



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

Testarea Sistemelor

23. Concepte avansate BIST

Concepte avansate BIST

Această secțiune descrie câteva dintre conceptele BIST avansate. În mare aceste concepte se referă la anumite caracteristici ale arhitecturilor BIST:

1. Planificarea sesiunilor de teste,
2. Controlul sesiunilor multiple de teste atunci când se utilizează o arhitectură BIST distribuită și dedicată (*embedded*), și
3. Proiectarea unor arhitecturi care implementează parțial structuri *BIST*.

Pentru simplitatea abordării se va presupune că se utilizează arhitectura BILBO. Un registru R_i se spune că este *driverul* unui bloc logic C dacă o parte dintre liniile de ieșire ale registrului R_i sunt linii de intrare pentru C .

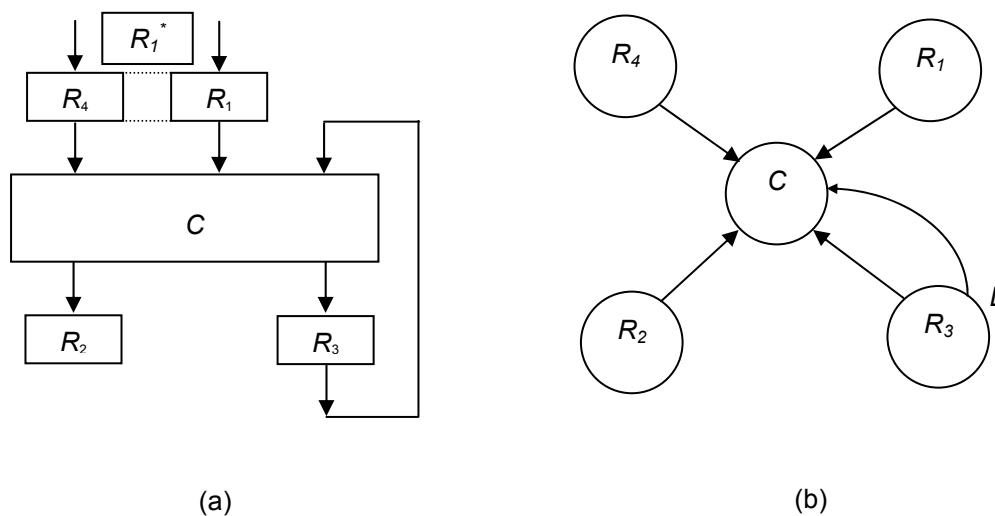


Figura 1.

- (a) Componente ale circuitului.
 (b) Graful adiacenței registrelor.

Un registru R_j se spune că este *receptor* al blocului C dacă unele dintre liniile de ieșire ale blocului C sunt linii de intrare pentru registrul R_j .

Registrul R_i se spune că este *adiacent* registrului R_j dacă există un bloc logic C astfel încât R_i un driver al blocului logic C iar R_j este un receptor al blocului C .

Dacă registrul R , este atât un receptor cât și un driver al blocului C , atunci registrul R este *auto-adiacent*.

Exemplul 1: Se consideră structura de circuit din figura 1. Registrele R_1 , R_3 și R_4 sunt drivere ale blocului C , R_2 iar R_3 și R_2 sunt receptori ai blocului C . Registrul R_3 este un registru auto-adiacent. Registrele R_1 și R_2 sunt adiacente dar, registrele R_1 și R_4 nu sunt adiacente.

Pentru testarea blocului C registrele R_1 și R_4 trebuie poziționate în modul PRPG iar registrul R_3 trebuie poziționat atât în modul PRPG cât și în modul MISR. Pentru reducerea volumului de circuite, registrele auto-adiacente vor fi operate în modul MISR.

De remarcat faptul că celulele de memorie și registrele funcționale pot fi grupate în noi registre în scopul formării unui registru BILBO.

Poate fi perfect fezabil, spre exemplu, să se grupeze registrele R_1 și R_4 într-un singur LFSR R_1^* pe durata modului de test. Pe de altă parte, însă, n-ar fi avantajos să se grupeze R_2 și R_4 laolaltă, deoarece aceste registre operează în moduri diferite pe durata modului de test.

◇

Planificarea testelor

O sesiune de teste este definită ca o atribuire a modurilor de test pentru registrele BILBO în vederea testării unuia ori mai multor blocuri logice. Un bloc logic se consideră testat dacă registrele care-l conduc sunt în modul PRPG iar registrele receptoare și auto-adiacente sunt în modul MISR.

Un circuit, în general, poate fi modelat printr-un graf bipartit $G = (N_A, N_B, E)$ referit prin graful de adiacență (GAR), unde:

- N_A este setul de noduri de tipul A fiecare nod din acest set reprezentând un registru,
- N_B este setul de noduri de tipul B fiecare nod din acest set reprezentând un bloc logic combinațional, iar
- E este mulțimea de arce dintre nodurile de tip A și B .

