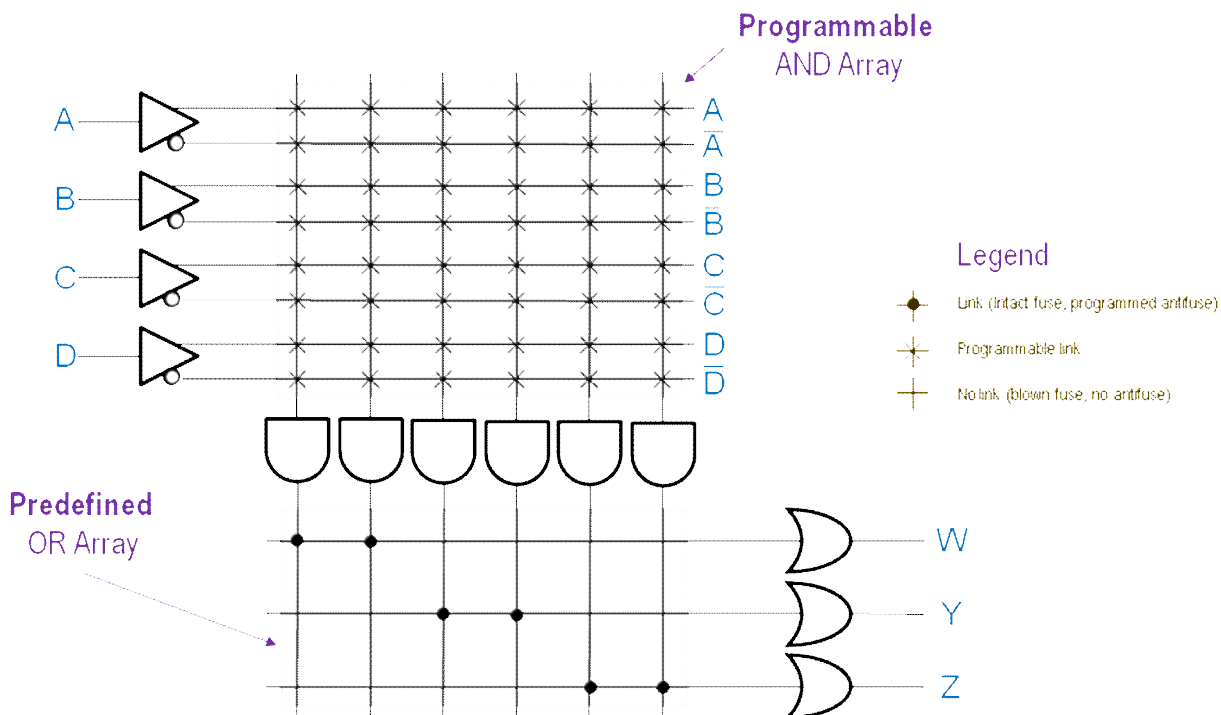


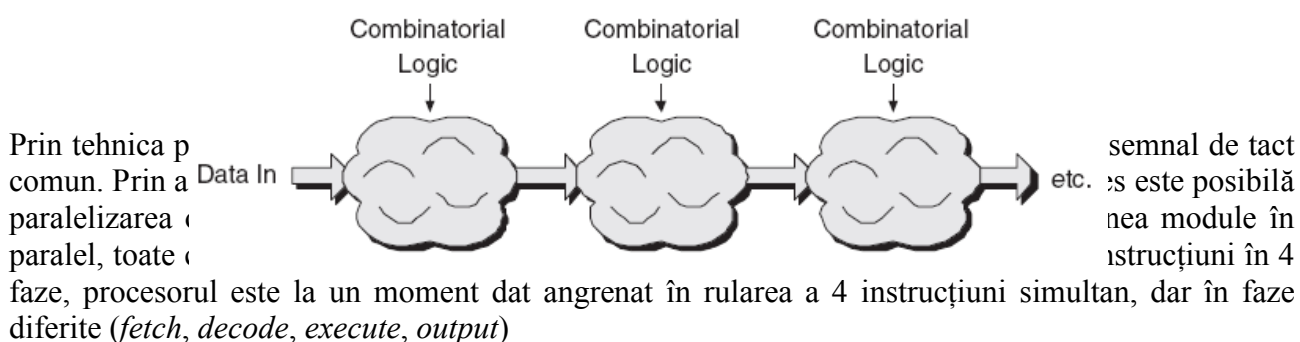
2. Arhitectura PAL și GAL: schemă, funcționare, avantaje și dezavantaje



