

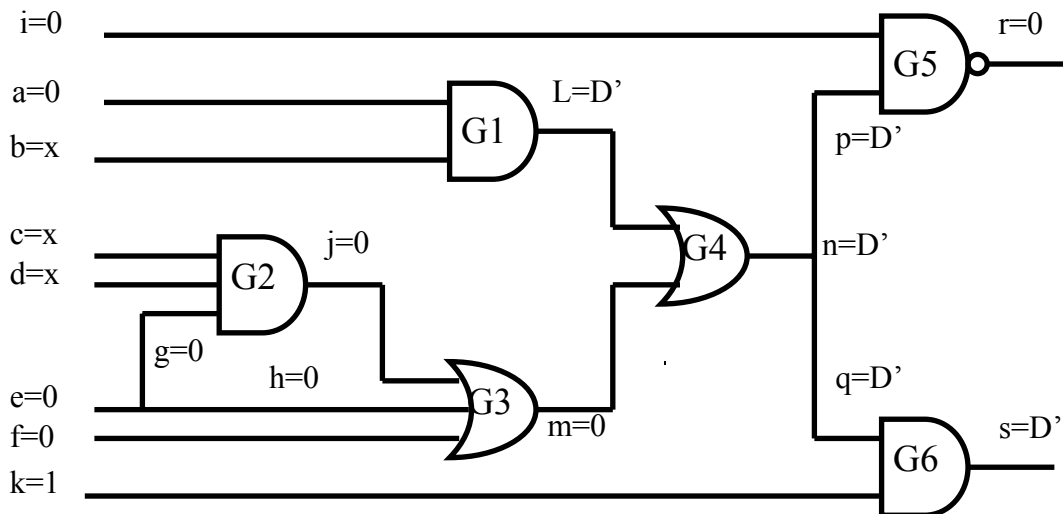
18-360

Intro to Computer-Aided Design

Homework 5 Solutions

1. Do the implications – the non-PODEM approach

Implication	Why implicated?
$m = 0$	Non-controlling value for propagation
$n = D'$	$L = D'$, $m = 0$
$p = D'$	$n = D'$ – fanout branch of n
$q = D'$	$n = D'$ – fanout branch of n
$r = 0$	$i = 0$ – i controls the output
$k = 1$	Non-controlling value for propagation to s
$s = D'$	$q = D'$, $k = 1$
$j = 0$	Justify $m = 0$
$h = 0$	Justify $m = 0$
$f = 0$	Justify $m = 0$
$e = 0$	$h = 0$ – fanout stem of e
$g = 0$	$e = 0$ – fanout branch of e
$a = 0$ (or $b = 0$)	Justify $L = 0$



2. You-da-Code: The PODEM Experience

a. i s-a-1

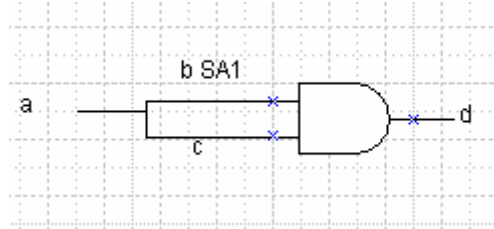
Objective	PI assignment	Implications	D Frontier	comments
i = 0	b = 0	f = 0, i = D', h = 0, j = 1	m	Activated fault
g = 0	d = 1	-	m	Trying to propagate the fault along m
g = 0	e = 1	g = 0, m = D	n	Propagated thru gate m
k = 1	a = 1	k = 1, n = D'	-	Propagate fault to output

b. i s-a-0

Objective	PI assignment	Implications	D Frontier	comments
i = 1	b = 1	-	-	Trying to activate fault
i = 1	c = 1	f = 1, h = 1, j = 0, k = 0, n = 1, i = D	m	Activated fault, but output is 1, so can't propagate, but let's still continue to show what PODEM will do. The second test in PODEM will fail.
	c = 0	f = 0, i = 0, h = 0, j = 1	-	Level 1 Backtrack - Failed to activate fault
	c = x	-	-	Failed decision
	b = 0	f = 0, i = 0, h = 0, j = 1		Level 2 Backtrack – reached first decision. Again failed to activate fault
	b = x	-	-	First decision failed

No test is obtained. The fault is not detectable – redundant.

3. Inside PODEM



Consider the above figure. We are trying to test for the fault b s-a-1.

PODEM first tries to set b to 0. In order to do this, it looks at the primary input a, and sets it to 0. This implies $b = D'$ and $c = 0$ and $d = 0$. So no test is possible with $a = 0$.