Un arc direct există între un nod de tipul A (corespunzător registrului R_i) și un nod de tipul B (corespunzător blocului logic C_k) dacă registrul R_i este un driver al blocului logic C_k .

Iar dacă între un nod de tipul B (corespunzător blocului logic C_k) și un nod de tipul A (corespunzător registrului R_j) există un arc atunci registrul R_j este un receptor al blocului logic C_k .

Suplimentar, un nod de tipul A , corespunzător unui registru auto-adiacent este etichetat prin marcajul L .

Figura 1.(b) prezintă graful de adiacență asociat figurii 1.(a).

O problemă de planificare a testelor se definește după cum urmează:

- determină numărul minim de sesiuni de test pentru testarea tuturor blocurilor logice combinaționale. Această problemă este legată de determinarea *numărului cromatic* al grafului G^* . Numărul cromatic al unui graf este numărul minim de culori ce pot fi atribuite nodurilor grafului G^* , astfel încât nici un arc să nu conecteze două noduri având aceeași culoare.

Exemplul 2: În figura 2 se înfățișează un circuit făcut testabil prin metodologia BILBO iar în figura 3 se prezintă graful GAR corespunzător. Circuitul poate fi complet testat prin doar trei sesiuni de test, așa cum se arată în figura 3.

Circuitul poate fi testat prin doar trei sesiuni, așa cum se arată în figura 4.

Figura 4.a prezintă o sesiune de test în care C_1 și C_5 sunt testate simultan.

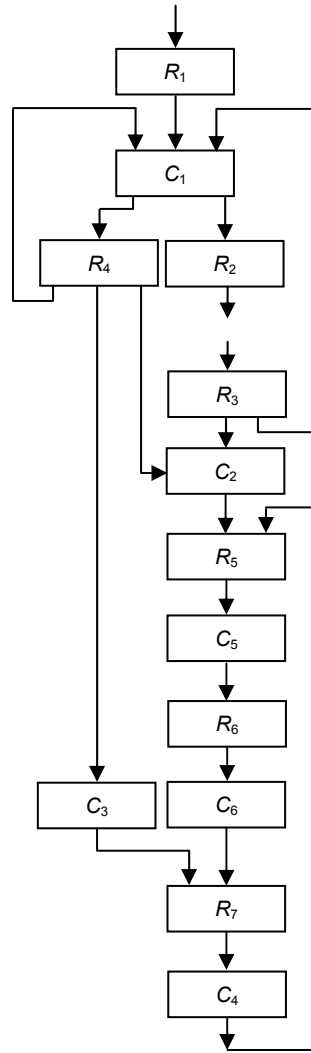


Figura 2. Circuitul testat din exemplul 2.

Problema planificării testelor devine mult mai complexă atunci când se ia în considerație și timpul de testare al fiecărui bloc.

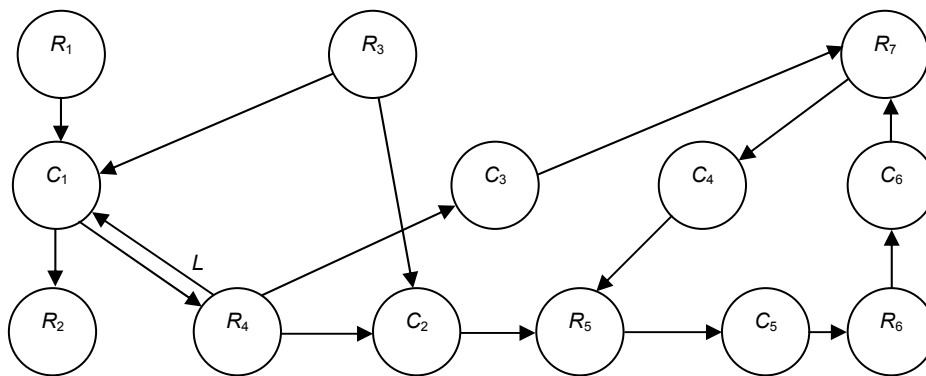
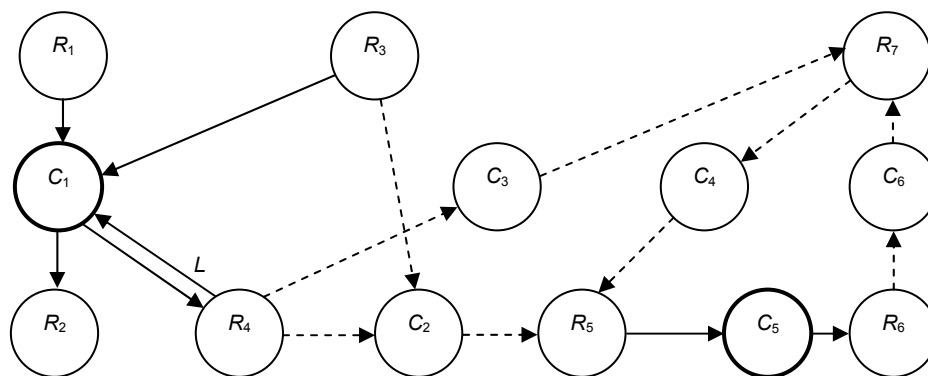