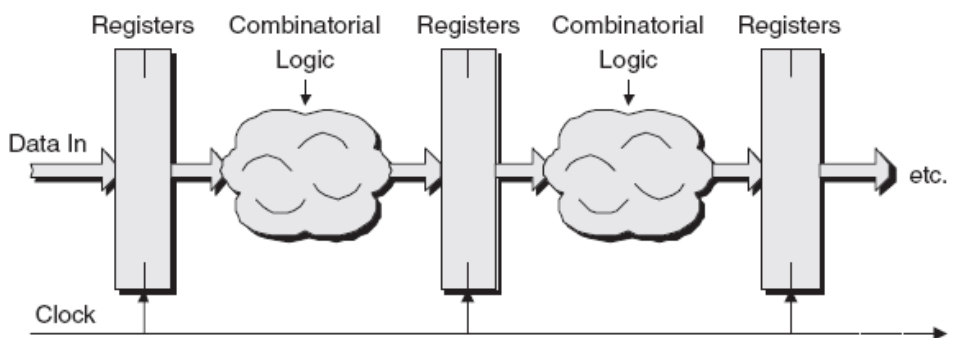
Structurile PAL și GAL au o arhitectură comună, formată dintr-o rețea de porți ȘI programabile și o rețea de porți SAU fixe. Avantajul principal este simplitatea arhitecturii, dezavantajul este ineficiența ei (un termen comun în două funcții se generează de două ori). Circuitele PAL se bazează pe tehnologia bipolară (nu se mai folosesc), iar cele GAL pe tehnologia CMOS (se mai folosesc uneori).

3. Explicarea arhitecturii pipeline

Un sistem numeric complet se compune dintr-o parte combinațională și una secvențială. În figura de mai jos este prezentată simplificat o secvență de operații (prelucrări) combinaționale.



Prin tehnica p... comun. Prin a... paralelizarea... paralel, toate... faze, procesorul este la un moment dat angrenat în rularea a 4 instrucțiuni simultan, dar în faze diferite (*fetch, decode, execute, output*)



4. Elemente structurale ale unui cod VHDL (enumerare, descriere sumară)

Un program VHDL este format din următoarele secțiuni:

1. Declararea de biblioteci (*LIBRARY declarations*): cuprinde lista tuturor bibliotecilor utilizate în proiect, de exemplu: *ieee*, *std* (implicită), *work* (implicită), etc.
2. **ENTITY**: specifică porturile circuitului, materializate prin pini de intrare, ieșire, intrare-ieșire sau de tip buffer.
3. **ARCHITECTURE**: descrie modul în care circuitul se comportă. Pot exista mai multe arhitecturi (funcționale) pentru aceeași entitate.

5. Moduri de descriere în VHDL (structural, flux de date, comportamental) – scurtă descriere, avantaje, dezavantaje, aria de utilizare

Un sistem numeric se poate descrie în oricare din următoarele variante:

- **structural**: sistemul este compus din mai multe blocuri, descrise separat și conectate asemănător unei liste de legături dintr-o schemă electrică prin instrucțiuni specifice VHDL (component, port map). Este o variantă laborioasă, dar foarte utilă pentru proiectele ierarhice.
- **dataflow**: se scriu direct ecuațiile boolene care descriu funcționarea circuitului.
- **behavioral**: se descrie (de obicei prin algoritmi) funcționarea circuitului. Este cea mai puternică variantă, deoarece nu face referire la tehnologie sau furnizor. În plus dimensiunea unui circuit anume nu influențează codul (un numărator pe 10 biți are aceeași descriere ca unul pe 32 de biți).

6. Comparație între semnale și variabile în VHDL (mod de asignare)

Semnalele reprezintă unicul mijloc de comunicare între procese.

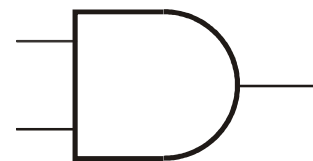
	SIGNAL / Semnal	VARIABLE/ Variabilă
Atribuire	<=	:=
Utilizare	Folosite pentru conectarea circuitelor / modulelor (<i>wires</i>)	Reprezintă informație locală
Vizibilitate	Globală	Locală (vizibile doar în PROCESS, FUNCTION, sau PROCEDURE)
Comportare	Actualizarea nu este imediată datorită proprietății de memorare (noua valoare se generează doar la încheierea PROCESS, FUNCTION, sau PROCEDURE)	Actualizare imediată (noua valoare se poate folosi în următoarea linie de cod)
Locul de declarare	În PACKAGE, ENTITY, sau ARCHITECTURE. În ENTITY, toate legăturile de PORT sunt de tip SIGNAL	Doar în cod secvențial (doar în PROCESS, FUNCTION, sau PROCEDURE)

7. Modelarea unui circuit în VHDL

Un prim exemplu este modelarea unui circuit ȘI cu două intrări, la care timpul de propagare este de 5 ns.

```
entity and2 is
  port ( a, b : in bit;
        y : out bit );
end and2;
```

```
architecture and2_behav of and2 is
begin
  y <= a and b after 5 ns;
end and2_behav;
```



