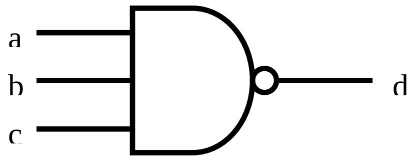
Colapsul defectelor şi setul minim de teste

Se dă circuitul:



Se cere să se determine setul minim de teste pentru defectele de tip blocat-la singulare din acest circuit.

Modelul de defect „blocat-la singular”

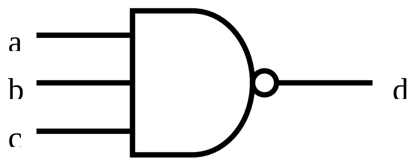
„Blocat-la” înseamnă linie scurtcircuitată la şi având valoarea de zero sau unu logic, fără să se considere valori intermediare sau instabile.
„Singular” înseamnă că circuitul ori este bun, fără defect, ori are un singur defect. De exemplu, linia *a* ar putea fi blocată la zero, iar celelalte linii şi restul circuitului sunt perfect funcţionale. Dacă se presupune că, de exemplu, linia *b* este blocată la unu, se consideră că restul liniilor, inclusiv *a*, sunt fără defect.

Observaţi că nu se face diagnoză, pentru identificarea defectului, ci se presupune defectul, apoi trebuie confirmată sau infirmată prezenţa lui.

Desigur, modelul de defect „blocat-la singular” nu este foarte realist, deoarece, pe de o parte, pot apare defecte „instabile”, adică scurt la zero, unu sau valori intermediare, iar pe de altă parte, într-un circuit integrat, de obicei defectele nu apar singure, ci în grup. Totuşi, acest model de defect este clasic, bine implementat de programele şi echipamentele de testare şi este o bază bună atât ca prima fază a testării unui circuit, cât şi ca baza altor modele de defect.

Teste

Circuitul (poartă ŞI-NU cu trei intrări), din nou:



Tabelul de adevăr pentru acest circuit:

abc	d
000	1
001	1
010	1
011	1
100	1
101	1
110	1
111	0

Puteţi calcula uşor ieşirea circuitului: deoarece este o poartă ŞI-NU, ieşirea va fi ca la o poartă ŞI, doar inversată. La o poartă ŞI, pentru majoritatea combinaţiilor de intrare, ieşirea este 0. Situaţia „specială”, sau „excepţia”, este când toate intrările au valoarea 1, atunci ieşirea va fi 1. Aşadar, la o poartă ŞI-NU, cum este aici, excepţia este când toate intrările sunt 1 şi ieşirea este 0, pentru restul combinaţiilor de intrare ieşirea va fi 1.
Desigur, această metodă rapidă nu este obligatorie, tabelul se poate calcula şi băbeşte, adică se pot lua pe rând combinaţiile de intrări şi se poate calcula ieşirea. Este şi mai sigur aşa.

De regulă, în timpul proiectării circuitului, se proiectează şi metodele cu care se vor testa circuitele fabricate pe baza proiectului. Motivul este că în momentul când primul circuit iese de pe bandă, el trebuie testat, ceea ce presupune că metoda de testare este deja pusă la punct şi documentată.

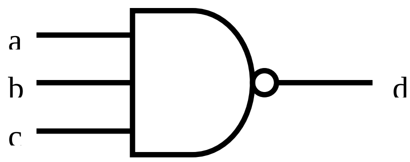
Noi aici încercăm să punem la punct metoda cu care să testăm diferitele defecte de tip blocat-la singulare care ar putea să fie în circuitul de mai sus, mai exact în circuitele fabricate pe baza schemei circuitului de mai sus.

Problema este că, deşi structura circuitului este clară pe ecran, noi nu avem acces în interiorul circuitului fizic care va fi fabricat, deoarece este circuit integrat. De exemplu, dacă presupunem că linia *a* este blocată la zero şi dorim să verificăm dacă această presupunere este adevărată sau nu (sperăm să nu fie adevărată), nu putem măsura tensiuni în interiorul circuitului, decât pe pinii capsulei integratului, adică *a*, *b*, *c* şi *d*, dar nu în proximitatea porţii, unde apare defectul, ci doar pe intrările şi ieşirile primare.

