

Capitolul 9: JTAG- principii de testare

9.1.Principii generale și unelte de testare

Orice sistem integrat pe bază de calculator trebuie testat, fie că este vorba despre un model experimental sau de producția de serie. Testabilitatea trebuie asigurată încă din faza de proiectare, conform următoarelor principii:

1. Principiul vizibilității care presupune accesibilitatea la nodurile care vor fi testate;
2. Principiul controlabilității care presupune posibilitatea controlului principalelor semnale de pe placă, adică inhibarea lor și substituirea cu semnale de test;
3. Principiul sincronizării generale solicită sincronizarea tuturor evenimentelor de pe o placă cu un singur semnal de tact care să fie posibil de generat și de echipamentul de testare.
4. Principiul partiționării presupune gruparea topologică pe placă a circuitelor în raport cu funcțiile realizate;
5. Principiul inițializării generale impune crearea posibilității de a aduce circuitele de pe placă într-o stare cunoscută în momentul declanșării testului;
6. Principiul autotestării recomandă transferarea unor operații de testare sau auxiliare testării la nivelul plăcii.

Integrarea permite includerea funcțiilor de autotestare la nivelul *chip*-urilor (Built In Self Test BIST), și funcții de testare a interconexiunilor de pe plăci (Boundary Scan).

Testele pot fi clasificate în:

1. Teste parametrice- se evaluează parametrii statici (ex. tensiune de offset etc.) sau dinamici (timpi de creștere etc.);
2. Teste funcționale- se evaluează funcționalitatea. Condiții le pot fi statice (cu frecvența mai mică decât în circuit) sau dinamice (frecvențe de test comparabile cu cele din circuit). Testele pot fi executate în condiții parametrice impuse.

Efectuarea unui test se realizează conform schemei bloc din figura 9.1.

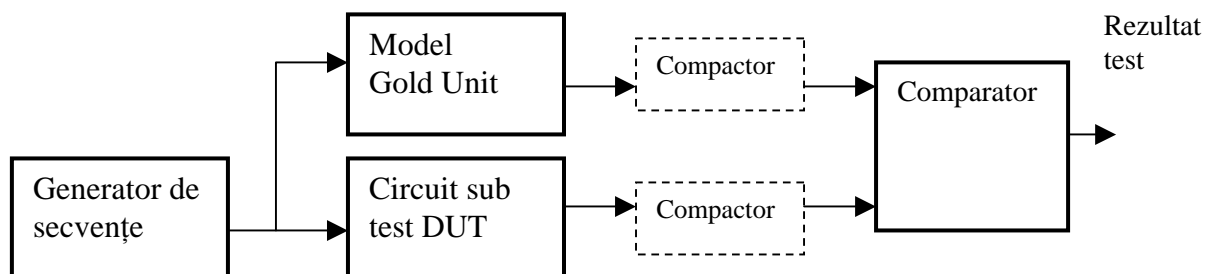


Figura 9.1: Testarea prin comparație cu un model

O secvență de test este aplicată circuitului de testat și unui model (o placă martor sau un model matematic). Dacă secvența de ieșire este identică cu cea de referință atunci circuitul testat se poate considera fără defecte. Prin compactare se micșorează lungimea secvenței de ieșire și comparația durează mai puțin. Avantajul este umbrit de posibilitatea apariției unor defecte nedetectabile.

Pentru testarea funcțională a unui modul se aplică secvențe de test la intrare și se citesc răspunsurile atât la ieșirile plăcii cât și în nodurile interioare pentru care s-a implementat principiul vizibilității și poate fi:

1. Testarea funcțională statică își propune detectarea defectelor statice PP0, PP1 și scurtcircuite;
2. Testarea dinamică, la frecvența de lucru, detectează de exemplu impulsuri parazite;
3. Testarea funcțională în sistem, cu 2 subvariante:
 - Placa funcționează în testor și se simulează funcționarea în sistem;
 - Placa funcționează în sistem, testorul culege informații (ex. analizor logic)

La testarea în circuit acțiunea testului este fixată asupra unui singur *chip*. Secvența se aplică direct pe pinii *chip*-ului testat și se citesc răspunsurile *chip*-ului. Accesul se face prin pat de cuie- *bed of nails*. Testorul trebuie să rezolve problemele:

1. Izolarea circuitului prin dezactivarea ieșirii tri state, inhibarea generatoarelor de semnal, întreruperea buclelor de reacție;
2. Protecția ieșirilor cuplate în nodurile forțate prin limitarea valorilor curenților injectați.

Defectul este o imperfecțiune fizică sau logică care apare în cadrul unei componente hardware sau software iar **Eroarea** este o manifestare a defectului și reprezintă o deviație de la corectitudinea de execuție a funcțiilor.

Cauzele defectelor pot fi:

1. Specificații de proiectare greșite
2. Implementări greșite
3. Componente defecte
4. Perturbații exterioare

Caracteristicile defectelor sunt:

1. Cauzele defectelor
2. Natura- hardware sau software
3. Durata defectelor- permanente, tranzitorii sau intermitente
4. Extinderea defectelor- generale sau locale
5. Valoarea defectului (la cele parametrice).

Defectele logice sunt:

1. Poneri pe 0 (PP0) și poneri pe 1 (PP1)
2. Scurtcircuite, care pot fi între un traseu de semnal și una din bornele de alimentare, figura 9.2 sau între două trasee de semnal, figura 9.3. Aceste defecte se pot detecta indirect prin măsurarea curentului absorbit și compararea lui cu un curent al unei plăci bune sau prin măsurarea temperaturii chip-urilor de pe placă.

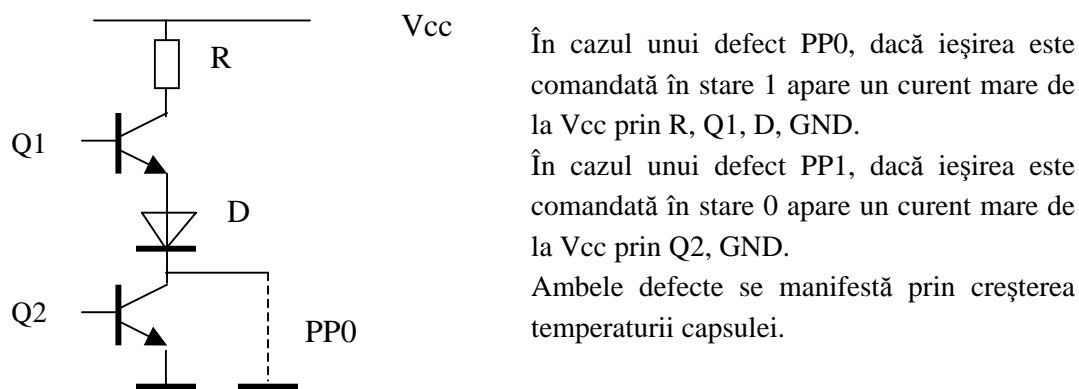


Figura 9.2: Scurtcircuit între un traseu de semnal și una din bornele de alimentare

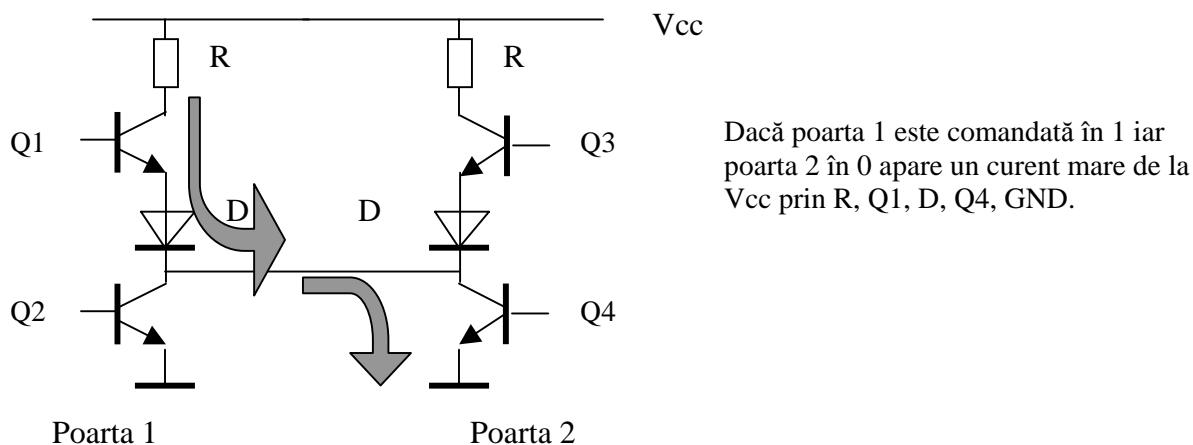


Figura 9.3: Scurtcircuit între două trasee de semnal

Apariția sistemelor structurate pe magistrale a dus la necesitatea unor noi moduri de testare. În 1973 a fost introdus de către HP analizorul logic. Analizorul logic este dedicat activității de laborator în etapele de proiectare și punere la punct a sistemului. Cerințele necesare pentru un analizor logic sunt:

1. Să urmărească și să vizualizeze mai multe semnale de intrare în același timp;
2. Să dispună de frecvențe de eșantionare mai mari decât cel mai rapid semnal din sistem;
3. Să dispună de un nivel de intrare de prag adaptabil la diferite tipuri de familii logice;
4. Să memoreze secvențe de date;
5. Să dispună de posibilități de declanșare care să permită captarea unor blocuri de date dintr-un flux de date;
6. Să asigure moduri variate de vizualizare.