Modelare întârziere

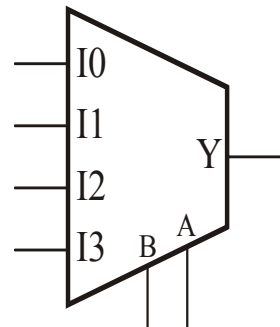
Un al doilea exemplu prezintă un MUX 4:1 cu un timp de propagare identic pentru cele 4 intrări, de 10 ns.

```
library ieee;
use ieee.std_logic_1164.all;

entity mux4 is
  port ( I0, I1, I2, I3, a, b: in std_logic;
        Y : out std_logic);
end mux4;
```

```
architecture mux4 of mux4 is
  signal sel: integer;
begin
  sel <= 0 when A = '0' and B = '0' else
        1 when A = '1' and B = '0' else
        2 when A = '0' and B = '1' else
        3 when A = '1' and B = '1' else
        4 ;
  with sel select
    Y <= I0 after 10 ns when 0,
    Y <= I1 after 10 ns when 1,
    Y <= I2 after 10 ns when 2,
    Y <= I3 after 10 ns when 3,
    Y <= 'X' after 10 ns when others;
```

```
end mux4;
```



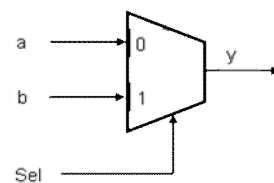
8. Multiplexor în VHDL, exemplu de cod

- Prima soluție (MUX 2:1, canalul de 1 bit)

```
library ieee;
use ieee.std_logic_1164.all;
```

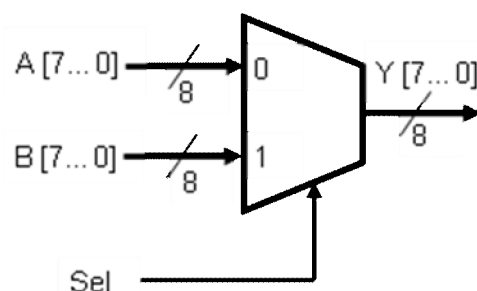
```
entity mux2to1 is
  port ( sel: in std_logic;
        a, b: in std_logic;
        y: out std_logic);
end mux2to1;
```

```
architecture bhv of mux2to1 is
  begin
    y <= a when sel = '0' else b;
  end bhv;
```



- A doua soluție (MUX 2:1, canalul de 8 biți)

```
library ieee;
use ieee.std_logic_1164.all;
```



```

entity mux2to1_8bit is
port (sel: in std_logic;
      A,B: in std_logic_vector (7 downto 0);
      Y: out std_logic_vector (7 downto 0));
end mux2to1;

```

```

architecture bhv of mux2to1_8bit is
begin
  Y <= A when sel = '0' else B;
end bhv;

```

Trebuie remarcată similitudinea celor 2 exemple de mai sus (diferă doar prin modul de declarare al semnalelor A și B).

- A treia soluție (MUX 4:1, canalul de 1 bit)

```

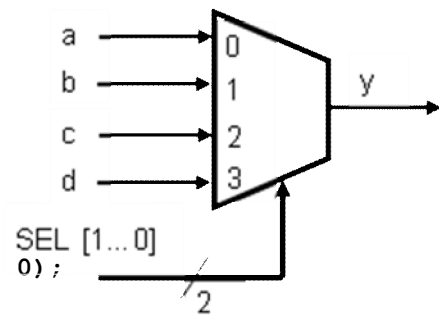
library ieee;
use ieee.std_logic_1164.all;

```

```

entity mux4to1 is
port (a, b, c, d: in std_logic;
      SEL: in std_logic_vector (1 downto 0);
      y: out std_logic);
end mux4to1;

```



```

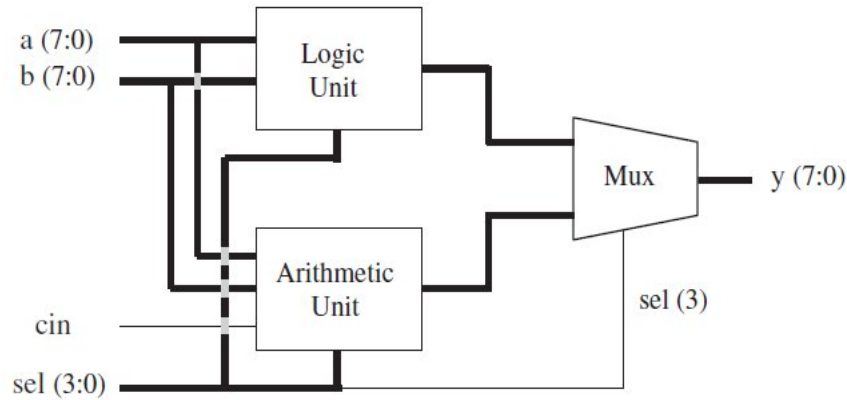
architecture mux1 of mux4to1 is
begin
  y <= (a and not SEL(1) and not SEL(0)) or
       (b and not SEL(1) and SEL(0)) or
       (c and SEL(1) and not SEL(0)) or
       (d and SEL(1) and SEL(0));
end mux1;

```

Notă: în curs sunt prezentate zece variante distincte de definiție pentru acest MUX 4:1. Oricare dintre ele este potrivită pentru o rezolvare corectă.

