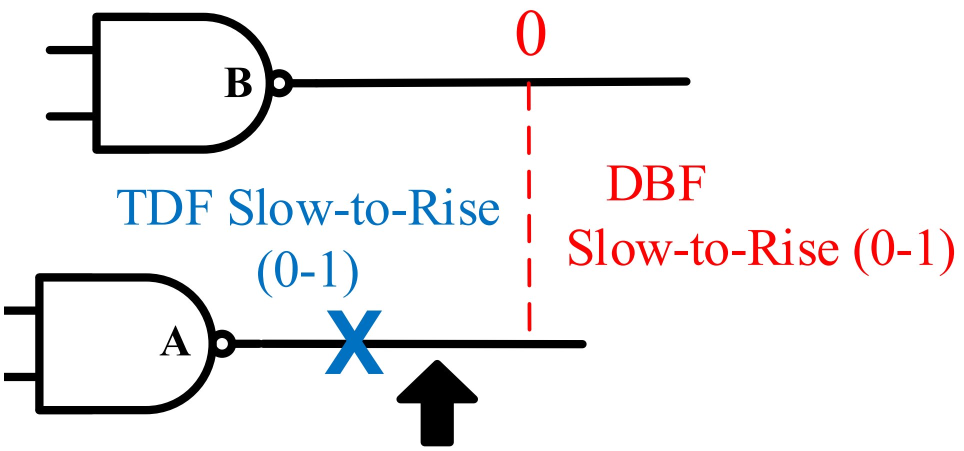
- Each DBF involves two wire nodes; one whose delay is affected is called the victim and the other is called the aggressor of the DBF.



- There are two types of DBFs including bridge slow-to-rise (bsr) and bridge slow-to-fall (bsf).

Fault Pairs

- Due to similar faulty effect, a fault pair is defined as a pair composed of a DBF and the corresponding TDF which located on the DBF's victim.



transition delay occurs

TDF? DBF?

Dominance

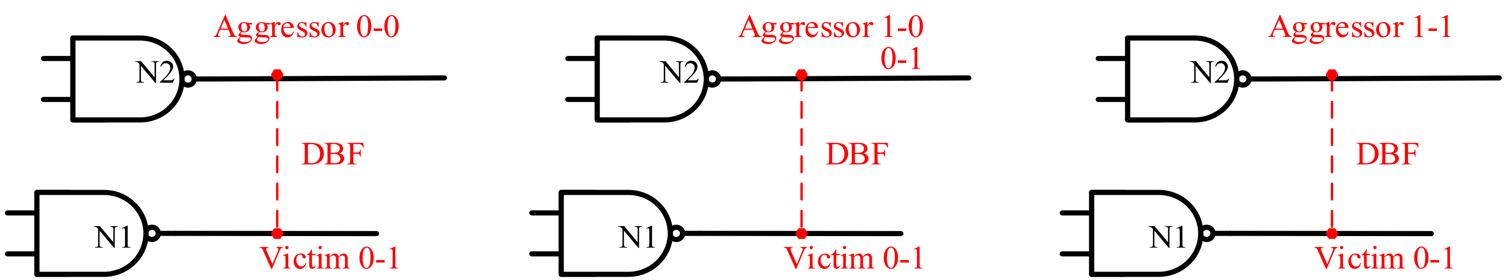
- **(1) Active a transition on target wire and propagate to some observable outputs.**
  (2) Set a constraint on the other target wire.

  To test DBF need to satisfy both **(1)** and **(2)**, but to test TDF only need to satisfy **(1)**, therefore TDF → DBF.

- In order to distinguish a DBF from its corresponding TDF, a pattern must detect the TDF without detecting the DBF.

DBF Detection

- To definitely detect a DBF, the aggressor must be set to a constant value that is opposite to the final value of the victim during the application of the test pattern.



