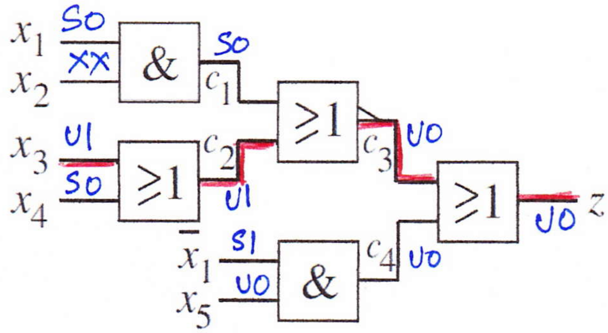
3.(i) Robust Pattern

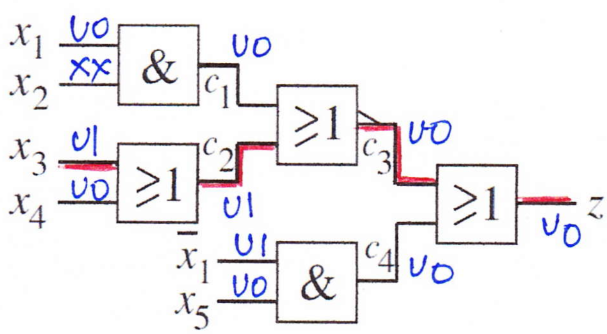


$\uparrow x_3 c_2 c_3 z$

∴ Test pattern sequence :

(0x00x, 0x100)

Non-robust Pattern

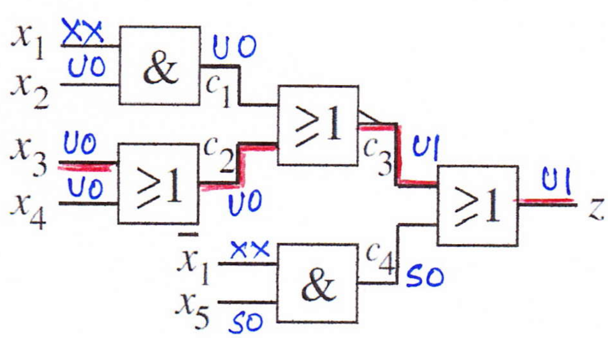


$\uparrow x_3 c_2 c_3 z$

∴ Test pattern sequence :

(xx0xx , 0x100)

3(ii) Robust Pattern

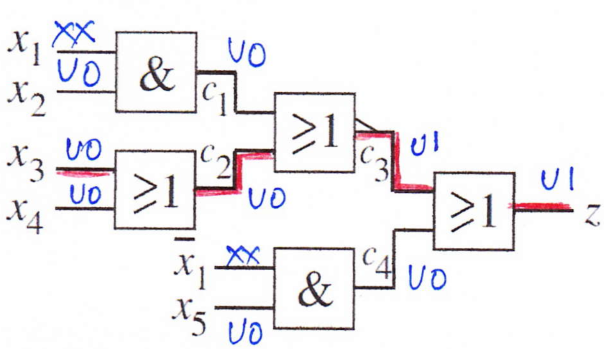


$\downarrow x_3 c_2 c_3 z$

∴ Test pattern sequence :

(xx1x0, x0000)

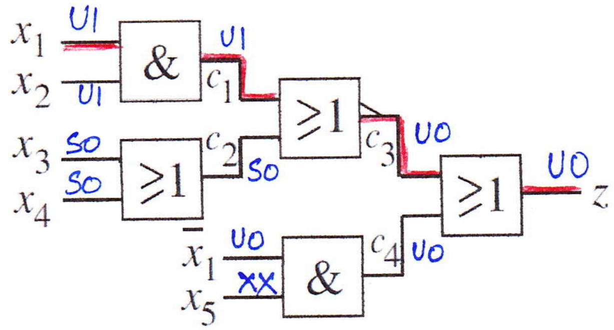
Non-robust pattern



$\downarrow x_3 c_2 c_3 z$

(xx1xx, x0000)