9. ALU în VHDL, exemplu de cod

ALU (Arithmetic Logic Unit) prezentat mai jos permite efectuarea de opt operații aritmetice și opt operații logice definite în tabelul de adevăr. Este alcătuit din două module distincte care procesează simultan informația – selecția finală fiind realizată în funcție de bitul cel mai semnificativ din *sel* prin intermediul unui multiplexor. Ceilalți trei biți din *sel* sunt utilizați pentru selecția operației specifice aritmetice sau logice.



Soluția prezentată utilizează doar cod concurrent (combinațional) și date de un singur tip atât pentru operațiile aritmetice cât și logice. Acest lucru e posibil deoarece a fost declarat un package specific - `std_logic_unsigned` din biblioteca `ieee`.

Două semnale, `arith` and `logic`, sunt folosite pentru transmiterea rezultatelor blocurilor aritmetic și logic către multiplexor.

sel	Operație	Descriere	Modulul activ
0 000	Y <= A	Transfer A	Aritmetic
0 001	Y <= A + 1	Increment A	
0 010	Y <= A - 1	Decrement A	
0 011	Y <= B	Transfer B	
0 100	Y <= B + 1	Increment B	
0 101	Y <= B - 1	Decrement B	
0 110	Y <= A + B	Add A and B	
0 111	Y <= A + B + Cin	Add A and B with carry	
1 000	Y <= not A	Complement A	Logic
1 001	Y <= not B	Complement B	
1 010	Y <= A and B	AND	
1 011	Y <= A or B	OR	
1 100	Y <= A nand B	NAND	
1 101	Y <= A nor B	NOR	
1 110	Y <= A xor B	XOR	
1 111	Y <= A xnor B	XNOR	

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity ALU is
port (A, B: in std_logic_vector (7 downto 0);
      sel: in std_logic_vector (3 downto 0);
      cin: in std_logic;
      Y: out std_logic_vector (7 downto 0));
end ALU;

architecture dataflow of ALU is
  signal arith, logic: std_logic_vector (7 downto 0);
begin

```

```

----- Arithmetic unit: -----
with sel(2 downto 0) select
    arith <= A  when "000",
           A+1 when "001",
           A-1 when "010",
           B   when "011",
           B+1 when "100",
           B-1 when "101",
           A+B when "110",
           A+B+cin when others;

----- Logic unit: -----
with sel(2 downto 0) select
    logic <= not A when "000",
           not B when "001",
           A and B when "010",
           A or B when "011",
           A nand B when "100",
           A nor B when "101",
           A xor B when "110",
           A xnor B when others;

----- Mux: -----
with sel(3) select
    Y <= arith when '0',
         logic when others;
end dataflow;

```

10. Numărător modulo- m în VHDL, exemplu de cod

Un numărător modulo- m parcurge ciclic un număr de stări egal cu modulul său (de obicei de la 0 la $m - 1$) după care reia acest ciclu. În VHDL este utilă crearea unui numărător *parametric* caracterizat de două constante generice:

- M , specifică modulul m și
- N , care specifică numărul de biți (și implicit de bistabile) pentru numărător

$$M = \lceil \log_2 M + 0.5 \rceil$$

Un secvență de cod mai avansat ar trebui să elimine redundanța dintre M și N , M fiind *unicul* parametru utilizat. Logica pentru starea următoare este realizată printr-o atribuire condițională. Atunci când numărătorul atinge valoarea $M - 1$, va fi reinițializat. În caz contrar este incrementat (cu o unitate).

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity mod_m_counter is
    generic ( N: integer := 4;      -- number of bits
             M: integer := 10);   -- mod-M

    port ( clk, reset : in std_logic;
          max_tick : out std_logic;
          Q: out std_logic_vector (N-1 downto 0));

```

```

end mod_m_counter;

architecture arch of mod_m_counter is
    signal r_reg: unsigned (N-1 downto 0);
    signal r_next: unsigned (N-1 downto 0);
begin
    process (clk, reset)
    begin
        if (reset='1') then
            r_reg <= (others => '0');
        elsif (clk'event and clk = '1') then
            r_reg <= r_next;
        end if;
    end process;
    r_next <= (others => '0') when r_reg = (M-1) else
        r_reg + 1;
    Q <= std_logic_vector (r_reg);
    max_tick <= '1' when r_reg = (M-1) else '0';
end arch;