Condition 1
Definitely detected

Condition 2
Probably detected

Condition 3
Definitely not detected

Summary of Fault Behaviors

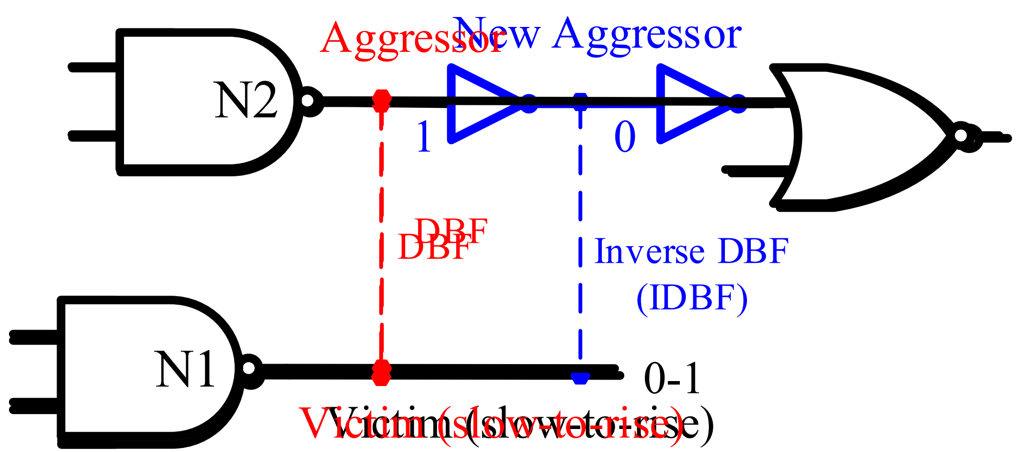
1. Conventional tests targeting TDFs may not detect some DBFs.
2. If a test for a TDF also detects a DBF dominated by the TDF, the test cannot distinguish the two faults.
3. The only way to distinguish a DBF and its corresponding TDF is to inactivate the DBF while detecting the TDF.
4. To detect the TDF without detecting the DBF, the aggressor of DBF must hold the opposite value of victim's first-cycle value.

Outline

- Introduction
- Background
  - Dynamic Bridging Faults (DBFs)
  - Dominance
  - DBF Detection
- Diagnosis Pattern Generation
  - Inverse Dynamic Bridging Faults (IDBFs)
  - Fault Pair Classification
  - Test and Diagnosis Pattern Generation Flow
- Experimental Results
- Conclusions

Inverse Dynamic Bridging Faults (IDBFs)

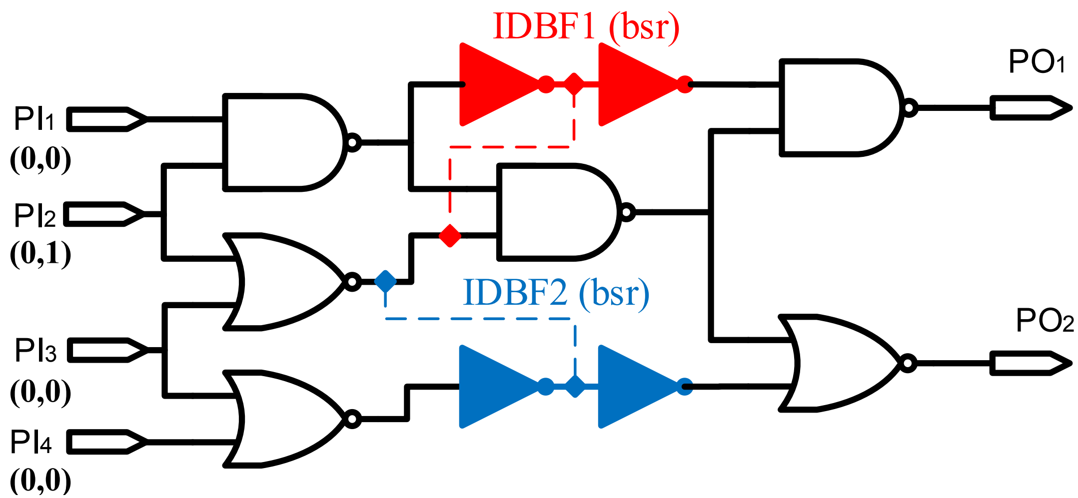
- Our method uses the inverter to set the opposite constant value on the aggressor by building a new DBF with a new aggressor.