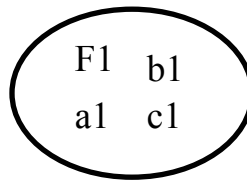
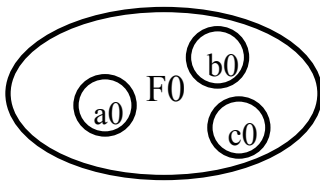
PODEM now backtracks and sets a to 1. This is when PODEM reaches point A in the algorithm.

To continue, the algorithm will realize that the fault cannot be activated when $a=1$, so no test is possible. In this circuit, b s-a-1 is redundant.

4. What dominates what?

$F = (a \wedge b) + (b \wedge c) + abc = a + b + c$. Below is the truth table, as well as the result of fault simulation.

a b c	F	a0	a1	b0	b1	c0	c1	F0	F1
0 0 0	0		X		X		X		X
0 0 1	1					X		X	
0 1 0	1							X	
0 1 1	1			X				X	
1 0 0	1	X						X	
1 0 1	1							X	
1 1 0	1							X	
1 1 1	1							X	



The only test vector for a0 is also one of the test vectors for F0, so F0 dominates a0. Similarly, F0 dominates b0 and c0.

F1 is functionally equivalent to a1, b1, and c1. So, any one of them dominates the others.

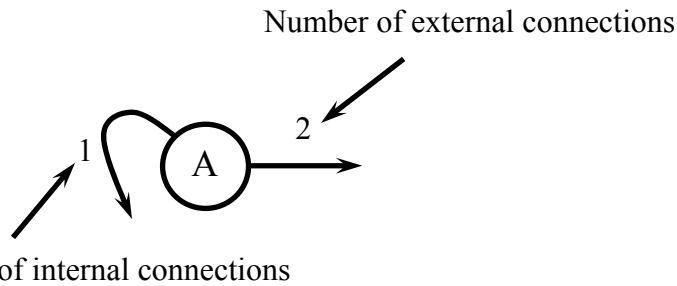
5. Partition it – K&L

The connectivity matrix is

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	
<i>A</i>	0	0	0	1	1	1	0	0	3
<i>B</i>	0	0	0	1	0	0	1	0	2
<i>C</i>	0	0	0	0	1	0	1	1	3
<i>D</i>	1	1	0	0	2	2	2	0	8
<i>E</i>	1	0	1	2	0	2	1	1	8
<i>F</i>	1	0	0	2	2	0	1	0	6
<i>G</i>	0	1	1	2	1	1	0	2	8
<i>H</i>	0	0	1	0	1	0	2	0	4

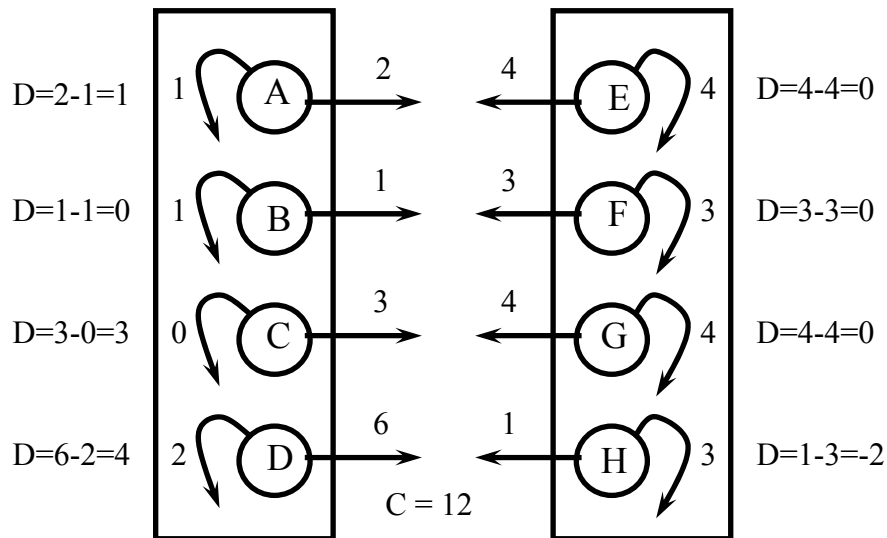
The rightmost values are the total numbers of both external and internal connections for individual nodes.