```

-- register

-- next-state logic

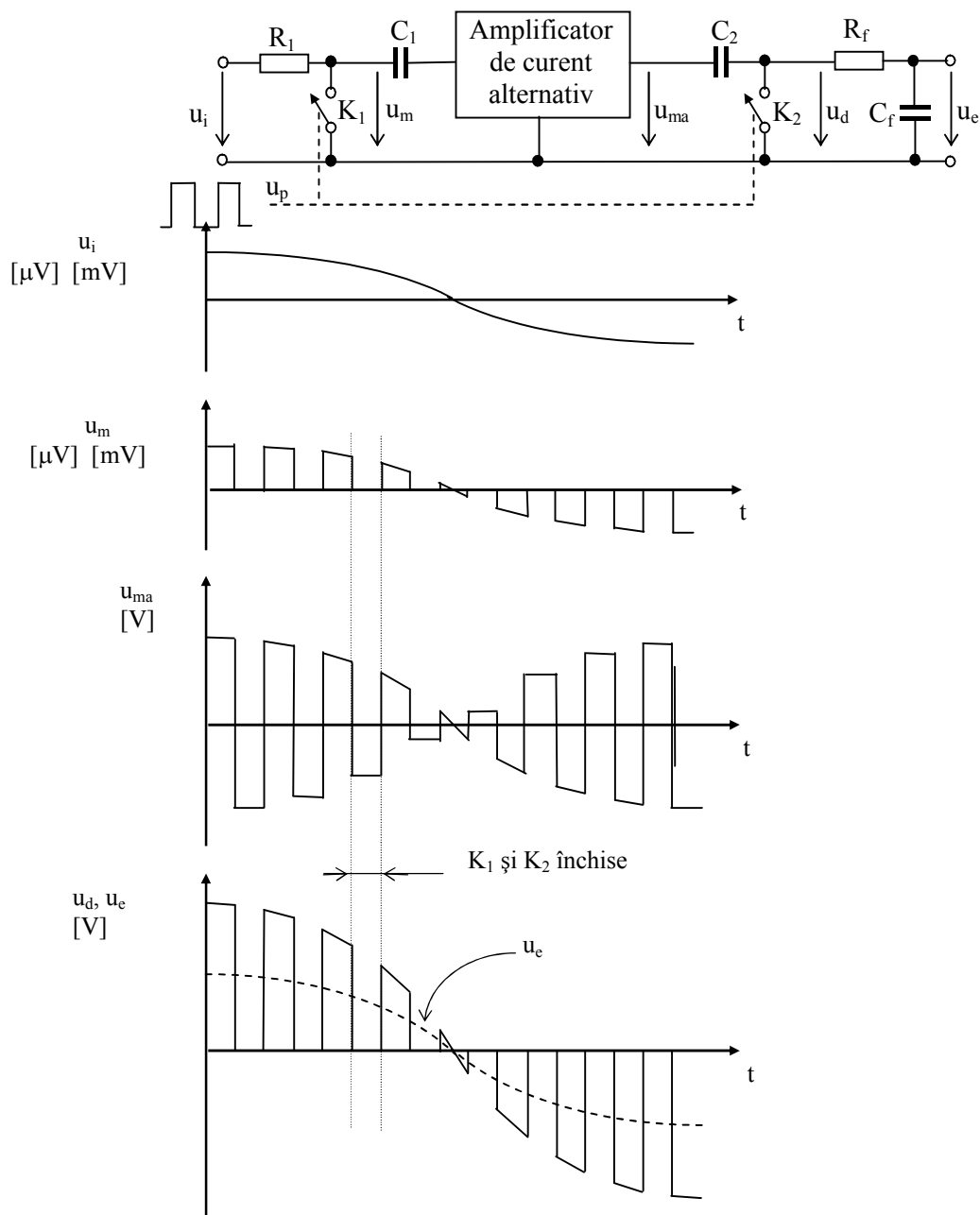
-- output logic

Sisteme de achiziții de date

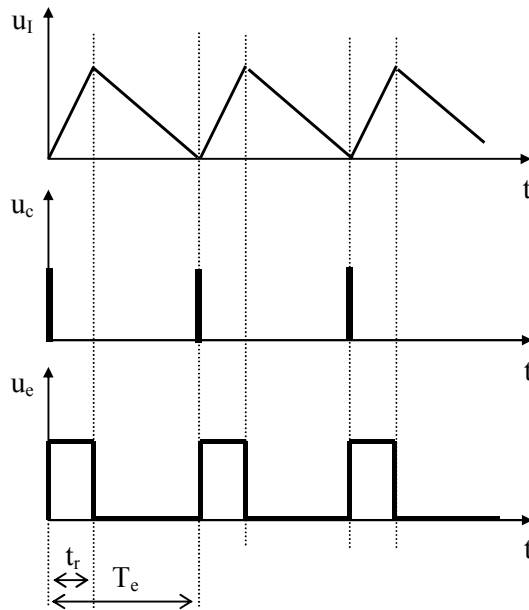
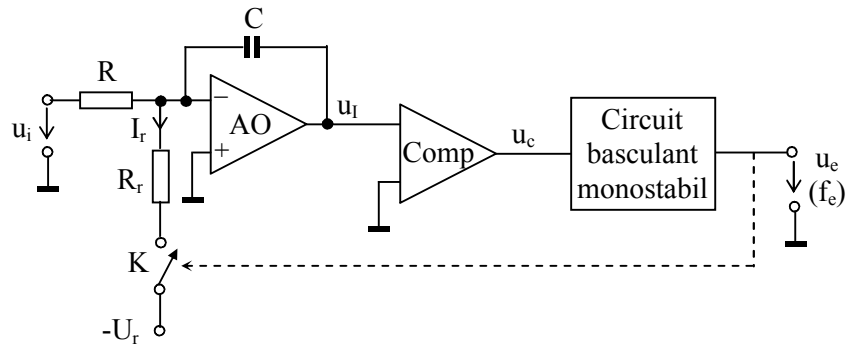
Anul III

1. Amplificator cu modulare-demodulare cu chopper – utilizare, structura de principiu, explicarea funcționării pe bază de diagrame de timp.

Amplificatorul cu modulare-demodulare (AMD) este utilizat pentru amplificarea cu 60÷100 dB a unor tensiuni de nivel foarte mic (microvolți sau milivolți) și de joasă frecvență.



2. Convertor tensiune-frecvență – structura de principiu, explicarea funcționării pe bază de diagrame de timp, funcția de transfer.



Din $\frac{u_l}{R} T_e = \frac{U_r}{R_r} t_r$ rezultă funcția de transfer $f_e = \frac{R_r}{R U_r t_r} u_i$.