- The DBF would not be detected when the corresponding IDBF is detected.

Diagnosis Method

1. Insert the inverter pairs for all wires at once, which does not affect original circuit’s function.
2. This method targets all inverse DBF at a time in the ATPG stage so as to achieve dynamic pattern compaction.



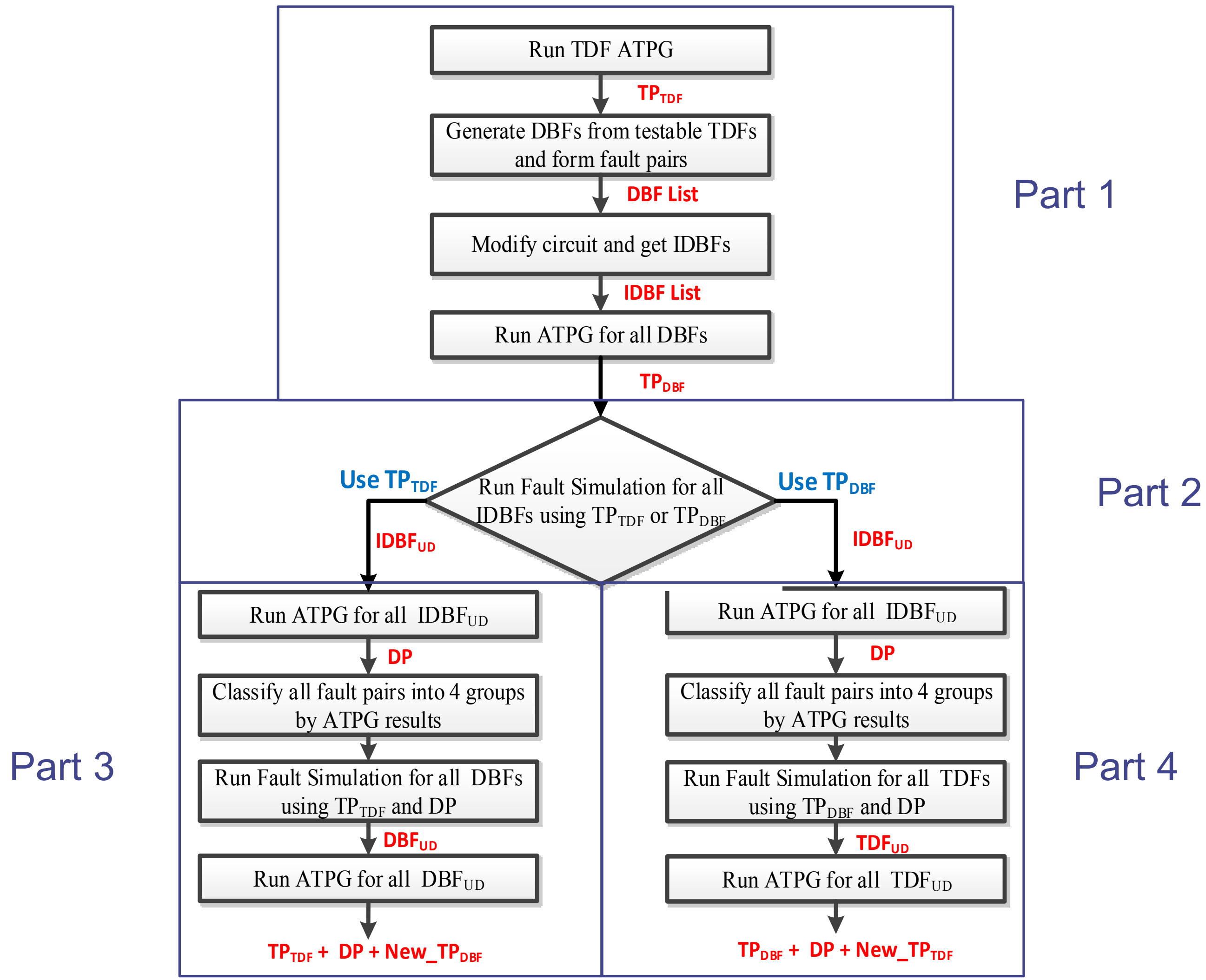
Fault Pairs Classification

IDBF \ DBF	Undetectable	Detected	Aborted
Detected		Group 2	
Undetectable	Group 1	Group 3	
Aborted		Group 4	