Tensiunile (valorile logice) de pe intrările primare nu are rost să le măsurăm, deoarece sunt puncte de controlabilitate, noi punem tensiunile acolo, sigur le cunoaştem. Singurul punct de observabilitate, unde putem să măsurăm ca să înţelegem ce se întâmplă în circuitul real, fabricat, este ieşirea *d*.

De aceea, dacă presupunem *a* b-l-0 (blocat-la zero), singurul mod de a verifica dacă presupunerea este adevărată sau nu este să punem o combinaţie binară bine aleasă pe intrările *a*, *b* şi *c*, să măsurăm valoarea logică de pe ieşirea *d* şi din această valoare a ieşirii să afirmăm prezenţa sau absenţa defectului a_0 (*a* b-l-0). Trucul constă în alegerea corectă a combinaţiei de intrare, pe care o vom numi *test*, de acum încolo.

Circuitul, din nou:



Completăm tabelul de adevăr:

abc	d	a_0
000	1	1
001	1	1
010	1	1
011	1	1
100	1	1
101	1	1
110	1	1
111	0	1

Noua coloană a_0 conţine tot valori ale ieşirii *d*, dar pentru situaţia în care *a* este blocat la zero. Dacă *a* este blocat la zero, orice combinaţie am pune pe intrările circuitului, la prima intrare (linia *a*) a porţii din integrat va ajunge zero, deoarece ea este blocată la zero. După cum se vede în ultimul rând al tabelului, problema apare la testul 111, când în circuit injectăm această combinaţie dar, din cauza *a* b-l-0, în poartă intră combinaţia 011, la care poarta dă la ieşire (linia *d*) 1.

De fapt, în cazul circuitului nostru, este foarte simplu să calculăm ieşirea în cazul *a* b-l-0: este 1 pe toată coloana, deoarece la orice poartă ŞI sau ŞI-NU, dacă cel puţin una din intrări este 0, ieşirea va fi 0, indiferent de celelalte intrări.
Similar, la orice poartă SAU sau SAU-NU, dacă cel puţin o intrare este 1, ieşirea va fi 1.

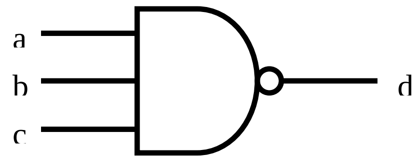
Completăm tabelul pentru celelalte defecte posibile, de tip blocat-la singular:

abc	d	a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1
000	1	1	1	1	1	1	1	0	1
001	1	1	1	1	1	1	1	0	1
010	1	1	1	1	1	1	1	0	1
011	1	1	0	1	1	1	1	0	1
100	1	1	1	1	1	1	1	0	1
101	1	1	1	1	0	1	1	0	1
110	1	1	1	1	1	1	0	0	1
111	0	1	0	1	0	1	0	0	1

Calcul rapid:
- a_0: este deja calculat; se putea calcula ca „albastru” de mai jos;
- la circuitul nostru, dacă o intrare este blocată la zero, ieşirea va fi 1 (albastru);
- a_1: la testele unde a=1, ieşirea cu defectul a_1 are aceeaşi valoare ca ieşirea fără defect d (roşu);
- b_1: la rândurile cu testele unde b=1, copiem ieşirea fără defect (verde);
- c_1: unde c=1 în test, putem copia ieşirea fără defect (portocaliu);
- d_0 şi d_1: dacă d (ieşirea) este blocată la zero, ieşirea circuitului va fi…zero (galben); la fel se „calculează” d_1;
- restul valorilor ieşirii, pentru diferite defecte şi teste, trebuie calculate (mov).

Nu uitaţi că toate valorile din tabelul de mai sus, în afară de tripleţii abc, sunt valori ale ieşirii *d* pentru diferite combinaţii de intrare abc şi defecte blocat-la singulare.