3. Convertor numeric analogic – relație de calcul a cuantei, funcția de transfer (ieșire curent, ieșire tensiune), definiții pentru eroarea de decalaj (de zero) și eroarea de câștig.

Calculul cuantei $q = 1LSB = \frac{FSR}{2^N} = \frac{FSR}{D_{max} + 1} = \frac{FR}{D_{max}}$, FSR – capăt de scală, FR – domeniul maxim al ieșirii,

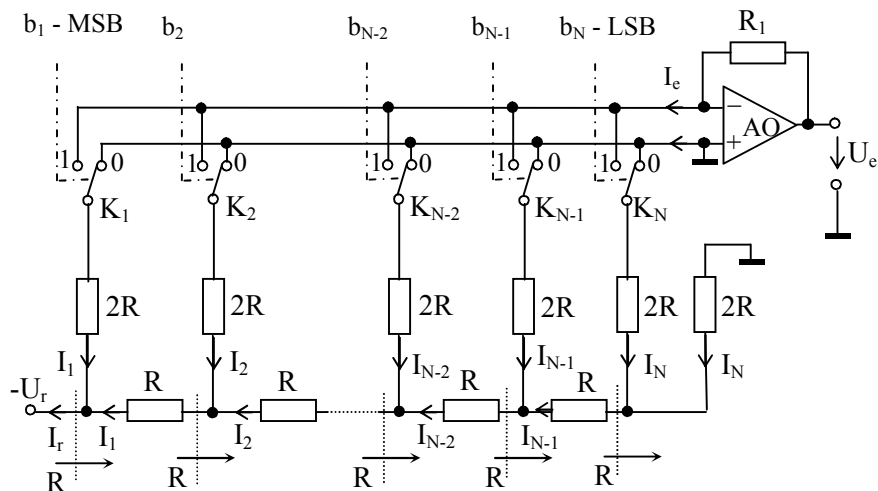
N – numărul de biți.

Funcție de transfer ieșire curent $I_e = \frac{U_r}{2^N \cdot R} \cdot D_{CBN}$, ieșire tensiune $U_e = \frac{R_1 \cdot U_r}{2^N \cdot R} \cdot D_{CBN}$ unde U_r – tensiune de referință, R – rezistență de referință, R_1 – rezistență din reacția negativă a AO, D_{CBN} – valoarea în zecimal a numărului de la intrare scris în CBN.

Eroare de zero (Δ_0) (eroare absolută) reprezintă valoarea mărimii de ieșire dacă la intrare s-a aplicat codul numeric pentru ieșire zero.