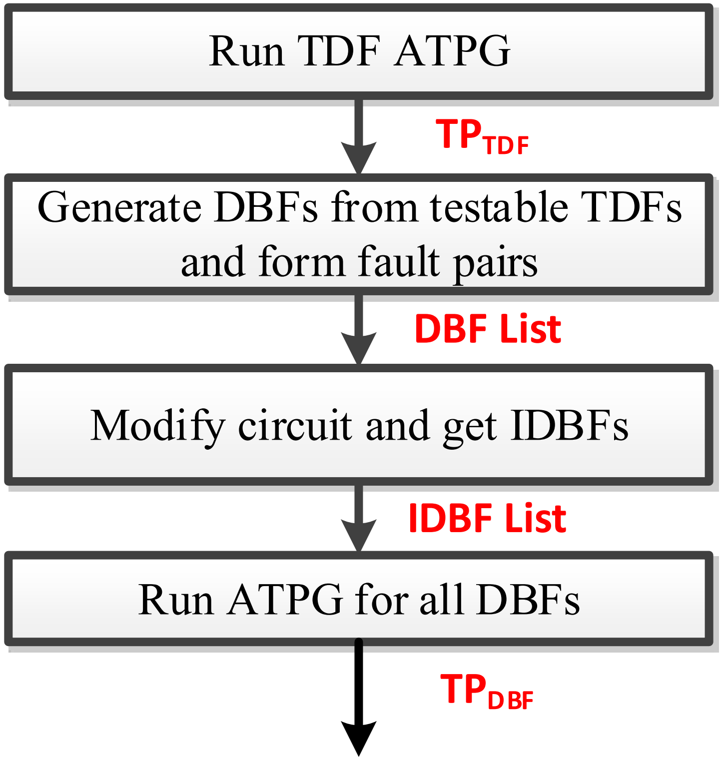
$$\text{DR\%} = \frac{\#Group\ 1 + \#Group\ 2 + \#Group\ 3}{\#All\ fault\ pairs\ under\ consideration}$$

- Group 1:
  - DBF = Undetectable
  - (Distinguished)
- Group 2:
  - DBF = Detected or Aborted
  - IDBF = Detected
  - (Distinguished)
- Group 3:
  - DBF = Detected
  - IDBF = Undetectable
  - (Equivalent)
- Group 4:
  - the remaining cases
  - (Unidentified)

