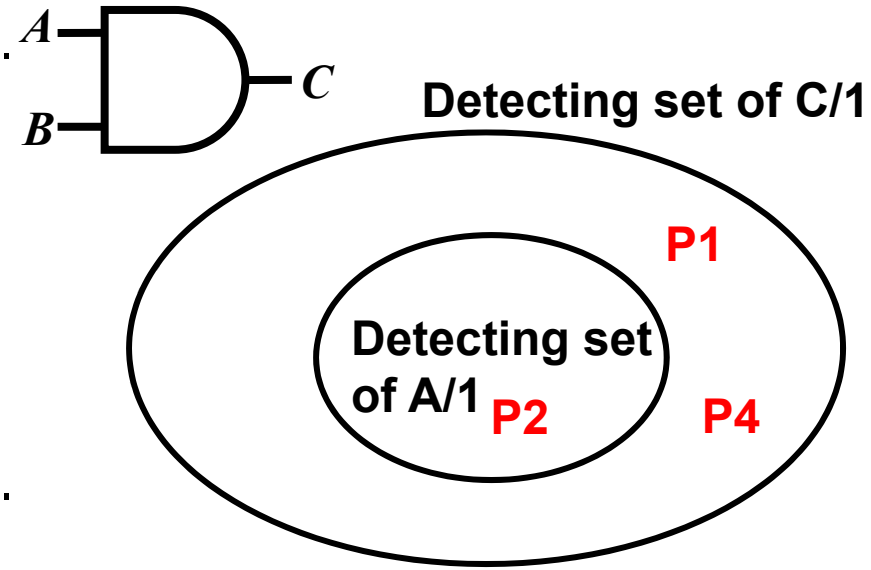
Fault Collapsing

- Introduction
- Equivalence Fault Collapsing
  - Fanout-free circuits
  - Fanout
- Dominance Fault Collapsing
  - Fanout-free circuits
  - Circuits with fanout
- Conclusion

Fault Dominance

- ***Detecting set*** of fault *f* (denoted as T_f) is set of all test patterns that detect fault *f*
- Fault *f* *dominates* fault *g* if the detecting set of fault *f* contains that of fault *g* *i.e.* $T_f \supseteq T_g$
- Example: C/1 fault dominates A/1 fault
  - C/1 detecting set {00, 01, 10} contains A/1 detecting set {01}
  - C/1 is ***dominating fault***; A/1 is ***dominated fault***

	Input		Output with Stuck Fault						
	A	B	good	A/0	A/1	B/0	B/1	C/0	C/1
P1	0	0	0	0	0	0	0	0	1
P2	0	1	0	0	1	0	0	0	1
P3	1	1	1	0	1	0	1	0	1
P4	1	0	0	0	0	0	1	0	1

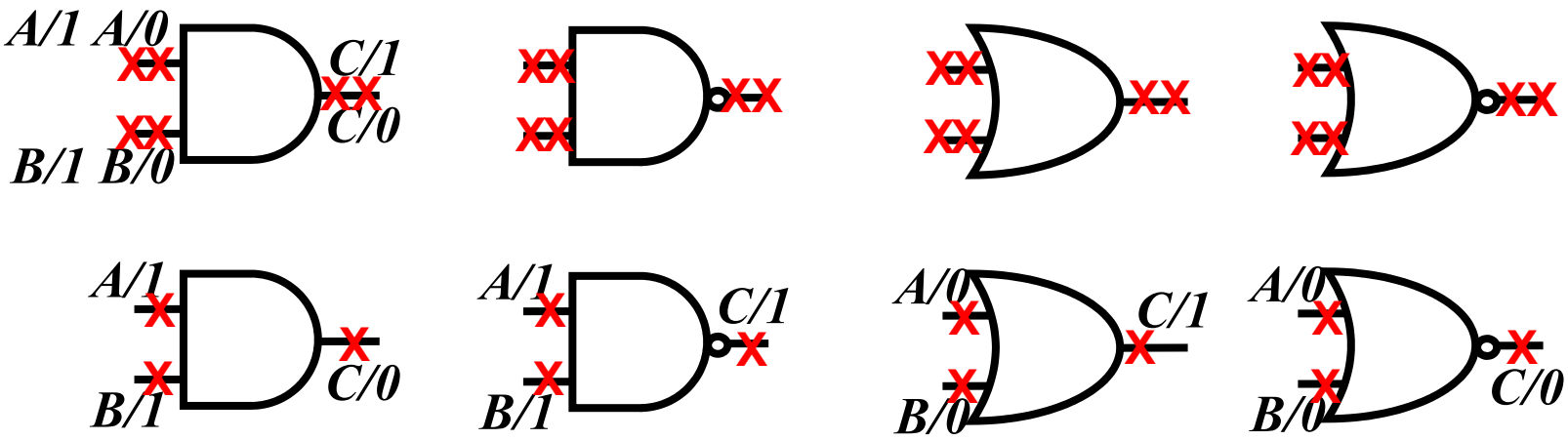


Dominance Fault Collapsing, DFC

- **DFC reduces fault list using fault dominance relation**
  - **If a test pattern detects a *dominated fault*, then it must detect its *dominating fault***
  - **so *dominating faults can be removed***
- **Property**
  - **If two faults are equivalent, then they dominate each other**
- **Therefore, the number of dominance collapsed faults must be smaller or equal to that of equivalence collapsed faults**