Notations:



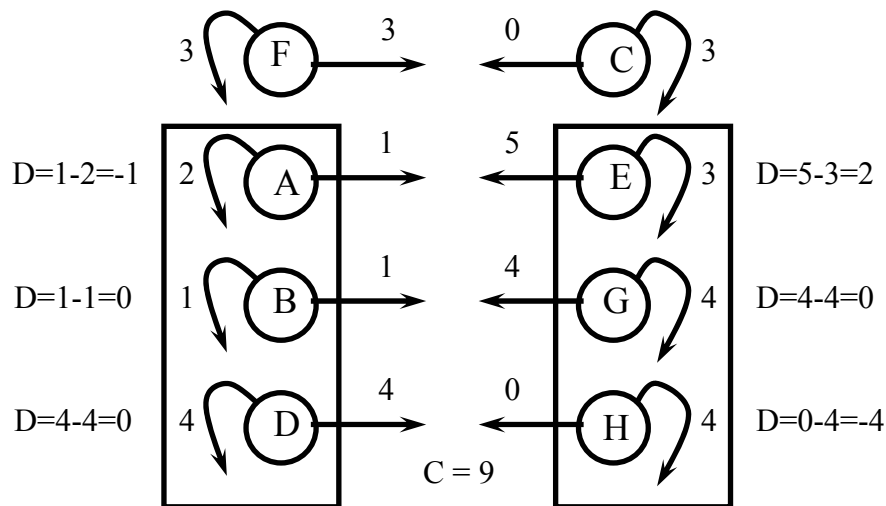
Step 0:

The initial partition is shown below, with the “gain” calculated for each node. The initial total gain is 0, and the number of nets crossing the cut line is $12 = (2 + 1 + 3 + 6 + 4 + 3 + 4 + 1)/2$.



Gain of swapping A & E = $1 + 0 - 2 = -1$;
 Gain of swapping A & F = $1 + 0 - 2 = -1$;
 Gain of swapping A & G = $1 + 0 - 0 = 1$;
 Gain of swapping A & H = $1 - 2 - 0 = -1$;
 Gain of swapping B & E = $0 + 0 - 0 = 0$;
 Gain of swapping B & F = $0 + 0 - 0 = 0$;
 Gain of swapping B & G = $0 + 0 - 2 = -2$;
 Gain of swapping B & H = $0 - 2 - 0 = -2$;
 Gain of swapping C & E = $3 + 0 - 2 = 1$;
 Gain of swapping C & F = $3 + 0 - 0 = 3$;
 Gain of swapping C & G = $3 + 0 - 2 = 1$;
 Gain of swapping C & H = $3 - 2 - 2 = -1$;
 Gain of swapping D & E = $4 + 0 - 4 = 0$;
 Gain of swapping D & F = $4 + 0 - 4 = 0$;
 Gain of swapping D & G = $4 + 0 - 4 = 0$;
 Gain of swapping D & H = $4 - 2 - 0 = 2$;

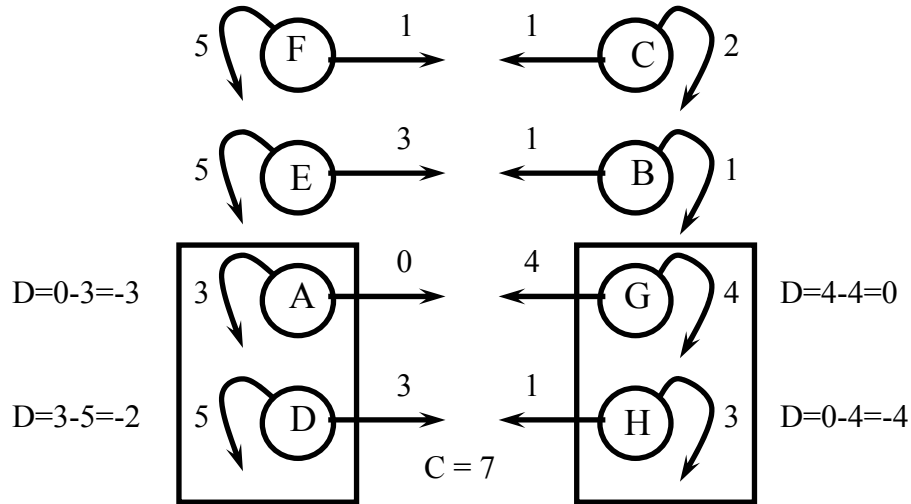
The maximum gain is 3. Swap C & F and lock them, and get the partition shown below. The total gain after this swap is $0 + 3 = 3$, and the number of nets is reduced to 9. The external and internal connections, and gain for each node are also updated.