Test and Diagnosis Flow - Overview

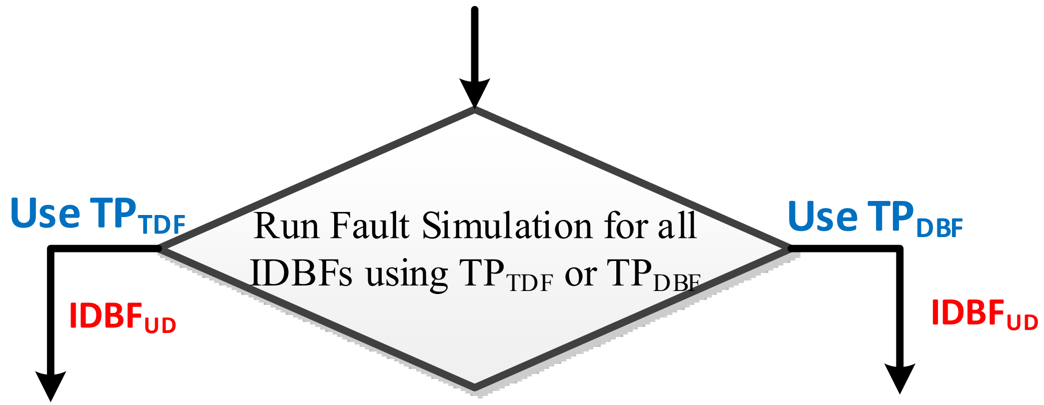


Test and Diagnosis Flow - Part 1



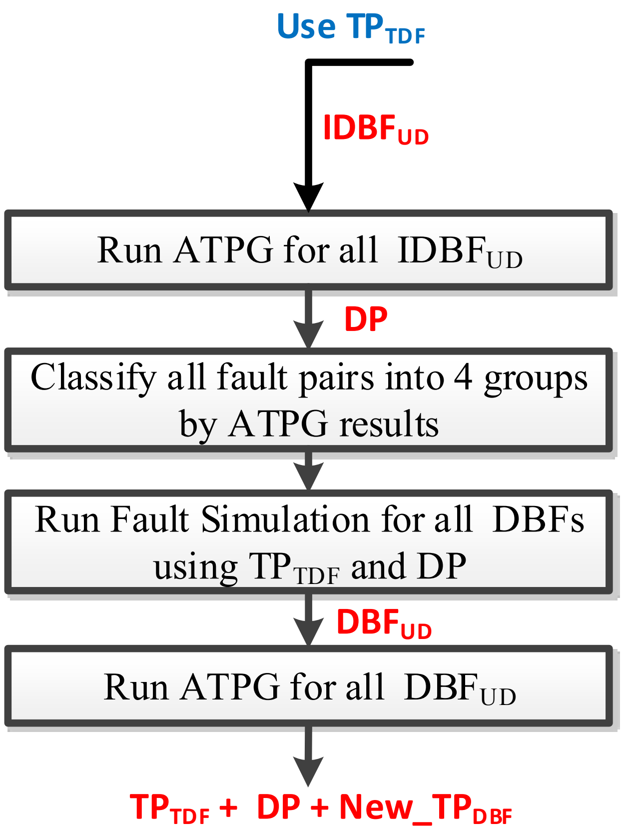
- Due to the dominance relation, we can drop all undetectable TDFs.
- Build a DBF list based on detectable TDFs .
- Build fault pairs according to the DBF list and IDBF list.
- Form a IDBF for each fault pair.

Test and Diagnosis Flow - Part 2



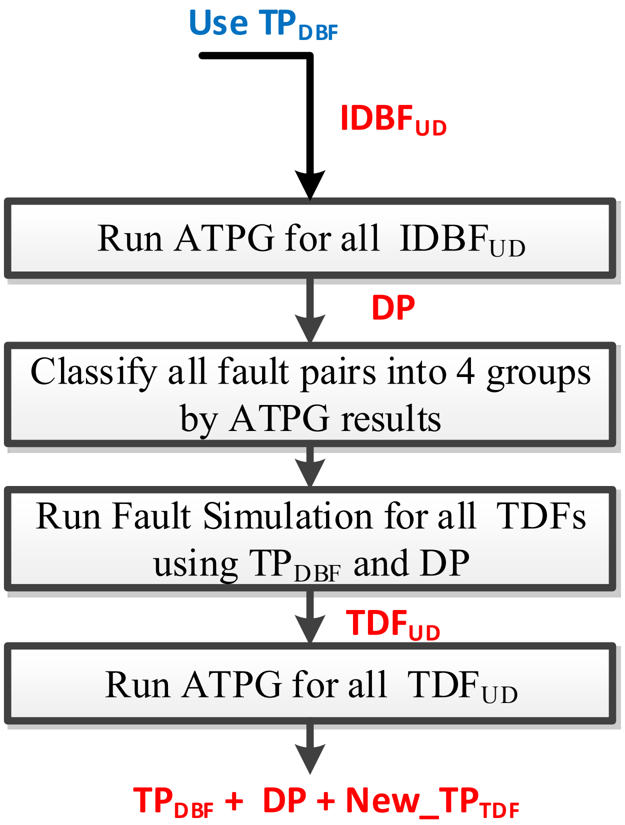
- The test patterns for IDBF fault simulation can use either TP_{TDF} or TP_{DBF}.
- The selection between TP_{TDF} and TP_{DBF} will cause the difference of total number of test patterns and diagnosis patterns since the sequence of ATPG processes are different.
- Distinguish fault pairs by the test patterns using IDBF's fault simulation process.