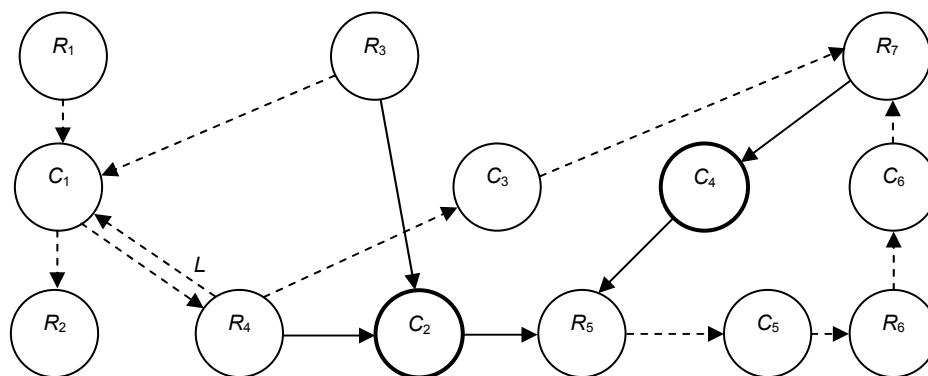
Figura 3. Graful de adiacență al circuitului din figura 2.

În această situație obiectivul care trebuie minimizat este timpul testării întregului circuit integrat.

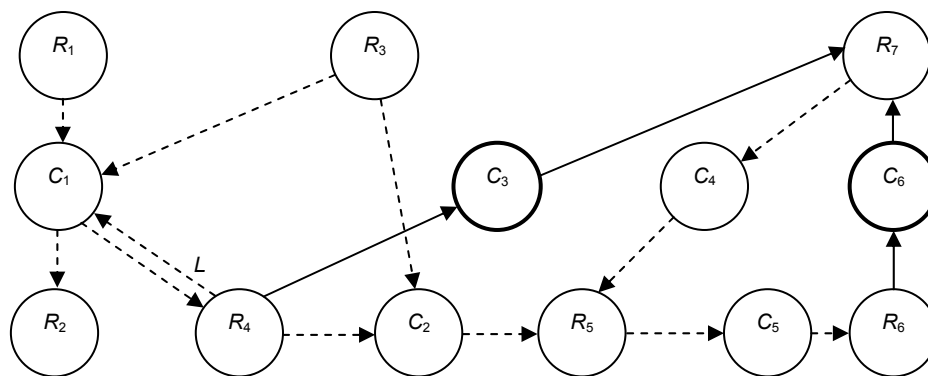
O sesiune de teste poate fi astfel mult mai complexă, câtă vreme se testează circuitul combinațional C_i , spre exemplu, se testează deasemenea circuitul C_j , iar după ce s-a testat C_j , se testează C_k și așa mai departe.



(a)




(b)



(c)

Figura 3. Cele trei sesiuni de test ale circuitului din figura 2.

Legenda:  - Bloc logic combinațional testat.

----- - Interconexiune neutilizată.

Controlul registrelor BILBO

Atunci când se desfășoară mai multe sesiuni de test, controlul registrelor BILBO apare ca fiind o chestiune importantă.

Fiecare registru BILBO necesită trei biți pentru controlul informației care să specifice în care din cele cinci moduri de operare posibile se află registrul: *normal*, *scan*, *inițializare*, *PRPG* sau *MISR*.

Ar fi de dorit ca acești biți de control să fie comuni tuturor registrelor și ar putea fi transmiși prin utilizarea a trei variabile de control comune.

Totuși, deoarece în fiecare sesiune de test rolul unui registru poate fi MISR ori PRPG este posibil să se transmită doar două semnale de stare.

Aceste două semnale pot fi desemnate să specifice unul din cele patru moduri, anume: *normal*, *scan*, *inițializare* și *test*.

Fiecare registru va avea un control unic care va specifica dacă acel registru operează ca PRPG sau MISR, atunci când registrul se află în modul *test*.

Distribuția acestor semnale poate necesita un surplus important de arie pe suprafața unui circuit integrat.

O abordare alternativă, este distribuirea controlului.

Aceasta s-ar putea implementa, spre exemplu, asociind fiecărui registru R_i , o celulă de memorie T_i .

Dacă $T_i = 1$ atunci registrul R_i operează în modul MISR, atunci când este valid modul *test*; altfel operează în modul PRPG.

Celulele de memorie pot fi organizate pe o cale separată de scan sau pot să facă parte din aceeași cale de scan care cuprinde și registrele BILBO.

Înainte să se execute o sesiune de test aceste celule pot fi inițializate astfel încât fiecare registru BILBO să opereze în modul corespunzător.