Step 1:

Gain of swapping A & E = $-1 + 2 - 2 = -1$;
 Gain of swapping A & G = $-1 + 0 - 0 = -1$;
 Gain of swapping A & H = $-1 - 4 - 0 = -5$;
 Gain of swapping B & E = $0 + 2 - 0 = 2$;
 Gain of swapping B & G = $0 + 0 - 2 = -2$;
 Gain of swapping B & H = $0 - 4 - 0 = -4$;
 Gain of swapping D & E = $0 + 2 - 4 = -2$;
 Gain of swapping D & G = $0 + 0 - 4 = -4$;
 Gain of swapping D & H = $0 - 4 - 0 = -4$;

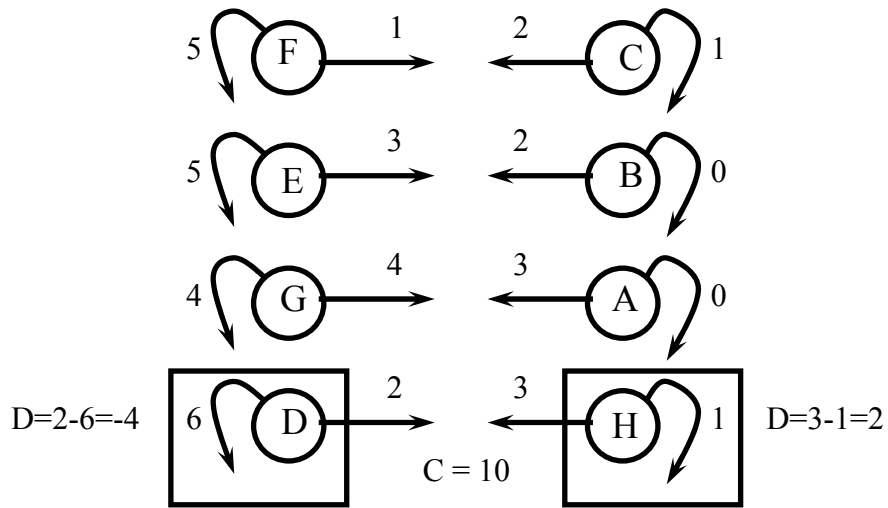
The maximum gain is 2. Swap B & E, and lock them, and get the partition shown below. The total gain after this swap is $0 + 3 + 2 = 5$, and the number of nets is reduced to 7. The external and internal connections, and gain for each node are also updated.



Step 2:

Gain of swapping A & G = $-3 - 0 - 0 = -3$;
 Gain of swapping A & H = $-3 - 4 - 0 = -7$;
 Gain of swapping D & G = $-2 + 0 - 4 = -6$;
 Gain of swapping D & H = $-2 - 4 - 0 = -6$;

The maximum gain is -3. Swap A & G, and lock them, and get the partition shown below. The total gain after this swap is $0 + 3 + 2 - 3 = 2$, and the number of nets is increased to 10. The external and internal connections, and gain for each node are also updated.



Step 3:

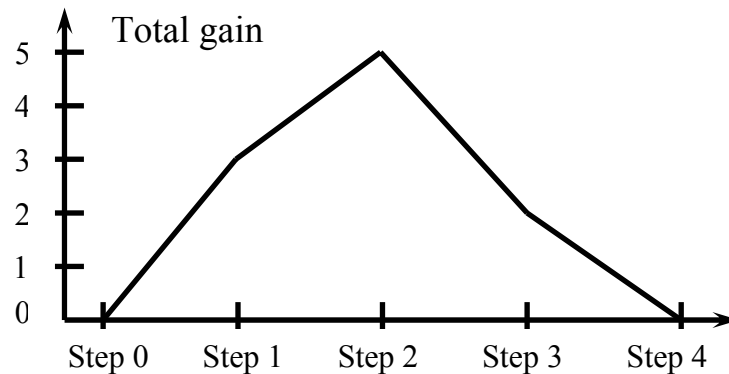
Gain of swapping D & H = $-4 + 2 - 0 = -2$.

The gain is -2. Swap D & H, and lock them, and get to the original partition. The total gain after this swap is $0 + 3 + 2 - 3 - 2 = 0$, and the number of nets is restored to 12.

So the sequence of swapping is

- (1) C & F;
- (2) B & E;
- (3) A & G;
- (4) D & H.

The total gain curve is



So for the best partition, we only need to commit the first two swaps, i.e., C & F and B & E, to minimize the number of nets crossing the cut line.

We unlock all the nodes and carry out extra K&L iterations similarly. For the second pass,

First swap: B & E, gain = -2, total = -2, number of nets crossing the cut line is increased to 9;

Second swap: A & G, gain = -1, total gain = -3, number of nets is increased to 10;

Third swap: F & H, gain = 0, total gain = -3, number of nets is still 10;

Fourth swap: C & D, gain = 3, total gain = 0, number of nets is 7. This gets the best solution, which is, however, the same as the beginning solution of the second pass.

Therefore, an extra pass doesn't help to reduce the number of nets crossing the cut line.