Analiza logică poate fi:

1. Sincronă (cu tactul sistemului), dedicată părții software. Se compară datele cu cele de referință și se stabilesc diferențele.
2. Asincronă (cu tactul mai mare decât tactul sistemului), dedicată părții hardware, care pune în evidență defecte hardware- impulsuri eronate etc.

Schema bloc a unui analizor logic este dată în figura 9.4.

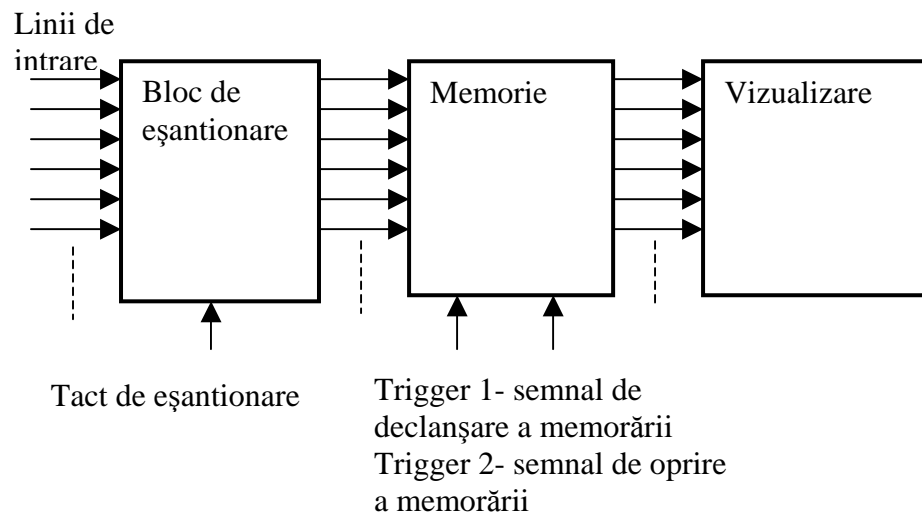


Figura 9.4: Schema bloc a unui analizor logic

De exemplu la o magistrală externă de MC *trigger*-ul de declanșare a memorării poate fi semnalul ALE care identifică începutul unui ciclu de acces la memorie sau I/O. Se poate genera un semnal special ca *trigger* prin software prin intermediul unui port de I/O. Există și posibilitatea de declanșare cu un cuvânt stocat în analizor. La întâlnirea aceluia cuvânt pe una dintre liniile de intrare se pornește memorarea. Se poate stabili o întârziere a declanșării după apariția cuvântului.

Cel mai comun mod de afișare este sub forma unei diagrame de timp, figura 9.5.

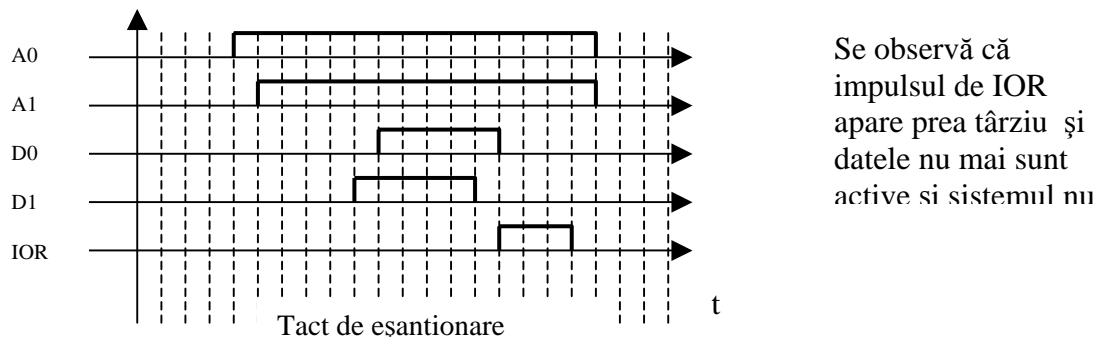


Figura 9.5: Afișarea datelor la un analizor logic

Analizorul de semnături se folosește în activitatea de depanare. Într-un punct se culege o “semnătură”, adică o formă comprimată a fluxului de date. Semnătura se compară cu cea determinată când circuitul funcționează bine (sau este determinată prin calcul) și se decide dacă nodul respectiv funcționează bine sau nu. Dacă nu, se culege semnătura dintr-un nod anterior ș.a.m.d. Analiza de semnături se poate aplica la fenomene sincrone, inclusiv pe bază de magistrale dar nu se poate aplica la fenomene asincrone, cum ar de exemplu DMA pe magistrală. Schema bloc este dată în figura 9.6.

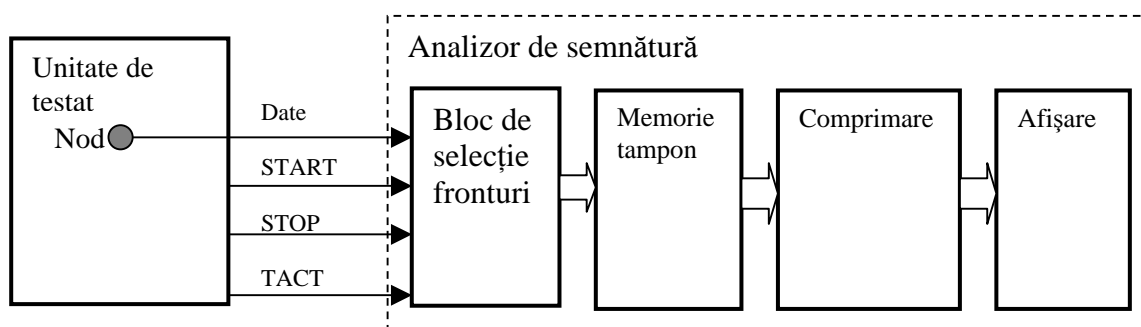


Figura 9.6: Schema bloc a unui analizor logic

Diagrama de semnal este dată în figura 9.7.

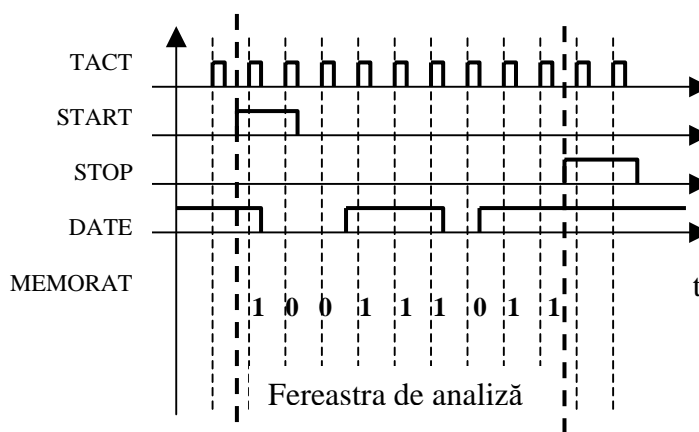


Figura 9.7: Diagrama de semnal preluat de analizorul de semnături

Comprimarea se realizează prin coduri ciclice. Schema de implementare a unei comprimări bazate pe cod ciclic se realizează prin registre cu reacție. De exemplu dacă ecuația reacției este $G=x^{16}+x^{12}+x^7$ atunci schema de implementare este cea din figura 9.8.