Test and Diagnosis Flow - Part 3



- The patterns to detect $IDBF_{UD}$ are diagnosis patterns (DP).
- Classify all fault pairs into 4 groups, thus the diagnosis resolution is obtained.
- Generate test patterns for the rest of TDFs or DBFs so as to achieve both TDF testing and DBF testing.

Test and Diagnosis Flow - Part 4



- The difference between part 3 and part 4 is the execution sequence of ATPG.

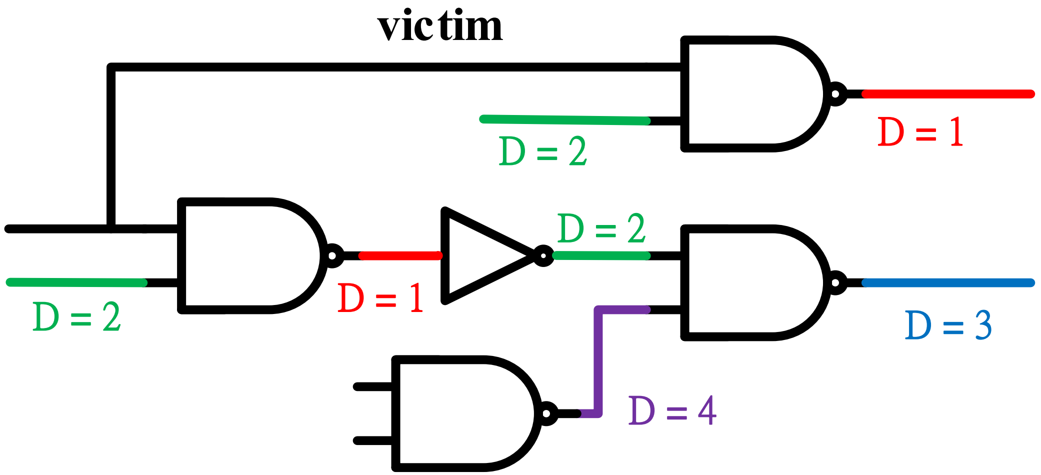
- TDF→IDBF→DBF on part 3
  DBF→IDBF→TDF on part 4

Outline

- Introduction
- Background
  - Dynamic Bridging Faults (DBFs)
  - Dominance
  - DBF Detection
- Diagnosis Pattern Generation
  - Inverse Dynamic Bridging Faults (IDBFs)
  - Fault Pair Classification
  - Test and Diagnosis Pattern Generation Flow
- Experimental Results
- Conclusions

Definition of Distance

- We define the distance between an aggressor and a victim of a fault pair as the number of gates on the shortest path from the aggressor to the victim with both forward paths and backward directions considered.



Experimental Results

- **Results of Fault Lists Generated by Distance ≤ 4.**

CKT	#TDF	Distance	#FP	Number (Percentage) of each Group				DR%
				Group 1	Group 2	Group 3	Group 4	
s15850	16104	≦1	24922	2501 (10.04%)	13326 (53.47%)	9021 (36.20%)	74 (0.30%)	99.70%
		≦2	132228	19290 (14.59%)	90968 (68.80%)	21688 (16.40%)	282 (0.21%)	99.79%
		≦3	412973	40148 (9.72%)	327297 (79.25%)	45015 (10.90%)	513 (0.12%)	99.88%
		≦4	1004722	75459 (7.51%)	844773 (84.08%)	83817 (8.34%)	673 (0.07%)	99.93%
s35932	37530	≦1	27426	384 (1.40%)	18154 (66.19%)	8888 (32.41%)	0 (0.00%)	100%
		≦2	1458620	49658 (3.40%)	1337149 (91.67%)	71813 (4.92%)	0 (0.00%)	100%
		≦3	4553456	79310 (1.74%)	4255628 (93.46%)	218518 (4.80%)	0 (0.00%)	100%
		≦4	14965716	431966 (2.89%)	14141303 (94.49%)	392447 (2.62%)	0 (0.00%)	100%
s38417	43757	≦1	127543	7192 (5.64%)	83829 (65.73%)	36414 (28.55%)	108 (0.08%)	99.92%
		≦2	708823	64976 (9.17%)	568857 (80.25%)	74112 (10.46%)	878 (0.12%)	99.88%
		≦3	2285025	120570 (5.28%)	2021045 (88.45%)	141690 (6.20%)	1720 (0.08%)	99.92%
		≦4	5298061	179665 (3.39%)	4917819 (92.82%)	196612 (3.71%)	3965 (0.07%)	99.93%
s38584	52007	≦1	116192	8717 (7.50%)	62085 (53.43%)	45358 (39.04%)	32 (0.03%)	99.97%
		≦2	567966	99059 (17.44%)	3616013 (63.67%)	107111 (18.86%)	193 (0.03%)	99.97%
		≦3	2346913	239687 (10.21%)	1792810 (76.39%)	313832 (13.37%)	584 (0.02%)	99.98%
		≦4	8771968	826337 (9.42%)	7212141 (82.22%)	731036 (8.33%)	2454 (0.03%)	99.97%
Average of ≦4				5.80%	88.40%	5.75%	0.04%	99.96%