DFC on Elementary Gates

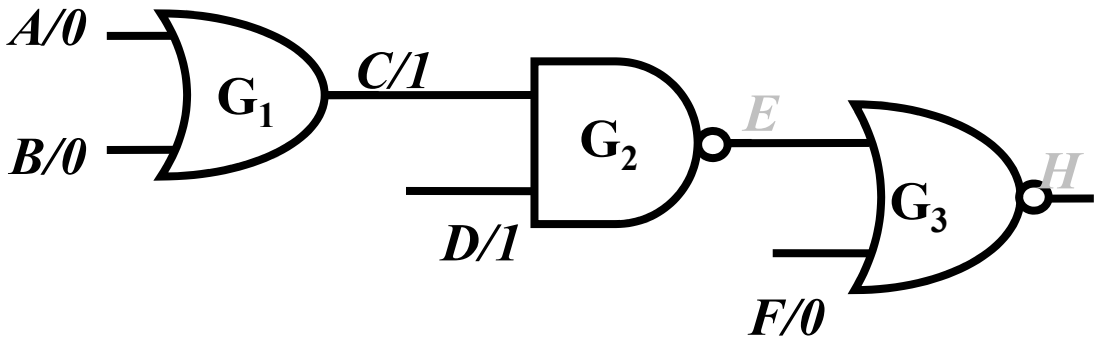
- Example: 2-input AND/OR/NAND/NOR gates
  - original 6 faults → 3 faults after DFC
- For *n*-input elementary gate,
  - *n*+1 stuck-at faults after dominance fault collapsing



- Q: is DFC unique?

DFC on Fanout-free Circuits

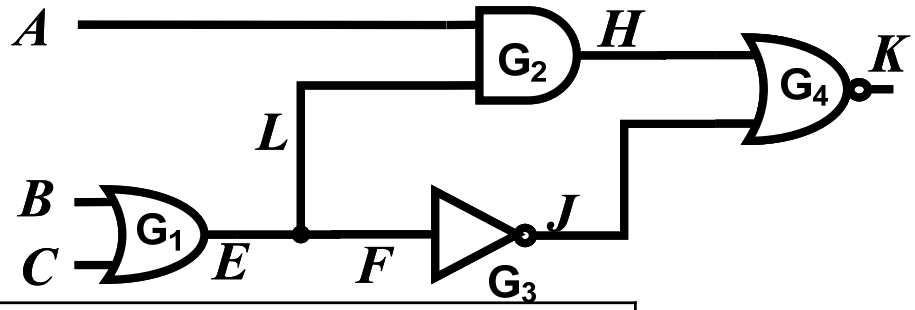
- Our DFC Rules
  - (1) one collapsed fault for every primary input
    - Stuck at non-controlling values (see p.19)
  - (2) one collapsed fault for each gate output whose gate inputs are *all* primary inputs
- Example
  - No fault is needed for G2 output because it is not an input gate.
  - E/0 fault dominates C/1 fault so the former can be removed.
  - Original 14 faults → 8 faults after EFC → 5 faults after DFC



  - Exercise: show H/0 can be collapsed

Fanout Stem and Branches

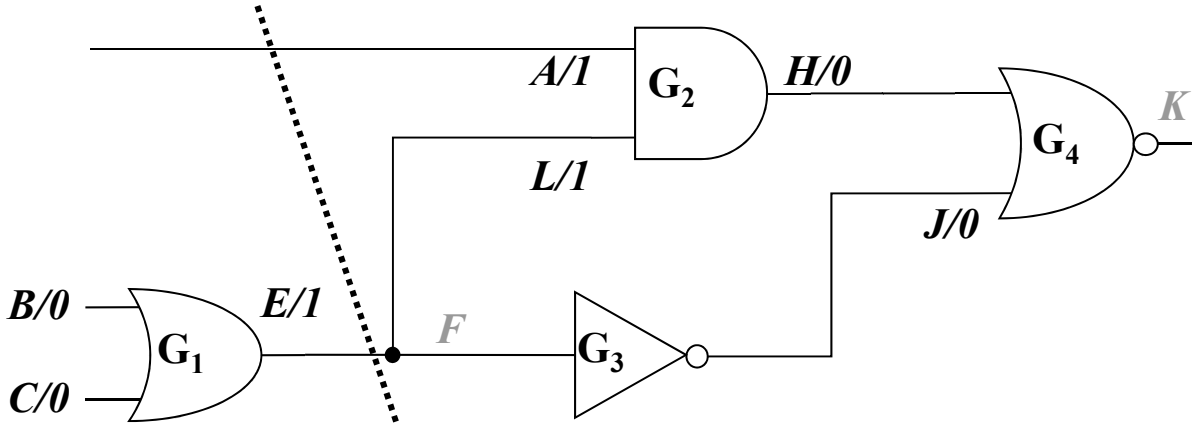
- Fanout branch faults do not always dominate stem faults
- Example
  - F/1 fault dominates E/1. but L/1 fault does not dominate E/1.
  - Actually, L/1 has empty detecting set so L/1 is a *redundant fault*