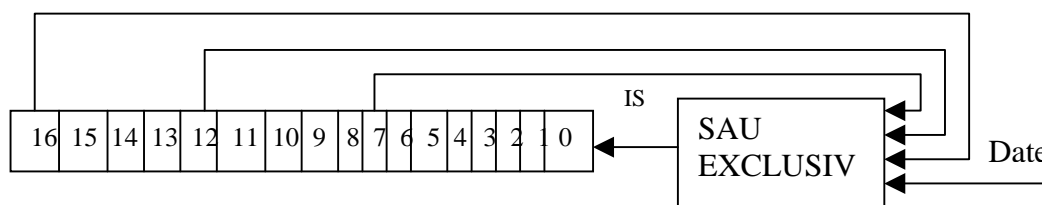


Figura 9.8: Implementarea comprimării

Modurile de lucru ale analizorului de semnături pot fi:

1. NORMAL, când datele, semnalele de START și STOP apar repetitiv în nodurile testate. Testarea continuă dacă posibilitatea punerii în evidență a defectelor intermitente (modificarea semnăturii la un ciclu din N cicluri).
2. REȚINERE, se afișează semnătura culeasă în prima fereastră după RESET
3. AUTOTESTARE, se introduc sondele la un generator de secvențe și se compară semnătura obținută cu cea de control și se poate decide dacă analizorul funcționează bine.

Metoda analizei de semnături poate fi aplicată și în domeniul analogic [1], semnătura unei componente analogice fiind caracteristica ei tensiune- curent. Aparatul folosit pentru analiza de semnături conține o sursă de tensiune V_s și o impedanță internă Z_s și este aplicat pentru ridicarea caracteristicii componentei Z_L . Dacă aceasta este o rezistență se obține pe un ecran caracteristica cunoscută, panta drepte fiind măsura valorii rezistenței, care poate fi comparată cu valoarea corectă. Reprezentarea se face în 4 cadrane. Semnătura unui capacitor este o elipsă, reprezentare obținută la aplicarea unei tensiuni alternative. O elipsa asimetrică așezată reprezintă semnătura unui condensator cu pierderi (cu componentă rezistivă). La fel se pot analiza semnăturile altor componente, cum este de exemplu joncțiunea pn. Semnătura joncțiunii pn poate fi folosită pentru analiza circuitelor integrate, prin analiza fiecărei intrări/ ieșiri, pornind de la premiza că majoritatea defectelor apar în zona I/O.

Observație:

Autotestarea este de o importanță vitală pentru orice echipament electronic, oricât de simplu. De exemplu aprinderea tuturor indicatoarelor luminoase din bordul auto la introducerea cheii în contact constituie un test al semnalizărilor de avarie. Osciloscopul are un generator dreptunghiular de 1kHz pentru test. Orice realizare profesională trebuie să aibă posibilități de autotest hardware sau software.

Pentru modulele realizate ca model experimental este indicată folosirea unor LED-uri de semnalizare. Un LED la tensiunea de alimentare poate asigura utilizatorul că modulul este alimentat. Un set de LED-uri care se aprind pe rând după efectuarea unor secvențe importante soft poate indica starea în care s-a blocat modulul. Unele plăci de bază de PC au fost prevăzute cu LED-uri pentru a marca evoluția autotestului și a încărcării sistemului de operare.

9.2.Structura SCAN

Un circuit care conține atât componente combinaționale cât și componente secvențiale poate fi redesenat astfel, figura 9.9.

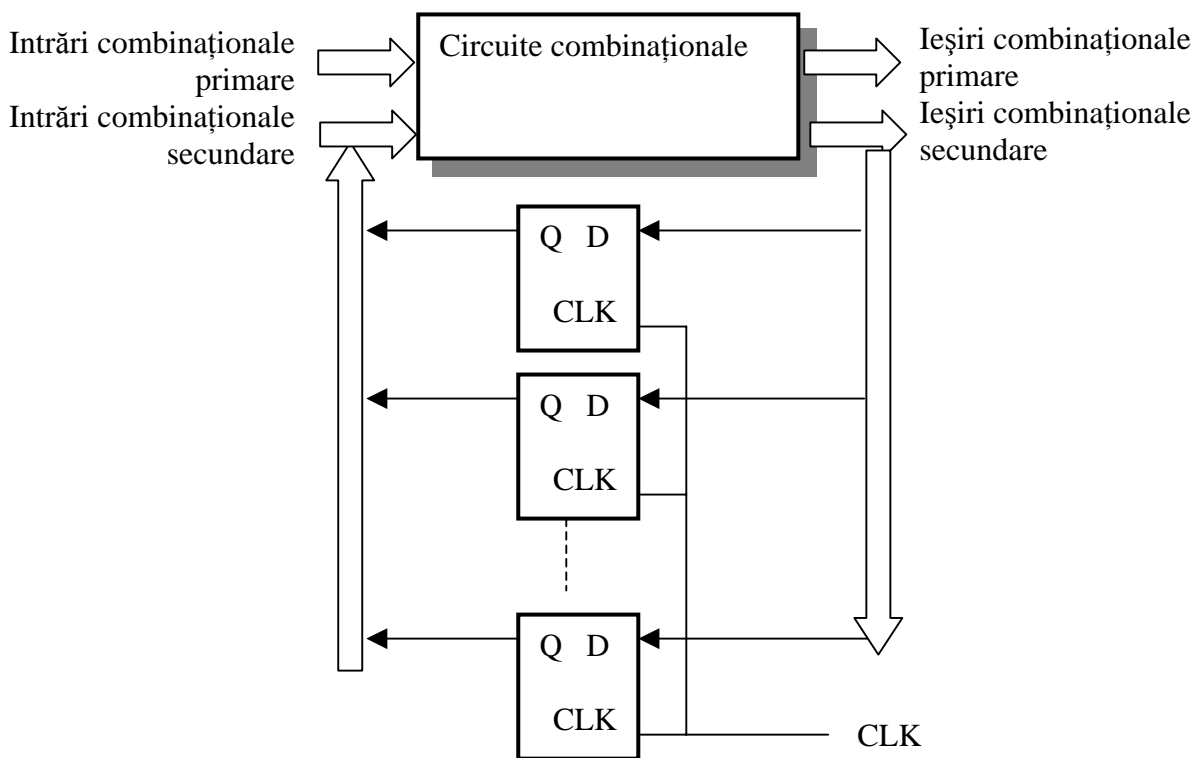


Figura 9.9: Partiționarea unei scheme

S-a împărțit (partiționat) această schemă pentru punerea în evidență a părții combinaționale și a celei secvențiale.

Proiectarea pentru testabilitate bazată pe registre de scanare se bazează pe adăugarea de elemente de circuit suplimentare unui *chip* (ceea ce nu mai este o problemă la gradul actual de integrare). Circuitele înzestrate cu aceste structuri de test au două moduri de funcționare:

- NORMAL când execută funcția pentru care este utilizat
- TEST se execută operații de testare. Se dorește ca semnalele de testare la pini să fie cât mai puține.

Pe baza acestei redesenări se poate introduce noțiunea de structură SCAN multiplexată – MDSD Multiplexed Data Scan Design în care se adaugă un MUX pentru fiecare bistabil existent, figura 9.10. Dacă intrarea de selecție S a multiplexoarelor validează calea I1 -D atunci circuitul funcționează normal (N) iar dacă validează calea I0 -D atunci se validează modul de test (T).