Circuitul, din nou:



Ultimul tabel, din nou:

abc	d	a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1
000	1	1	1	1	1	1	1	**0**	1
001	1	1	1	1	1	1	1	**0**	1
010	1	1	1	1	1	1	1	**0**	1
011	1	1	**0**	1	1	1	1	**0**	1
100	1	1	1	1	1	1	1	**0**	1
101	1	1	1	1	**0**	1	1	**0**	1
110	1	1	1	1	1	1	**0**	**0**	1
111	0	**1**	0	**1**	0	**1**	0	0	**1**

Observaţi că unele valori sunt subliniate şi îngroşate (de obicei, se încercuiesc, dar în Word a fost mai simplu să le subliniez şi să le îngroş).
Aceste valori sunt importante, deoarece ele ne spun care teste detectează pe care defecte.
De exemplu, testul 000 nu detectează defectul a_0, deoarece dacă presupun *a* b-l-0 şi încerc combinaţia 000 la intrările *abc* ale circuitului, ieşirea *d* va avea valoarea 1. Din păcate, va avea valoarea 1 şi când circuitul este bun, adică fără defectul presupus *a* b-l-0. Nu ştim dacă ieşirea este 1 din cauza defectului a_0, sau din cauză că am injectat **a=0**, b=0, c=0, şi trebuie să dea 1, circuitul fiind bun. Deoarece pentru acest test 000, ieşirea are aceeaşi ieşire pentru circuitul bun şi circuitul cu defectul a_0, testul 000 nu detectează defectul a_0.
Pe de altă parte, testul 111 îl detectează pe a_0, pentru că, la acest test, circuitul bun ar da 0 iar circuitul cu defect (a_0) ar da 1 la ieşire, adică se poate observa diferenţa dintre circuitul bun şi circuitul defect.
Nu uitaţi că nu trebuie să detectăm care defect apare în circuit, ci doar să presupunem defectul, apoi să verificăm dacă este acolo sau nu.
Aşadar, încercuim (subliniem) valoarea ieşirii acolo unde ea diferă de circuitul fără defect (coloana a doua).

Se poate redesena tabelul doar cu aceste poziţii importante, dar nu este obligatoriu:

abc	a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1
000							x	
001							x	
010							x	
011		x					x	
100							x	
101				x			x	
110						x	x	
111	x		x		x			x

Observaţi că am omis coloana *d*, nu mai este necesară.

Astfel, am obţinut un set de teste T={000, 001, 010, 011, 100, 101, 110, 111}. De exemplu, dacă presupunem *d* blocat la zero, acesta poate fi detectat cu 000 (sau 001, 010, 011, 100, 101, 110, de altfel). d_1 este detectabil cu 111.

Setul de teste minimal

Problema cu setul de teste de mai sus este că el este exhaustiv, adică conţine toate combinaţiile posibile ale intrărilor. La trei intrări, nu este o problemă, sunt doar 2^3=8 teste. Dar la un circuit cu 1000 de intrări? $2^{1000} \approx 1.07 \times 10^{301}$, adică mult. Nu este rentabil să se încerce toate combinaţiile posibile, pentru că ar dura prea mult testarea unui singur integrat.

Este posibilă reducerea numărului de teste, chiar la un minim, care se numeşte „setul de teste minimal”. Acest lucru este posibil datorită modelului de defect, care este „singular”. De exemplu, în tabelul de mai sus, testul 011 îl detectează pe a_1, dar şi pe d_0. Nu uitaţi, se presupune că cele două defecte nu apar simultan şi, mai mult, nu trebuie să diferenţiem între ele – ştim dinainte pentru care defect aplicăm testul.

Pentru determinarea setului minim de teste, se foloseşte metoda de mai jos.

Testele esenţiale

Tabelul anterior, din nou:

	abc	a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1
	000							x	
	001							x	
	010							x	
es.	011		x					x	
	100							x	
es.	101				x			x	
es.	110						x	x	
es.	111	x		x		x			x

Scopul este scurtarea, chiar minimizarea numărului de teste, dar doar până la limita ca pentru fiecare defect de tip blocat-la singular posibil din acest circuit, să avem cel puţin un test capabil să-l detecteze.