- Our diagnosis method can be applied on 88.40% of fault pairs.
- 99.96% of fault pairs can be identified as distinguished or equivalent.
  - Group 1 = Distinguished
    Group 2 = Distinguished
    Group 3 = Equivalent
    Group 4 = Unidentified

Experimental Results

- **Numbers of Test Patterns and Diagnosis Patterns for Fault Lists Determined by Distance.**

CKT	#TDF	Distance	#FP	DATPG after using TP_{TDF}						DATPG after using TP_{DBF}					
				$\#TP_{TDF}$	$\#D\_DIS_{TDF}$	$\#DP_{TDF}$	$\#D\_DIS_{TDF}/\#DP_{TDF}$	#New TP_{DBF}	#TP+#DP	$\#TP_{DBF}$	$\#D\_DIS_{DBF}$	$\#DP_{DBF}$	$\#D\_DIS_{DBF}/\#DP_{DBF}$	#New TP_{TDF}	#TP+#DP
s15850	16104	≦1	24922	201	1598	229	6.98	99	529	415	1131	105	10.77	96	616
		≦2	132228		8306	507	16.38	196	904	510	1262	163	7.74	75	748
		≦3	412973		30366	816	37.21	349	1366	984	2730	162	16.85	62	1208
		≦4	1004722		74691	1494	49.99	487	2182	1701	3989	20	19.09	55	1965
s35932	37530	≦1	27426	52	2063	48	42.98	12	112	74	1524	36	42.33	7	117
		≦2	1458620		49559	64	774.36	98	214	315	718	14	51.29	2	331
		≦3	4553456		262854	223	1178.72	23	298	395	1217	19	64.05	2	416
		≦4	14965716		627867	255	2462.22	40	347	492	358	11	32.55	1	504
s38417	43757	≦1	127543	214	9594	325	29.52	190	729	472	3662	251	14.59	19	742
		≦2	708823		54303	801	67.79	385	1400	1019	3839	338	11.36	16	1373
		≦3	2285025		188071	1606	117.11	532	2352	1837	5568	589	9.45	8	2434
		≦4	5298061		454406	2428	187.15	669	3311	2709	5973	670	8.91	9	3388
s38584	52007	≦1	116192	253	6050	377	16.05	200	830	433	4286	355	12.07	40	828
		≦2	567966		28480	919	30.99	628	1800	1311	4113	476	8.64	31	1818
		≦3	2346913		145725	2342	62.22	801	3396	2310	10966	1171	9.36	19	3500
		≦4	8771968		381876	2426	157.41	2354	5033	4561	19576	583	33.58	20	5164

Conclusions

- Based on the dominance relation between TDFs and DBFs, we propose a novel circuit modified method to generate very compact diagnosis patterns to distinguish TDFs and DBFs.

- The results show that very high diagnosis resolution can be achieved by our proposed flow. (99.96% for fault pairs where DBF's aggressor and victim are near to each other, 99.99% for fault pairs that are randomly selected).