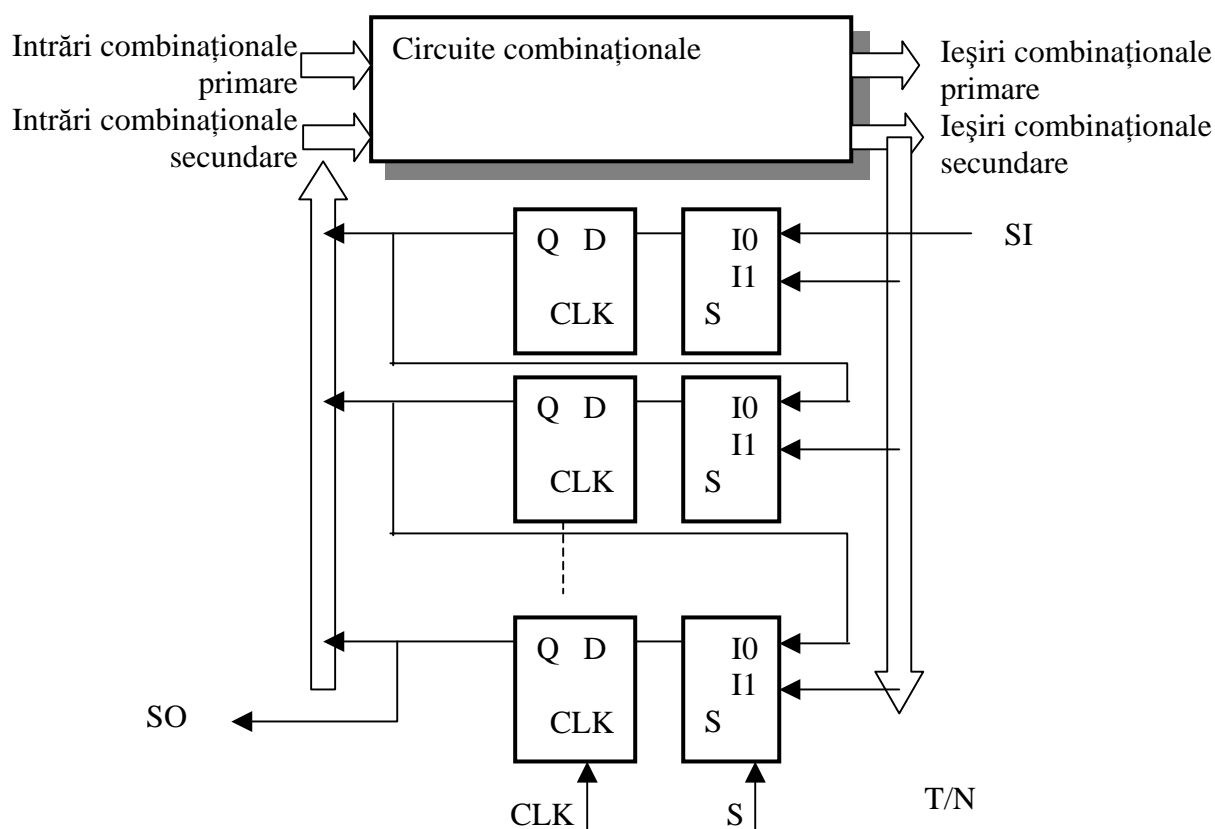


Figura 9.10: Structura SCAN multiplexată

Testarea părții secvențiale în modul test (T):

1. 0 la SI și un număr de impulsuri CLK egal cu numărul de bistabili din structură asigură inițializarea tuturor bistabililor cu 0 (în lipsa defectelor);
2. 1 la SI și apoi n-1 de 0 înseamnă trecerea fiecărui bistabil în 1, pe rând la fiecare impuls de CLK;
3. se pot aplica 2^n (n fiind numărul de bistabili) secvențe pentru testarea integrală cu toate combinațiile posibile.

Condiția de eroare se obține dacă la ieșirea SO șirul de impulsuri nu este cel așteptat.

Testarea părții combinaționale în modul test (T):

1. Prin încărcarea bistabililor (prin SI) cu o secvență de test se obține un vector pe intrările combinaționale secundare. Se aplică un vector de test pe intrările primare și astfel se poate asigura o secvență de test completă la intrările circuitului combinațional și se poate citi **răspunsul la ieșirile combinaționale primare**;
2. Se pune T/N pentru funcționare normală și în bistabili se încarcă răspunsul părții combinaționale (ieșiri secundare);
3. Se pune T/N pentru test și se citește serial la SO, după n impulsuri de tact, **răspunsul** citit la **ieșirile secundare**. Împreună cu ieșirile primare se obține răspunsul părții combinaționale.

În structura BOUNDARY SCAN fiecărui pin de I/O al unui *chip* i s-a atașat o celulă specializată care separă logica internă de exteriorul *chip*-ului, figura 9.11.

1. T/N pe poziția normal, DO identic cu DI și circuitul realizează funcția pentru care a fost proiectat;
2. T/N pe poziția test, bistabilii pot fi încărcăți serial prin intrarea SI și în $n/2$ impulsuri de CLK se poate încărcă un vector de test la pinii de intrare, apoi se pot citi răspunsurile circuitului citind ieșirile. Tot pe poziția test, bistabilii pot fi încărcăți paralel cu vectorul de test prezent la intrare, cu semnalul SHIFT/ LOAD (S/L) pe poziția LOAD (L). Ieșirile pot fi testate deplasând serial vectorul de intrare la ieșiri.

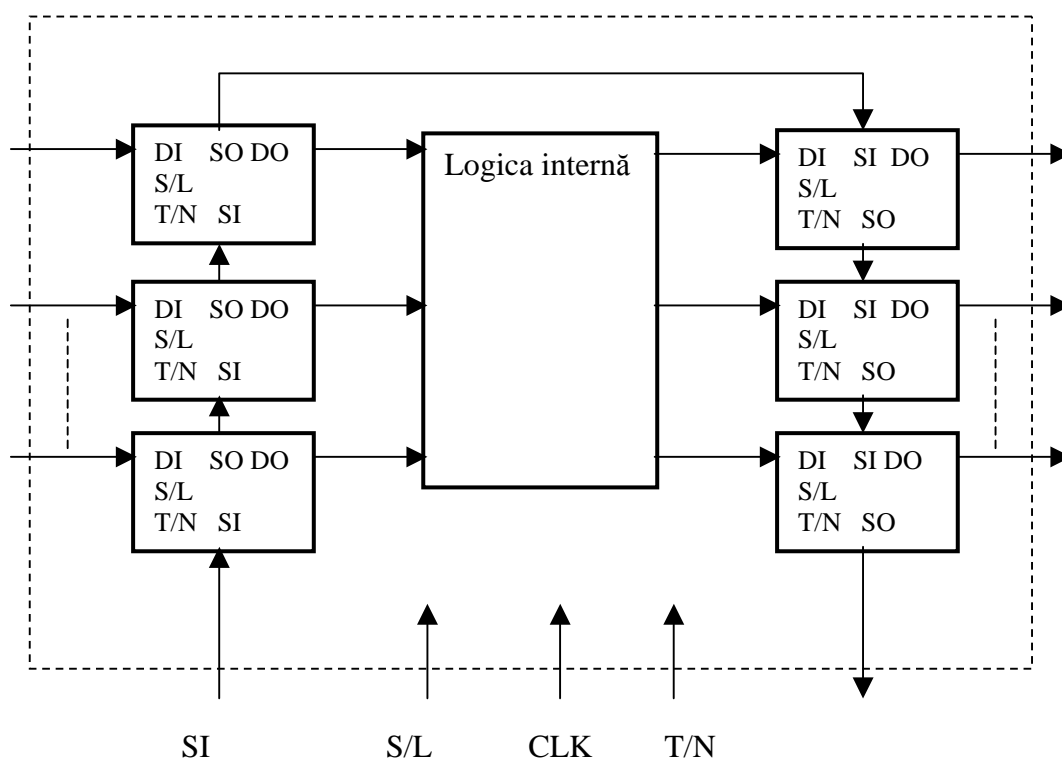


Figura 9.11: Structura BOUNDARY SCAN

Testarea interconexiunilor se poate realiza cu o buclă Boundary Scan pe o placă cu *chip*-uri care admit acest mod de test, figura 9.12:

1. Se încarcă paralel în primul circuit (A) un vector de test 010101... ;
2. Se deplasează serial vectorul de test de la intrări la ieșiri;
3. Se încarcă în circuitul B setul de ieșiri din circuitul A ș.a.m.d. până datele apar la ieșirea SO a ultimului circuit;
4. Se analizează secvența de ieșire, de exemplu 2 de 0 alăturați arată existența unui defect de punere la 0 PP0.

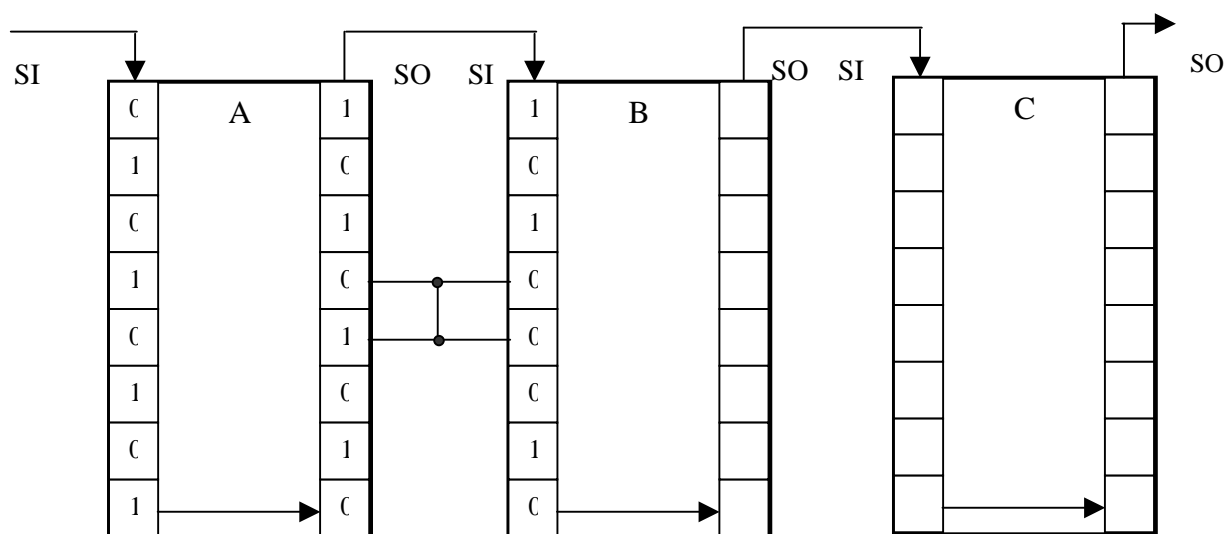


Figura 9.12: Testarea interconexiunilor

Testarea se face pe etape:

1. Testarea conexiunilor
2. Testarea chip-urilor.

Motivul testării în 2 etape este acela că o conexiune PP0, PP1 sau scurtcircuit între 2 trasee poate avea aceleași manifestări ca un defect intern. Pentru testare se aleg mai multe tipuri de șiruri de date, capabile să pună în evidență defectele logice PP0, PP1 sau scurtcircuitate. De exemplu se poate aplica un șir de 0 și se detectează punerile pe 1, apoi un șir de 1 și se detectează punerile pe 0, apoi alternativ 0 și 1 care detectează scurtcircuitate.

9.3.JTAG. Standardele IEEE 1149.1, IEEE 1149.4, IEEE1149.6 și IEEE 1532

Grupul de lucru Joint Test Action Group (JTAG) a fost format în 1985 de către Harry Wardrop pentru a dezvolta o metodă de testare a plăcilor echipate după ce au fost produse. Grupul JTAG a realizat primul standard de testare în 1990 (Standardul IEEE 1149.1) în care se regăsește arhitectura SCAN și Boundary Scan. Tot în 1990 Intel a realizat primul procesor dotat cu JTAG, 80486. În 1994 a fost adăugat la standard un supliment care descrie limbajul BSDL (Boundary Scan Description Language). Curând JTAG a depășit utilizarea la testarea plăcilor și a început să fie folosit la accesul modulelor interne ale circuitelor integrate pentru urmărirea funcționării lor (debugging). La un circuit cu interfață JTAG funcționarea este vizibilă prin JTAG imediat după Reset. Un emulator sau interfață JTAG permite accesul la modulele interne care pot astfel să fie urmărite. Prin JTAG funcționarea unui procesor poate fi oprită și un program se poate rula pas cu pas sau procesorul rulează un program și se introduc puncte de întrerupere în program. Există structuri cu acest mod de testare bine pus la punct cum sunt microcontrollerele ARM. O altă noutate introdusă este posibilitatea programării memoriei flash a circuitului prin JTAG.

Circuitele cu JTAG cuprind în arhitectura lor un port de testare TAP (Test Access Port) care permite:

1. Testarea funcțiilor circuitului;
2. Testarea conexiunilor cu alte circuite;
3. Excluderea circuitului din procesul de testare la nivelul plachetei;
4. Operații de autotestare executate de circuit (opțional).

În schema bloc a unui modul JTAG (figura 9.13) se poate observa structura bazată pe regiștrii.

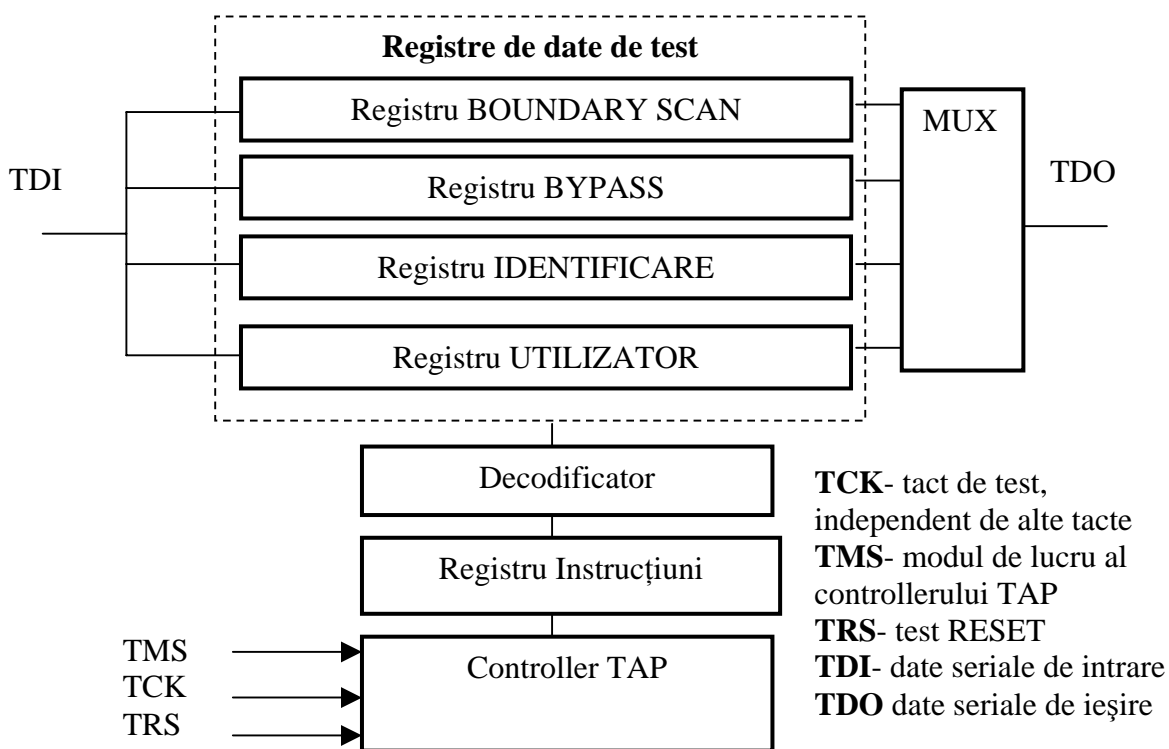


Figura 9.13: Modul JTAG

Regiștrii JTAG sunt:

1. Prin registrul BYPASS se exclude *chip*-ul din procesul de testare;
2. Registrul BOUNDARY SCAN este folosit pentru testarea circuitului și conexiunilor;
3. Registrul UTILIZATOR, liber ca utilizatorul să realizeze operații de test sau autotest;
4. Registrul IDENTIFICARE permite identificarea circuitului.
5. În registrul INSTRUCȚIUNI se trimit serial (prin TDI) instrucțiunile de executat care pot fi:
 - BYPASS, *chip*-ul este exclus din lanțul de testare. Logica *chip*-ului funcționează normal. Datele merg de la TDI spre TDO și poate fi testat următorul circuit din lanț;

- SAMPLE/ PRELOAD, logica *chip*-ului funcționează normal și ieșirile pot fi citite serial prin Boundary Scan. Se poate face astfel o operație de test funcțional în sistem;
- EXTEST, se testează conexiunile prin înscrierea registrului Boundary Scan cu valorile de test, apoi se citesc valorile din alt circuit.

Există și instrucțiuni opționale:

- HIGHZ- se comandă toate ieșirile în stare de înaltă impedanță;
- IDCODE- se citește codul de identificare *chip*.
- USERCODE – se selectează registrul UTILIZATOR (32 de biți).

Observație: la ultimele 2 instrucțiuni nu este afectată funcționarea normală simultană a *chip*-ului.

Standardul IEEE 1149.4 are scopul de a reduce dificultatea testării sistemelor care prelucrează semnale mixte digitale și analogice. Standardul de testare a circuitelor digitale IEEE 1149.1 se poate extinde pentru testarea circuitelor analogice. În schema de testare a fost introdus ABM Analogue Boundary Module și DBM Digital Boundary Module, figura 9.14.