Pentru defectul a_0, testul 111 este foarte important, deoarece este singurul care îl detectează. Dacă nu-l includem pe testul 111 în setul minim, setul minim va fi incomplet, nu va conţine nici un test care să-l detecteze pe *a* b-l-0. De aceea, testul 111 este **test esenţial** şi va trebui inclus în setul minim de teste. Similar, 011 este esenţial din cauza lui a_1, 111 din cauza lui b_0, 101 din cauza lui b_1, 111 din cauza lui c_0, 110 din cauza lui c_1, 111 din cauza lui d_1. Observaţi că d_0 nu are teste esenţiale, deoarece are mai multe teste capabile să-l detecteze.

Colapsul defectelor

Tabelul anterior, din nou:

abc	a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1

	000							x	
	001							x	
	010							x	
es.	011		x					x	
	100							x	
es.	101				x			x	
es.	110						x	x	
es.	111	x		x		x			x

ech.

dom.

Echivalenţe

Observaţi că **mulţimea de teste** pentru defectele a_0, b_0, c_0 şi d_1 **coincide**. Desigur, în circuitul nostru, această mulţime constă într-un singur test, anume 111. De aceea, defectele a_0, b_0, c_0 şi d_1 se consideră echivalente şi ele formează o **clasă de echivalenţă**.
Pe de altă parte, de exemplu defectul c_1 nu este echivalent cu d_0, deoarece d_0 are teste care nu-l detectează pe c_1, aşadar mulţimile lor de teste nu coincid. Acest circuit are o singură clasă de echivalenţă a defectelor de tip blocat-la singulare.
Pentru setul minim de teste, trebuie ales câte un defect reprezentativ din fiecare clasă de echivalenţă şi găsit un test pentru acel defect. Celelalte defecte din acea clasă de echivalenţă se pot ignora la căutarea de teste, deoarece testele care se găsesc pentru defectul reprezentativ vor detecta şi defectele echivalente cu acesta.

Dominanţe

Observaţi că mulţimea de teste a defectului d_0 include mulţimea de teste a lui a_1, mulţimea de teste a lui b_1 şi mulţimea de teste a lui c_1. De aceea, defectul d_0 se consideră **dominant** pentru a_1, b_1 şi c_1, sau „defectul d_0 domină defectele a_1, b_1 şi c_1" sau „defectele a_1, b_1 şi c_1 sunt dominate de defectul d_0".
Pe de altă parte, de exemplu defectul d_1 nu-l domină pe c_0, deoarece mulţimea de teste c_0 nu este submulţime a mulţimii de teste d_1. Oricum, aceste două defecte sunt deja echivalente.
Pentru setul minim de teste, este suficient să se găsească teste pentru defectele dominate din fiecare clasă de dominanţă şi se poate ignora defectul dominant, deoarece testele care detectează defectele dominate îl vor detecta şi pe cel dominant.

Setul minim de teste

Setul minim de teste trebuie să includă testele esenţiale, teste pentru defectele reprezentative din fiecare clasă de echivalenţă şi teste pentru defectele dominate.

Tabelul de mai sus, din nou:

	abc	a_0	a_1	b_0	b_1	c_0	c_1	d_0	d_1
	000							x	
	001							x	
	010							x	
es.	011		x					x	
	100							x	
es.	101				x			x	
es.	110						x	x	
es.	111	x		x		x			x

ech.

dom.

Setul minim de teste: T_{min}={011, 101, 110, 111 – am scris testele esenţiale, dar este încă incomplet. Acum trebuie să includem cel puţin un test pentru fiecare defect posibil. Din fericire, nu trebuie să căutăm teste pentru b_0, c_0 sau d_1, deoarece sunt echivalente cu a_0 , ales reprezentativ pentru singura clasă de echivalenţă din acest circuit. De asemenea, putem să-l omitem pe d_0, pentru că este dominant, şi este suficient să

găsim teste pentru defectele dominate de el.
Test pentru a_0: 111 (deja inclus, este test esenţial); test pentru a_1: 011 (deja inclus, esenţial); b_1: 101, deja inclus; c_1: 110, deja inclus.
Setul minim de teste: T_{min}={011, 101, 110, 111}.