Input			Output						
A	*B*	*C*	good	*E/0*	*F/0*	*L/0*	*E/1*	*F/1*	*L/1*
0	0	0	0	0	0	0	1	1	0
0	0	1	1	0	0	1	1	1	1
0	1	0	1	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0	0	0
1	1	1	0	0	0	1	0	0	0

Example of Simple DFC

- Partition into 2 subcircuits
- Original 18 faults, → after EFC 10 faults → after DFC 7 faults
- Result is NOT optimal
  - J/0 is equivalent to F/1,which dominates E/1.
- Notice:
  - G_2 is input gate
  - J is consider input because inverter G_3 is ignored



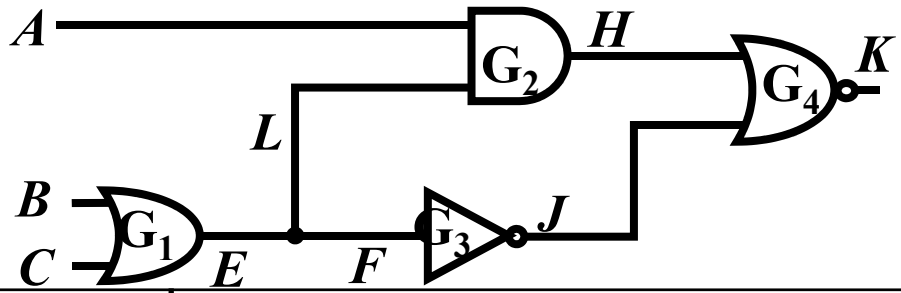
Simple DFC Algorithm

- **Linear time**
  - **No stem analysis**

```
simple_DFC (N)   /*N is a netlist*/
 0.  fault_list = {};
 1.  foreach gate or PI or PO g in N
 2.     if ((g is PI and fanout stem) || (g is PO and fanout branch)) then
 3.        fault_list = fault_list ∪ g output stuck-at 0 and 1;
 4.     else if (g is gate) then
 5.        foreach gate input i of gate g
 6.              h = backtrace inverters starting from i;
 7.              if (h is PI or fanout branch) then  /* rule #1 */
 8.                 if (gate g is AND) || (gate g is NAND) then
 9.                    fault_list = fault_list ∪ i stuck-at 1;
10.                 else if (gate g is OR) || (gate g is NOR)
11.                    fault_list = fault_list ∪ i stuck-at 0;
12.                 end if
13.              end if
14.        end foreach
15.        if (every gate input of g has a fault) then   /* rule #2 */
16.           if (gate g is AND) || (gate g is NOR) then
17.              fault_list = fault_list ∪ g output stuck-at 0;
18.           else if (gate g is OR) || (gate g is NAND) then
19.              fault_list = fault_list ∪ g output stuck-at 1;
20.           end if
21.        end if
22.     end if
23.  end foreach
24.  return (fault_list );
```

Why DFC not Used Often?

- Because DFC fault coverage is ***pessimistic.*** *A* test pattern can detect a dominating fault without detecting its dominated fault
- Example: ABC=100 does *NOT* detect dominated fault E/1
  - but it detects dominating fault F/1 (but not in DFC fault list)



Input			Output						
A	*B*	*C*	good	*E/0*	*F/0*	*L/0*	*E/1*	*F/1*	*L/1*
0	0	0	0	0	0	0	1	1	0
0	0	1	1	0	0	1	1	1	1
0	1	0	1	0	0	1	1	1	1
0	1	1	1	0	0	1	1	1	1
1	0	0	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0	0	0
1	1	0	0	0	0	1	0	0	0
1	1	1	0	0	0	1	0	0	0

Fault Collapsing

- Introduction
- Equivalence Fault Collapsing
  - Fanout-free circuits
  - Fanout
- Dominance Fault Collapsing
  - Fanout-free circuits
  - Circuits with fanout
- Conclusion

Concluding Remarks

- **Fault collapsing helps**
  - **Speed up ATPG**
  - **Shorten test length**
- **EFC is often used but DFC is not**
- **Fault coverage often cited in two numbers**
  - ***Uncollapsed fault coverage***

$$F.C._{uncollapsed} = \frac{\text{detected } uncollapsed \text{ faults}}{\text{total } uncollapsed \text{ faults}} \times 100\%$$

  - ***Collapsed fault coverage*** **(usually EFC)**

$$F.C._{collapsed} = \frac{\text{detected } collapsed \text{ faults}}{\text{total } collapsed \text{ faults}} \times 100\%$$

Example ATPG Report

	uncollapsed	Collapsed
Total Faults	1,234	800
Detected faults	1,000	700
Untestable faults	230	98
aborted faults	4	2
Fault Coverage	1,000/1,234 %	700/800 %
ATPG effectiveness	1,000/1,004 %	700/702 %