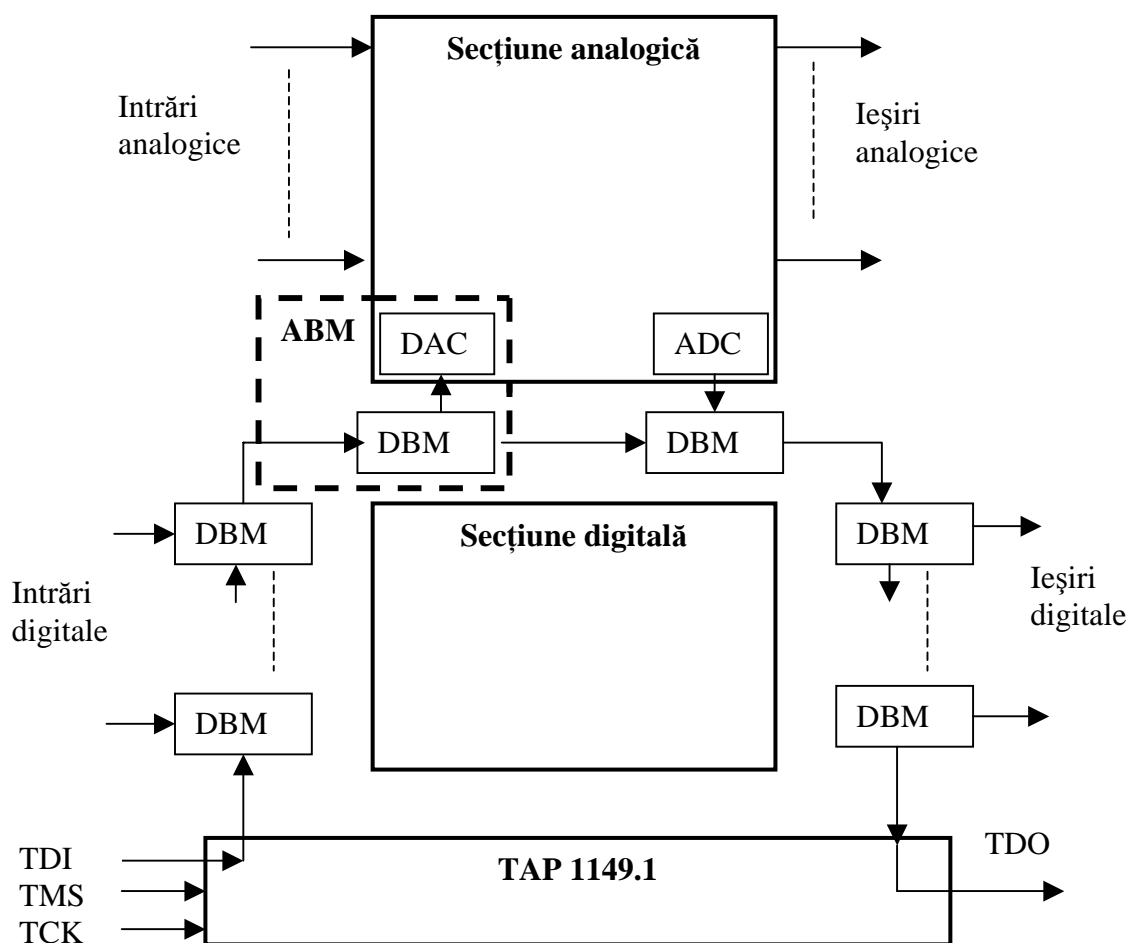


Figura 9.14: Structura de testare analogică și digitală

Modulele DBM au structura prezentată în figura 9.14. Pentru a se putea testa partea analogică care a fost separată se introduc modulele ABM care conțin convertoare AD și DA de un bit. Se pot astfel unifica vectorii de test și răspunsul circuitului în forma digitală. La prima vedere pare că se poate testa o valoare analogică prin comparare cu un singur prag (în convertorul AD de un bit), ceea ce este un nivel sărac de informație. De fapt pragul poate fi programat și se pot face comparații succesive cu mai multe nivele, ceea ce mărește precizia testării dar prelungește și timpul de testare. Transformarea din digital în analogic și invers are loc în celula ABM, figura 9.15.

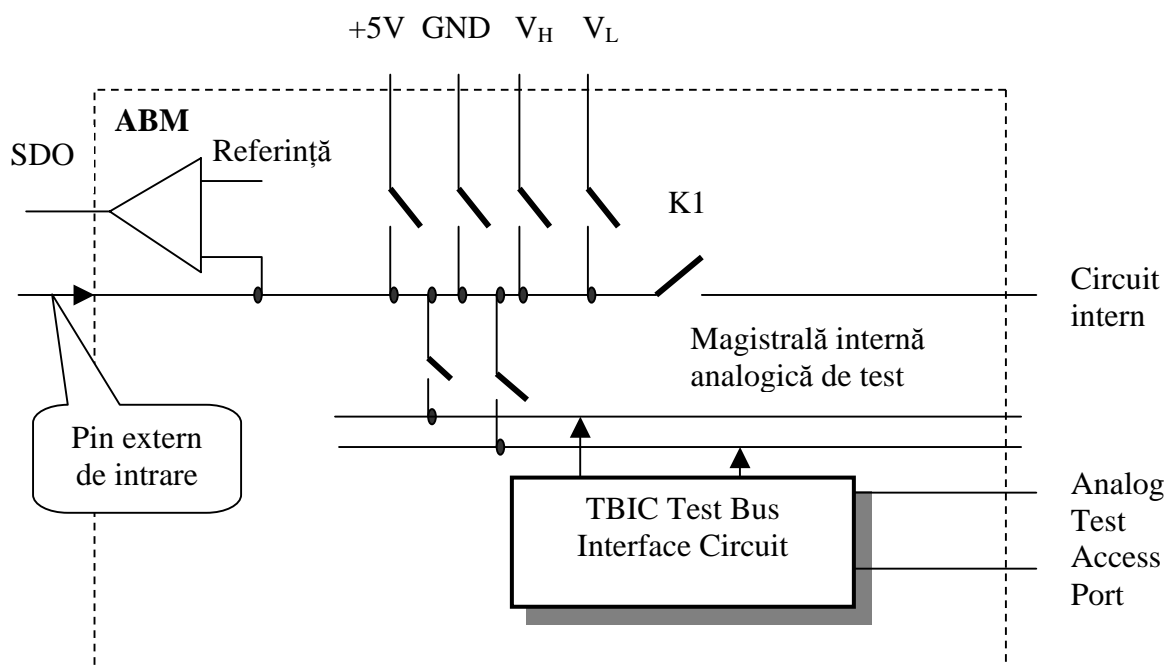


Figura 9.15: Structura ABM Analogue Boundary Module

Procedura de test:

- Se citește semnalul IN, se compară cu valoarea de referință și rezultatul (1 bit) se trimite pe SDO (CAD de un bit). Astfel s-a măsurat intrarea (K1 deschis);
- Se închide K1 și se pot genera spre circuitul intern +5V, GND, V_H , V_L (pentru un circuit digital) dar și valori analogice de la TBIC;
- Pentru o celulă ABM de ieșire se schimbă pe schema bloc poziția pin extern cu circuit intern.
- Pentru citiri mai exacte a tensiunii de intrare, la referință se pot conecta valori analogice prin TBIC.

Dezvoltarea în timp a JTAG a dus la necesitatea introducerii a noi standarde [2]. IEEE 1149.6 a fost introdus pentru testarea circuitelor LVDS (Low Voltage Data (Differential) Signal) și a circuitelor cuplate capacitiv. Testarea se realizează cu impulsuri generate cu un bit în BSR (Boundary Scan Register) și recepționate pe 2 biți (prin recepție diferențială). Acest mod de testare se mai numește și AC-EXTEST.

IEEE 1532 a fost introdus în anul 2000 pentru a standardiza programarea PLD și FPGA, indiferent de producătorul circuitului. Algoritmul care programează, șterge, citește și verifică circuitul se află într-un fișier BSDL (Boundary Scan Description Language).

9.4. Interfața JTAG la microcontrollerul Atmega64

La familia ATmega interfața JTAG poate fi folosită la testarea plăcii prin scanare, la programarea memoriei nevolatile și la debugging.

Funcționarea JTAG este validată de un bit cu înscriere fuzibilă (fuse bit JTAGEN) care inițial este programat pentru validarea JTAG. Pe lângă această validare este nevoie ca bitul JTD din MCUCSR (MCU Control and Status Register) să fie setat. Pentru a evita programările false, aplicația trebuie să repete setarea acestui bit, adică să execute de două ori instrucțiunea de scriere a bitului JTD. Dacă interfața JTAG nu este validată atunci pinii au semnificații de port paralel de uz general și controllerul TAP este în Reset. Dacă se dorește programarea MC prin JTAG și bitul JTD nu este setat atunci se forțează linia de Reset în Low două tacte și bitul JTD este setat.

TAP (Test Access Port) este un automat cu 16 stări care controlează funcționarea interfeței JTAG. Navigarea între stările TMS se face cu comenzi pe pinul TMS (comenzi seriale pe 4 biți). Întotdeauna LSB este trimis ca prim bit în cuvintele seriale JTAG.