3.(iii) Robust Pattern

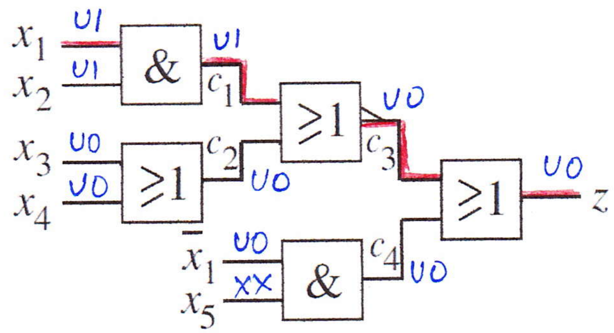


$\uparrow x_1 c_1 c_3 z$

$\therefore$ Test pattern sequence:

$(0X00X, 1100X)$

Non-robust pattern

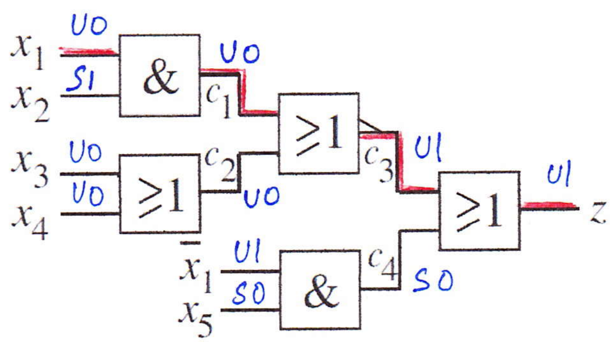


$\uparrow x_1 c_1 c_3 z$

$\therefore$ Test pattern sequence:

$(0XXXX, 1100X)$

3.(iv) Robust pattern

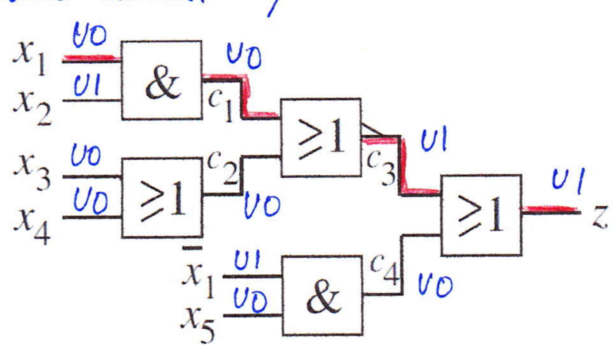


$\downarrow x_1 c_1 c_3 z$

$\therefore$ Test pattern sequence:

$(11XX0, 01000)$

Non-robust pattern

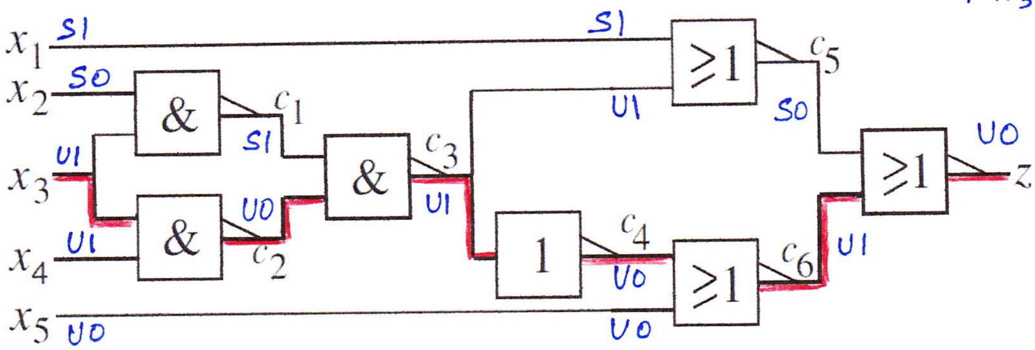


$\downarrow x_1 c_1 c_3 z$

$\therefore$ Test pattern sequence:

$(1XXXX, 01000)$

4.(i) Robust pattern

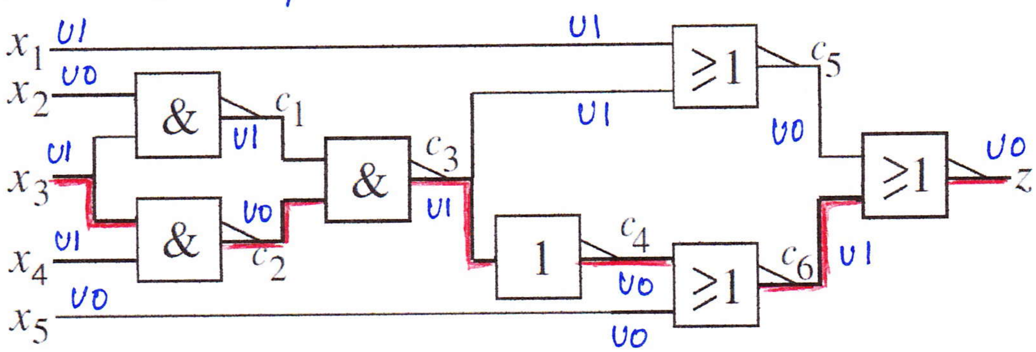


↑ $x_3 c_2 c_3 c_4 c_6 z$

∴ Test vector:

(100XX, 10110)

~~(ii)~~ Non-robust pattern

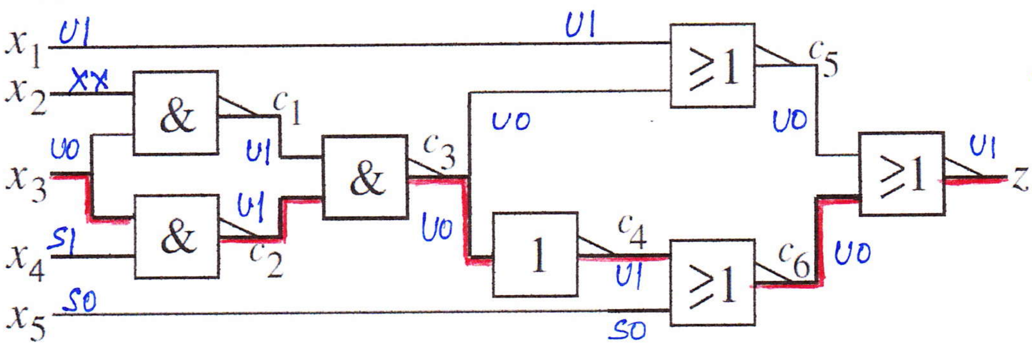


↑ $x_3 c_2 c_3 c_4 c_6 z$

∴ Test vector:

(XX0XX, 10110)

4 (ii) Robust Pattern

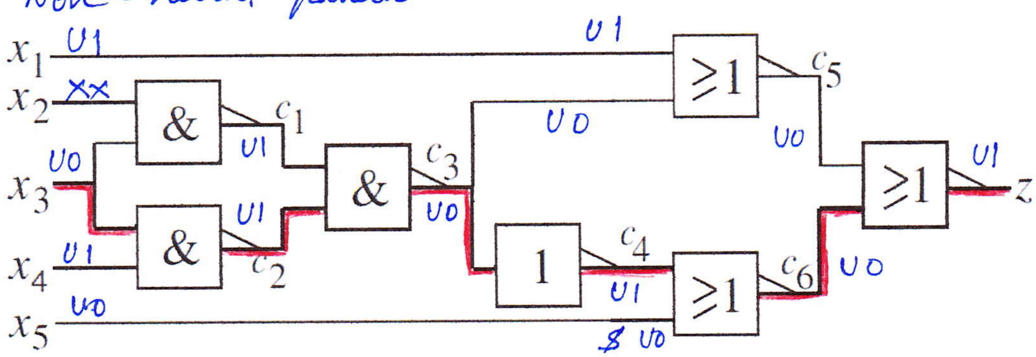


↓ $x_3 c_2 c_3 c_4 c_6 z$

∴ Test vector:

(XX110, 1X010)

Non-robust pattern

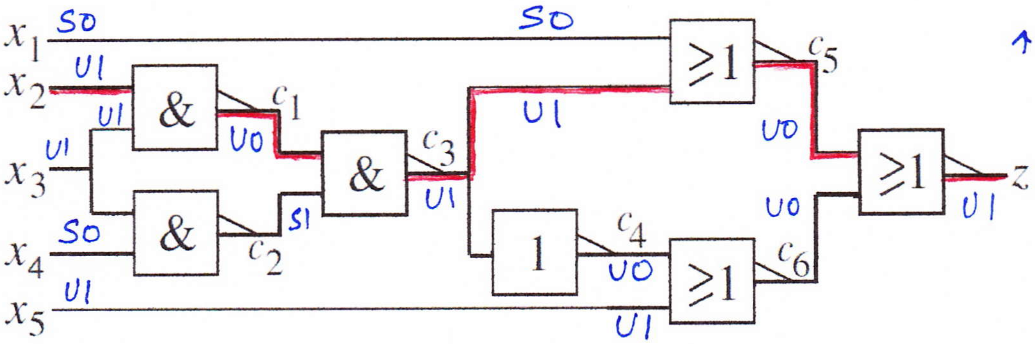


↓ $x_3 c_2 c_3 c_4 c_6 z$

∴ Test vector:

(XX1XX, 1X010)

4 (iii) Robust pattern

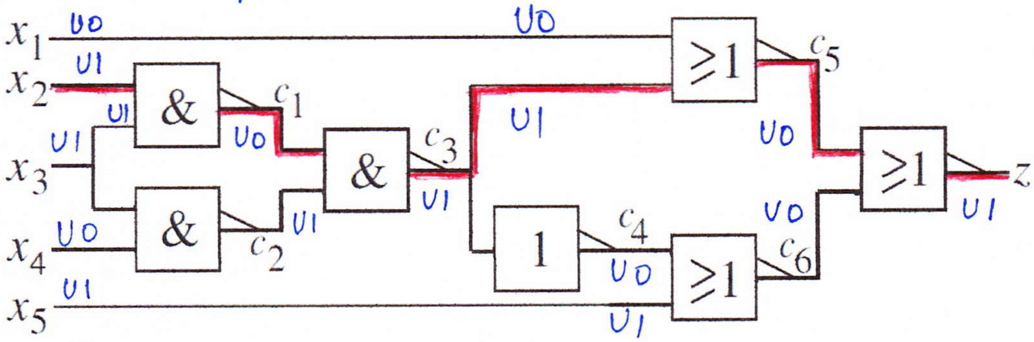


↑ $x_2 c_1 c_3 c_5 z$

∴ Test vector:

(00X0X, 01101)

Non-robust pattern

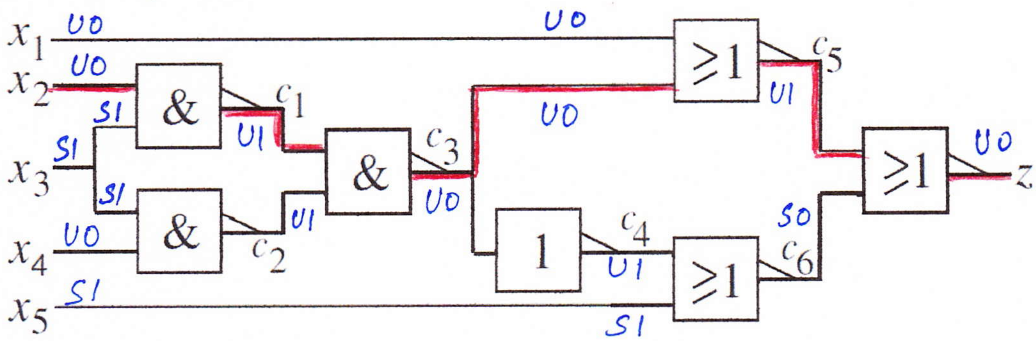


↑ $x_2 c_1 c_3 c_5 z$

∴ Test vector:

(X0XXX, 01101)

4·(iv) Robust pattern



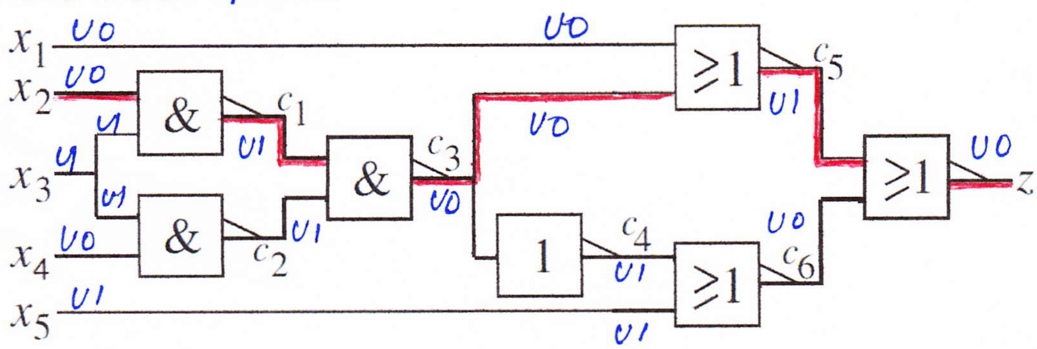
↓ $x_2 c_1 c_3 c_5 z$

∴ Test vector:

~~(Xxxx~~

(X11X1, 00101)

Non-robust pattern



↓ $x_2 c_1 c_3 c_5 z$

∴ Test vector:

(X1XXX, 00101)

[redacted] *Problem 1– continued*

Problem 14.4 of textbook

Problem 14.4: Proof that March C- detects all unlinked CFdss

March C- algorithm:	{ $\Updownarrow$(w0);	$\Uparrow$(r0, w1);	$\Uparrow$(r1, w0);	$\Downarrow$(r0, w1);	$\Downarrow$(r1,w0);	$\Updownarrow$(r0)	}
March elements:	M0	M1	M2	M3	M4	M5	

Note: The fault <r0; ↓>av means that the r0 operation applied to the a-cell causes a ↓ transition in the v-cell; for the case a<v (i.e., the address of the a-cell is lower than the address of the v-cell).

The fault <w1;↓>va means that the w1 applied to the a-cell causes a ↓ transition in the v-cell, for v<a.

Operation on a-cell	Disturbance of v-cell	Fault detected by march element, or march pair	
		a < v	a > v
r0	↓	M3-M4	M1-M2
r0	↑	M1	M3
r1	↓	M2	M4
r1	↑	M4-M5	M2-M3
w0	↓	M2	M4
w0	↑	M4-M5	M2-M3
w1	↓	M3-M4	M1-M2
w1	↑	M1	M3

From inspecting the table, one can verify that all CFdss are detected by March C-.

[redacted]

[redacted]

[redacted]

Problem 14.5 of textbook

Consider the following MATS+ scheme

$$\{\Updownarrow(w0)\,;\ \Uparrow(r0, w1)\,;\ \Downarrow(r1, w0)\}$$

Expansion B-cells/word the following scheme is applied

$$\{\Updownarrow(w_a)\,;\ \Uparrow(r_a, w_b)\,;\ \Downarrow(r_b, w_a)\}$$

where b is complement of a, 'a' & 'b' are called background word

also written as; [redacted]

$$\{\Updownarrow(w_{00\ldots0})\,;\ \Uparrow(r_{00\ldots0}, w_{11\ldots1})\,;\ \Downarrow(r_{11\ldots1}, w_{00\ldots0})\}$$