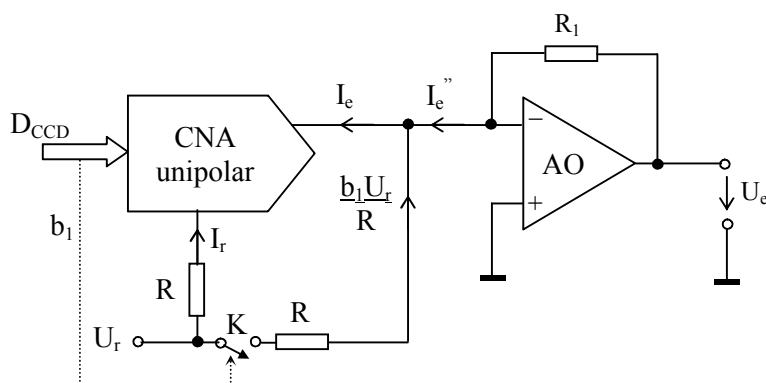
Eroare de câștig (Δ_{FR}) reprezintă eroarea absolută a domeniului maxim al ieșirii. Dacă eroarea de zero nu este zero trebuie ținut cont de ea la calculul erorii de câștig

4. Convertor numeric analogic cu rețea de rezistențe R-2R – structura de principiu, funcție de transfer.



Funcția de transfer $U_e = I_e R_1 = \frac{U_r R_1}{2^N R} D_{CBN}$.

5. Conectarea unui convertor numeric analogic unipolar pentru conversie bipolară în codul complementului lui doi – structura de principiu, deducere matematică a funcției de transfer.



Din $I_e = \frac{U_r}{2^N R} D_{CBN}$ și $D_{CBN} = D_{CCD} + b_1 2^N$ rezultă succesiv $I_e = \frac{U_r}{2^N R} (D_{CCD} + b_1 2^N)$,
 $I_e - b_1 \frac{U_r}{R} = \frac{U_r}{2^N R} D_{CCD}$, $I_e'' = I_e - b_1 \frac{U_r}{R}$, $U_e = I_e'' \cdot R_1 = \frac{U_r R_1}{2^N R} D_{CCD}$.

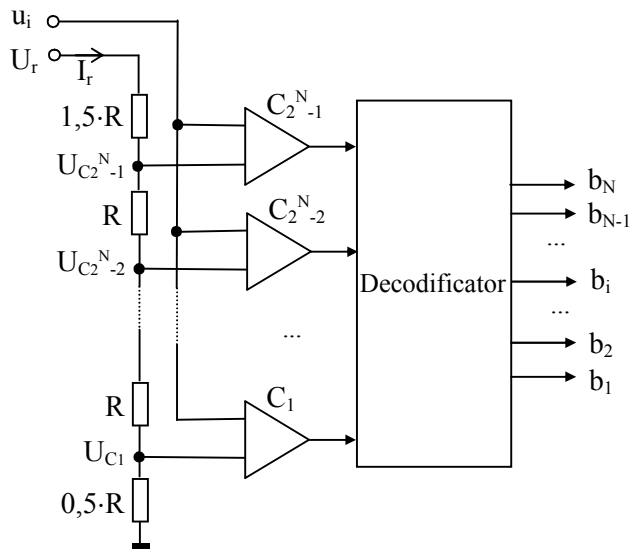
6. Convertor analog numeric cu comparare de tip paralel – cuanta, structura de principiu, calculul tensiunilor de comparare.

Cuanta unui CAN se calculează cu relația $q = 1LSB = \frac{FSR}{2^N}$, unde FSR este intervalul de variație a semnalului

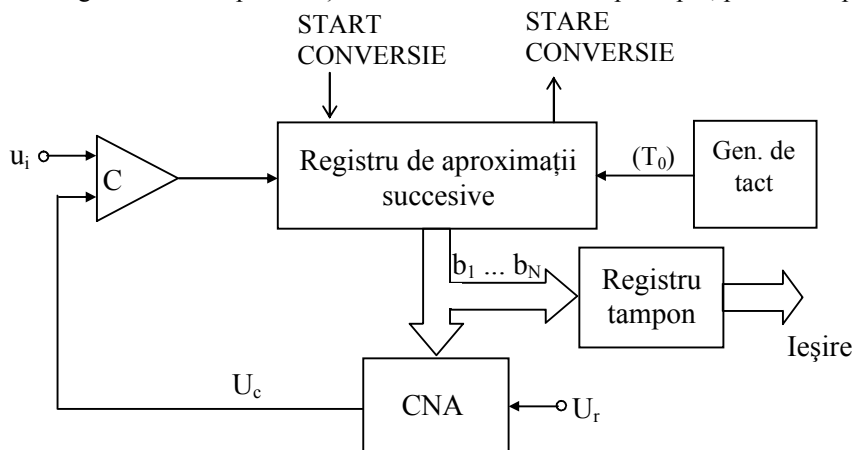
de intrare. Pentru un CAN unipolar $1LSB = \frac{FSR}{2^N} = \frac{U_r}{2^N}$.