9.4.1. Programarea memoriei prin JTAG

Programarea memoriei Flash și EEPROM prin JTAG se poate face după validarea interfeței JTAG. Fiecare instrucțiune generată printr-un cuvânt serial de 4 biți trimis pe linia TMS acționează asupra unui registru care devine după instrucțiune registru de date. Datele se introduc în acest registru sau se extrag pe liniile TDI respectiv TDO. Instrucțiunile care pot fi trimise serial către TAP sunt (valoarea hexa este dată în paranteză):

- AVR_RESET (CH) se resetează MC fără a se reseta controllerul TAP. Reset Register este selectat ca registru de date și durata reset-ului este dată de valoarea acestui registru;
- PROG_ENABLE (4H) se validează scrierea memoriei Flash sau EEPROM prin JTAG, registrul Programming Enable Register este selectat ca registru de date de 16 biți;
- PROG_COMMANDS (5H) registrul Programming Command Register este selectat ca registru de date de 15 biți;
- PROG_PAGELOAD (6H) registrul de 1024 biți Virtual Flash Page Load Register este selectat ca registru de date, egal cu numărul de biți dintr-o pagină de memorie Flash;
- PROG_PAGEREAD (7H) registrul de 1032 biți Virtual Flash Page Read Register este selectat ca registru de date. În acest registru se citesc cei 1024 biți dintr-o pagină Flash plus 8 biți.

După instrucțiunea PROG_ENABLE pe linia TDI se introduce o secvență 1010 0011 0111 0000 (A370H) numită semnătura de programare pentru a valida programarea în Programming Enable Register. După terminarea programării registrul trebuie resetat.

După validarea programării, cu instrucțiunea PROG_COMMANDS se introduce în registrul Programming Command Register o comandă care realizează citirea/ scrierea/ ștergerea memoriei, a biților fuzibili și a biților de blocare. Comenzile sunt date în foile de catalog a MC. De exemplu citirea memoriei Flash se face prin introducerea în ordine pe linia TDI, după instrucțiunea PROG_COMMANDS:

0100011 00000010 pe TDI este comanda de citire Flash;

0000111 aaaaaaaa pe TDI este adresa, octetul superior;

0000011 bbbbbbbb pe TDI este adresa, octetul inferior;

0110010 00000000 pe TDI comanda de citire a locației;

0110110 00000000 pe TDO apare octetul inferior în forma xxxxxxxx 00000000;

0110111 00000000 pe TDO apare octetul superior.

La fel există secvențe pentru citirea /scrierea Flash, citirea/scrierea EEPROM, citirea/scrierea biților fuzibili și ai celor de blocare.

9.4.2.Debugging prin JTAG

Debugging-ul prin JTAG se bazează pe existența unei căi de scanare între CPU și modulele interne existând posibilitatea inserării de maximum 4 Break Point-uri. Ca și instrument software se poate folosi AVR Studio, modul debug fiind posibil atât în limbaj de asamblare cât și în C. Pentru a lucra în modul debug trebuie ca bitul JTAGEN să fie programat și nici un bit de blocare să nu fie programat. Orice bit de blocare programat duce la blocarea sistemului de debug din rațiuni de securitate.

În mod debug utilizatorul poate executa un program pas cu pas, poate trece peste anumite instrucțiuni, poate executa un program până la atingerea unei anumite stări, poate opri sau reseta execuția. Instrucțiunile JTAG pentru debug (8H, 9H, AH, BH) sunt considerate private și sunt documentate doar pentru firmele care vând software pentru Atmel.

Pentru a ușura munca de debugging Atmel a creat un emulator în circuit numit JTAGICE mkII, figura 9.16.

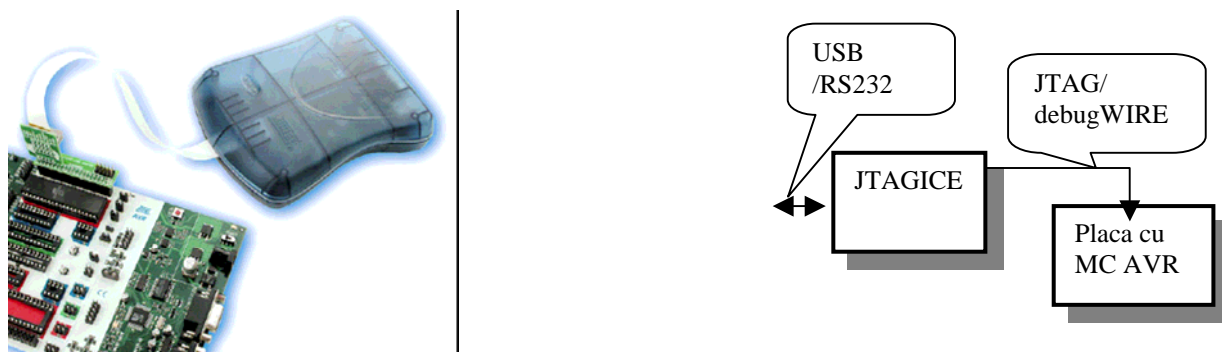


Figura 9.16: JTAGICE mkII

JTAGICE permite accesul la toate resursele microcontrollerelor din familia AVR. Locul acestui dispozitiv este dat în figura 9.16 dreapta, între un PC care rulează AVR Studio și placa pe care este plasat microcontrollerul. Interfața cu PC-ul poate fi RS232 sau USB. Pentru MC cu număr mic de pini a fost introdusă ca interfață de testare debug/WIRE în care comunicația se face pe un singur pin, cel de Reset. JTAGICE poate realiza:

1. Emularea tuturor funcțiilor analogice și digitale ale MC AVR de la ATmega 16 la ATmega 6490 prin JTAG și de la Attiny 13 la ATmega 168 prin debug/WIRE. La apariția unor modele noi de MC, prin descărcarea unei noi variante de AVR Studio JTAGICE va putea lucra și cu modelele noi;
2. Programarea MC clasică sau prin JTAG;
3. Mod debug în care se pot introduce Break Point-uri, când programul ajunge la o adresă sau într-o gamă de adrese, când datele sunt citite/ scrise la o adresă sau într-un interval de adrese, la schimbarea fluxului de citire a programului din memoria program;
4. Monitorizarea tuturor resurselor interne ale MC.

Conectorii de acces pe placă sunt recomandați de Atmel cu configurația pinilor dată în figura 9.17, acces SPI și debug prin debug/WIRE (stânga) și JTAG (dreapta).

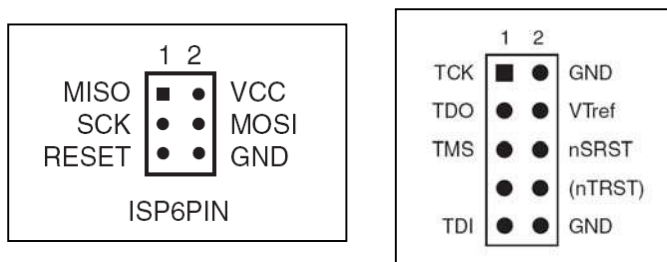


Figura 9.17: conectori recomandați de Atmel

Pentru debug de tip debug/WIRE, bitul care validează operarea este neprogramat, ca urmare este nevoie de programarea lui, ceea ce se realizează prin programarea SPI. La conectorul JTAG semnalul nTRST este un Reset al interfeței JTAG și nu este folosit.

9.4.3. Testarea prin JTAG

Calea de testare Boundary Scan are rolul de a stabili valori logice și de a le citi la pinii de I/O. Toate circuitele care au interfață JTAG se conectează într-un lanț prin TDI/TDO, testarea formând un cuvânt serial lung. Secvența de test este generată de un controller exterior care apoi citește rezultatul și îl compară cu cel așteptat. Se recomandă ca MC să fie în Reset pe timpul testării, altfel, la ieșirea din mod test MC va rămâne într-o stare nedeterminată. Intrarea în Reset se poate face comandând pinul RESET cu un nivel low sau executând instrucțiunea JTAG proprietară AVR_RESET. După intrarea în Reset instrucțiunea JTAG HIGHZ devine inutilă. Pentru a putea lucra în acest mod bitul JTAGEN trebuie să fie programat și bitul JTD din registrul MCUCSR trebuie să fie 0. De asemenea se recomandă ca tactul TCK de test să fie mai mare decât tactul intern. Instrucțiunile utilizate pentru testare sunt:

- EXTEST (0H), instrucțiunea selectează calea de scanare ca registru de date pentru testarea conexiunilor exterioare. Pentru pinii de I/O se poate realiza invalidarea Pull-up, se poate stabili direcția, se poate seta sau reseta un pin;
- IDCODE (1H), instrucțiunea (opțională) selectează registrul de identificare (ID) de 32 de biți ca registru de date. Registrul ID conține: versiunea MC(4 biți), tipul MC (16 biți, ATmega 64 are 9602H) și fabricantul pe 11 biți (Atmel are 01FH);
- SAMPLE_PRELOAD (2H), instrucțiunea permite inspectarea pinilor de I/O fără a afecta funcționarea MC și de a preîncărca datele de test în latch-ul de ieșire, fără a le transmite la pinii de ieșire. Calea de scanare este selectată ca registru de date;
- AVR_RESET (CH), MC este forțat în Reset, mai puțin controllerul TAP. Instrucțiunea este folosită și la programare;
- BYPASS (FH), se selectează registrul Bypass ca registru de date. Când este selectat acest registru ca și cale între TDI și TDO se scurtează calea de scanare, mod utilizat când se testează alte circuite JTAG din sistem.