* Since SAF is related to single cell, SAFs can be detected only

* Since ($r_{x\ldots x}$ —— $w_{\bar{x}\ldots\bar{x}}$) and ($r_{\bar{x}\ldots\bar{x}}$ —— $w_{x\ldots x}$) are satisfied with M1 & M2 it can detect AFs

Ex: Example of a MATS+ algo for a memory with B=4

consider the foll scheme.

$$\{\underset{M0}{\Updownarrow(w_{0000})}\,;\ \underset{M1}{\Uparrow(r_{0000}, w_{1111})}\,;\ \underset{M2}{\Downarrow(r_{1111}, w_{0000})}\}.$$

The above scheme can detect SA-0 & SA1

4 Zeroes are expected which will detect SA1 in any cell

Illy SA0 is detected with M1 & M2.

[redacted]- Problem 1– continued

Problem 14.6 of textbook

Problem 14.6: The restrictions for march tests when applied to FIFOs are:

1. The RA (WA) is automatically *incremented* upon completion of a Read (Write) operation; hence, only the $\Uparrow$ address order can be used in march elements.
2. Because of the fact that the RA and WA automatically increment upon completion of every Read (Write) operation, a march element can only contain a single Read and/or a single Write operation; i.e., only march elements of the following forms are possible:
   $\Uparrow(rx)$, $\Uparrow(wx)$, $\Uparrow(rx, wy)$, $\Uparrow(wy, rx)$; where $x, y \in \{0, 1\}$.

E.g., $\Uparrow(w0)$, $\Uparrow(r0, w1)$, $\Uparrow(r1, w0)$, $\Uparrow(w0, r1)$, etc., are examples of possible march elements. The consequence of these restrictions is that not all AFs and not all CFs will be detectable.

7.) Problem 14.7 of textbook:

→ Memory is one-bit wide, thus, the following take place:
$A[0] = 0$, $A[1] = 1$, $A[2] = 0$, ... $A[256] = 0$ ... $A[1023] = 1$

→ Read $A[I]$ and write the complement.
$A[0] = 1$, $A[1] = 0$, $A[2] = 1$, ... $A[256] = 1$ .... $A[1023] = 0$

→ Read $A[I]$

(i) To detect unlinked AFs, either $(rx \ldots w\bar{x})$ or $(r\bar{x} \ldots wx)$ is used. If $(r0, w1)$ was used only $A[0]$, $A[2]$ were detected. Thus, all the AFs cannot be detected.

(ii) For Stuck-At faults: SA0: $\{ \Updownarrow w0; \Uparrow(w1, r1) \}$ is used and SA1: $\{ \Updownarrow w1; \Uparrow(w0, r0) \}$ is used. Thus, all the SAFs can be detected.

(iii) Since, there is only one transition, all TFs cannot be detected.

(iv) All the CFs cannot be detected by this test. e.g $\langle 0; 1 \rangle$ cannot be detected.

[redacted] Problem 1– continued

Problem 14.8 of textbook

Problem 14.8: Design a minimal march test which detects the following faults:

1. Linked CFids of the form $<\uparrow;\downarrow>a_1v \# <\uparrow;\uparrow>a_2v$
   Note: For the set of linked CFids of the form: $<\uparrow;\downarrow>a_1v \# <\uparrow;\uparrow>a_2v$ holds that $a1<v$ and $a2<v$; in addition the set consists of two members: 1. $a1<a2$ and 2. $a2<a1$. This fault will be detected by the test: $\{\Updownarrow(w0); \Uparrow(r0, w1); \Uparrow(r1, w0, w1)\}$

2. Linked CFids of the form $<\uparrow;\downarrow> \# <\uparrow;\uparrow>$
   Note: For the set of linked CFids of the form $<\uparrow;\downarrow> \# <\uparrow;\uparrow>$ holds that the a-cells may take on any position relative to the v-cell; the set therefore consists of six members. This fault will be detected by the test: $\{\Updownarrow(w0); \Uparrow(r0, w1); \Uparrow(r1, w0, w1); \Downarrow(r1, w0, w1); \Updownarrow(w0); \Downarrow(r0, w1)\}$