O schemă bloc simplificată a unui pin de I/O testabil JTAG (schema completă este dată în foile de catalog) este dată în figura 9.18.

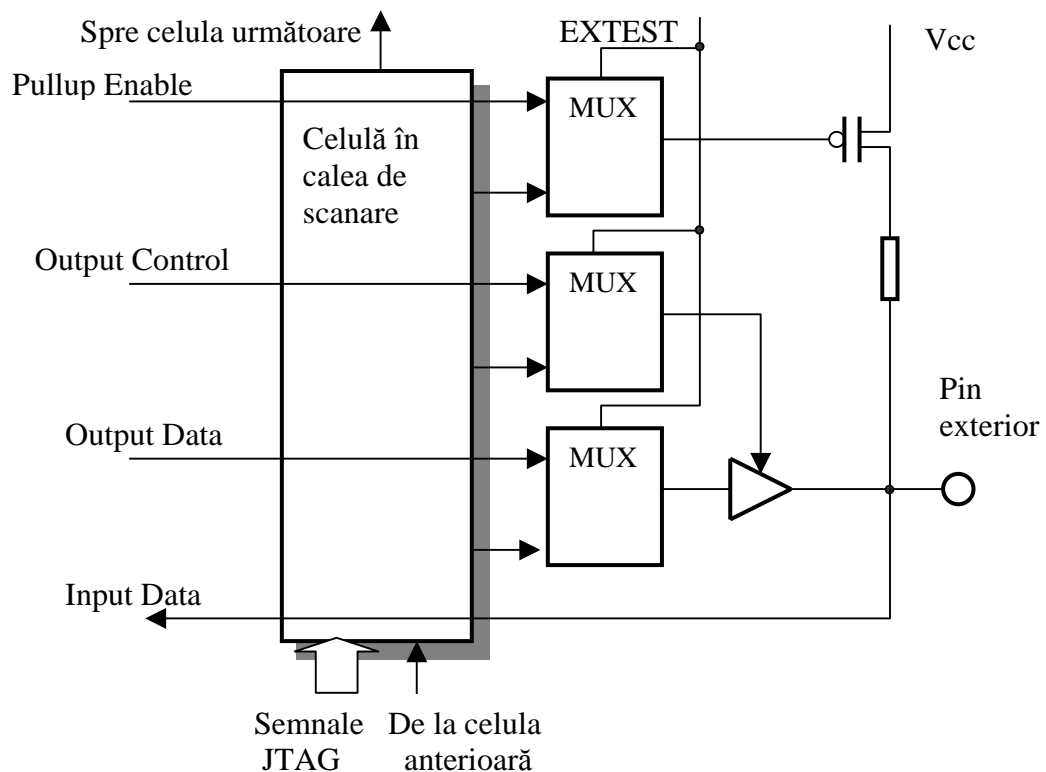


Figura 9.18: Schema simplificată a unui pin testabil JTAG

Multiplexoarele comandă funcționarea normală sau testarea JTAG. Celula de testare este conectată în lanțul JTAG. Calea de testare la un ATmega 64 are 205 biți și poate fi găsită în foile de catalog. De exemplu, primii biți din lanț sunt alocați portului F astfel:

Bit	Pin	Semnificație
0	PF0	Pullup Enable
1	PF0	Control
2	PF0	Data
3	PF1	Pullup Enable
4	PF1	Control
4	PF1	Data
.....

Testarea modulelor de interfață integrate în MC impune niște precauții. Astfel de exemplu la modulul de interfață serială TWI (Two Wire Interface, compatibil cu I²C), se pot testa în mod normal cei 2 pini SCL și SDA și este accesibil semnalul de control TWIEN. Acest semnal este accesibil pentru a dezactiva un bloc din componența interfeței care controlează viteza de creștere / descreștere a frontului semnalelor. Setarea TWIEN și OC (Output control) poate duce la rezultate imprevizibile.

O celulă specială care permite doar observarea este alocată semnalului de Reset. Tactul are de asemenea celule speciale, doar de observare. MC admite o gamă largă de semnale de tact (RC intern, tact extern, cristal extern, rezonator ceramic) și validarea tactului este realizată cu celule de testare normale. Comparatorul analogic și convertorul analog digital au celule de test cu structuri speciale.

Pentru a ușura testarea circuitelor, un fișier de stimuli și răspunsuri de referință este în general disponibil pe site-urile producătorilor. Aceste fișiere se numesc fișiere **BSDL** (Boundary Scan Description Language).

9.5.Unelte pentru testare JTAG

Testarea plachetelor cu circuite se poate realiza la scară diferită în diferite situații- pe scară largă post producție în firmele specializate sau în proiectare unde se testează un număr mic de plăci. Din acest motiv echipamentele de testare JTAG sunt foarte diverse. O selecție din controllerele JTAG de productivitate mică prezentate în [2] este:

- Controllere sub forma unor module separate, prevăzute cu interfață USB, Ethernet și FireWire, cu tact TCK până la 40MHz, figura 9.19 stânga;
- Plachete pentru conectarea pe magistrale de tip PXI, PCI sau PCIE, cu tact TCK până la 40MHz;
- Controllere compatibile DIMM care pot fi introduse în conectori DIMM;
- Module de dimensiuni mici și ieftine cu interfață USB, TCK până la 6MHz, figura 9.19 mijloc;
- Montabile în dulapuri standard (Rack Mountable), figura 9.19 dreapta;
- Plăci cu controller JTAG și spațiu de dezvoltare pentru utilizatori.

Toate aceste variante pot avea 1 până la 4 controllere TAP și pot fi prevăzute cu pini de I/O pentru preluarea semnalelor (de exemplu cel montabil în dulap are 256 pini de I/O).



Figura 9.19: Controllere JTAG- JT 37x7/TSI independent, JT 3705/USB cel mai ieftin, JT 37x7/RMI montabil în dulap standard de 19"

În [3] este prezentat un aparat de testare AEROFLEX de mare productivitate, figura 9.20 stânga, prevăzut pe lângă JTAG cu 2048 de canale de testare prevăzute cu pini. În [4], figura 9.20 dreapta este prezentat aparatul Digital Test cu 1400 de canale de testare. Cu aceste aparate se pot face teste automate la ieșirea de pe linia de producție și se pot implementa strategii de testare. De regulă, în activitatea industrială de mare productivitate testarea JTAG este combinată cu testarea cu pat de ace.



Figura 9.20: Aparatul AEROFLEX 4250 și Digital Test Eagle MTS180

Partea software pentru aplicațiile JTAG poate fi împărțită în două mari clase, cu nenumărate variante:

- Pe stații dedicate, cum este aplicația AEX (Application Executor) [2], în care sunt definite nivele de acces. Cel care a proiectat placa își poate defini secvența de test JTAG și stabilește strategia de testare, în pași dependenți de pașii anteriori de testare. La cel mai jos nivel de acces operatorul poate rula secvența de testare și observă rezultatele;
- În anumite situații este util ca secvențele de test să fie încorporate în programele utilizate și care sunt familiare utilizatorului. Astfel există secvențe realizate în LABVIEW, pentru C sau pentru Visual Basic. Acest tip de programe se numesc PIP.

Un software important este cel de verificare BSDL (Boundary Scan Description Language), care verifică discrepanțele între fișierul de verificare BSDL și circuitul de testat, verifică

erorile de sintaxă și de semantică. De asemenea se poate crea un fișier BSDL dacă există circuitul fizic.

Bibliografie

[1] www.polarinstruments.com

[2] www.jtag.com

[3] http://www.aeroflex.com/ats/products/product/ATE/Test_Systems/

[4] <http://www.digitaltest.de/digitaltest.php?loadtype=products&lang=en&filename=mts180>