Problem 5

Faults	Sensitizes	Detects
AF	M2	M3
	M1	M4
SAF	M0	M1
	M1	M2
TF	M1	M2
	M2	M3
CFin $(a < v, \langle\uparrow; \updownarrow\rangle)$	M1	M1
CFin $(a < v, \langle\downarrow; \updownarrow\rangle)$	M2	M2
CFin $(a > v, \langle\uparrow; \updownarrow\rangle)$	M3	M3
CFin $(a > v, \langle\downarrow; \updownarrow\rangle)$	M4	M4
CFid $(a < v, \langle\uparrow; 0\rangle)$	M3	M4
CFid $(a < v, \langle\uparrow; 1\rangle)$	M1	M1
CFid $(a < v, \langle\downarrow; 0\rangle)$	M2	M2
CFid $(a < v, \langle\downarrow; 1\rangle)$	M4	M5
CFid $(a > v, \langle\uparrow; 0\rangle)$	M1	M4
CFid $(a > v, \langle\uparrow; 1\rangle)$	M3	M3
CFid $(a > v, \langle\downarrow; 0\rangle)$	M4	M4
CFid $(a > v, \langle\downarrow; 1\rangle)$	M2	M3
CFst$(a < v, \langle 0; 0\rangle)$	M2	M2
CFst$(a < v, \langle 0; 1\rangle)$	M4	M5
CFst$(a < v, \langle 1; 0\rangle)$	M3	M4
CFst$(a < v, \langle 1; 1\rangle)$	M1	M1
CFst$(a > v, \langle 0; 0\rangle)$	M4	M4
CFst$(a > v, \langle 0; 1\rangle)$	M2	M3
CFst$(a > v, \langle 1; 0\rangle)$	M1	M2
CFst$(a > v, \langle 1; 1\rangle)$	M3	M3
linked CFid	Cannot detect	Cannot detect

Problem 6

Faults	Sensitizes	Detects
AF	M5	M5
SAF	M1	M1
	M2	M2
TF	M1	M1
	M2	M2
CFin $(a < v, \langle\uparrow; \updownarrow\rangle)$	M1	M1
CFin $(a < v, \langle\downarrow; \updownarrow\rangle)$	M2	M2
CFin $(a > v, \langle\uparrow; \updownarrow\rangle)$	M3	M3
CFin $(a > v, \langle\downarrow; \updownarrow\rangle)$	M4	M4
CFid $(a < v, \langle\uparrow; 0\rangle)$	M3	M4
CFid $(a < v, \langle\uparrow; 1\rangle)$	M1	M1
CFid $(a < v, \langle\downarrow; 0\rangle)$	M2	M2
CFid $(a < v, \langle\downarrow; 1\rangle)$	M4	M5
CFid $(a > v, \langle\uparrow; 0\rangle)$	M1	M2
CFid $(a > v, \langle\uparrow; 1\rangle)$	M3	M3
CFid $(a > v, \langle\downarrow; 0\rangle)$	M4	M4
CFid $(a > v, \langle\downarrow; 1\rangle)$	M2	M5
CFst$(a < v, \langle 0; 0\rangle)$	M2	M2
CFst$(a < v, \langle 0; 1\rangle)$	M2	M3
CFst$(a < v, \langle 1; 0\rangle)$	M3	M4
CFst$(a < v, \langle 1; 1\rangle)$	M1	M1
CFst$(a > v, \langle 0; 0\rangle)$	M1	M1
CFst$(a > v, \langle 0; 1\rangle)$	M2	M2
CFst$(a > v, \langle 1; 0\rangle)$	M4	M4
CFst$(a > v, \langle 1; 1\rangle)$	M3	M3
linked CFid	M2	M3
	M1	M4
	M4	M5
	M3	M6