Tensiunile de comparare utilizate în procesul de conversie pentru un CAN de tip paralel vor avea valorile:

$$U_{Ci} = (i - 0,5) \frac{U_r}{2^N}, i = 1, 2, \dots, 2^N - 1, \text{ unde } N \text{ este numărul de biți.}$$



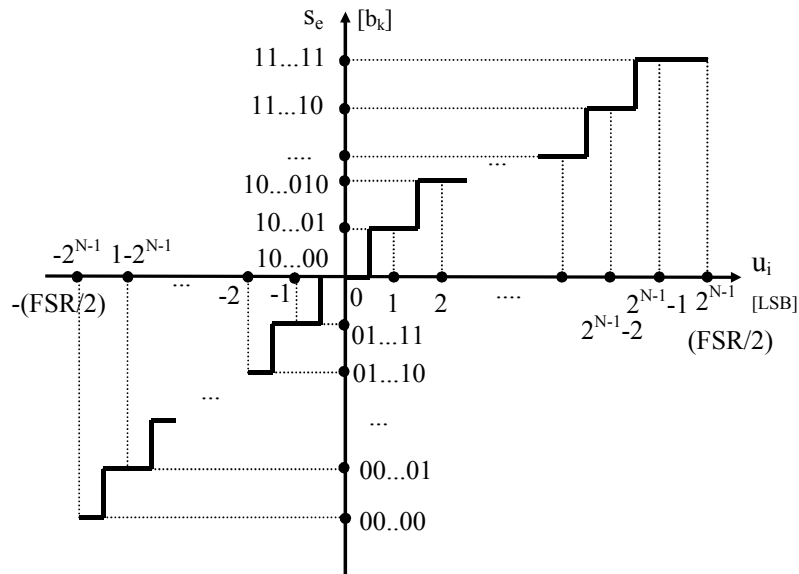
7. Converter analog numeric cu aproximații succesive – structura de principiu, prezentare principiu de conversie.



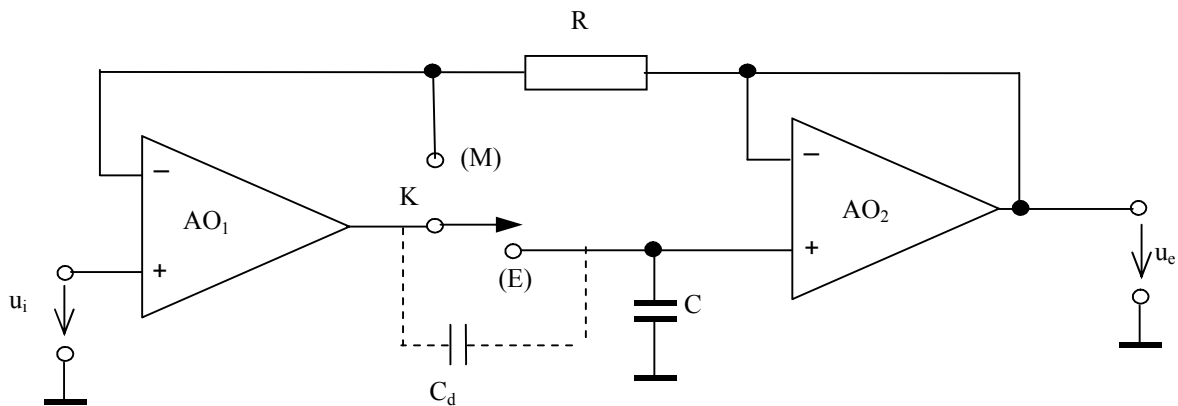
Valoarea fiecărui bit se obține în câte o perioadă T_0 a semnalului de tact. Stabilirea valorilor începe cu bitul b_1 (MSB) și se termină cu bitul b_N (LSB). Registrul de aproximații succesive comandă intrările CNA pentru obținerea unor tensiuni de comparare U_{ck} cu valorile $U_{ck} = U_r \sum_{i=1}^k b_i 2^{-i}$, $k = 1, 2, \dots, N$, unde $b_k = 1$ iar biții cu indici mai mari decât k au valoarea 0. Dacă în urma comparării rezultă $u_i \geq U_{ck}$ atunci bitul b_k rămâne pe 1, în caz contrar b_k este pus (înapoi) pe zero. Astfel, tensiunea de comparare U_{ck} se formează pe baza bitului b_k a cărei valoare urmează a fi stabilită și a biților b_i , $i = 1, 2, \dots, k-1$, cu valori stabilite pe baza unor comparații anterioare.

8. Etalonarea convertorului analog numeric - etalonarea decalajului și câștigului pentru un convertor bipolar pe N biți, rezultat sub formă de cod binar deplasat; tensiuni aduse la intrarea CAN, rezultate obținute la ieșirea CAN.

Pentru reglajul decalajului se aduce la intrarea convertorului tensiunea de $0,5LSB$ iar la ieșire trebuie să obținem secvențe numerice ce prezintă o instabilitate între valorile $10 \dots 00$ (codul pentru 0 – bitul de semnificație maximă pe 1, restul pe 0) și $10 \dots 01$. Pentru reglajul câștigului se aduce la intrarea convertorului tensiunea de $0,5FSR - 1,5LSB$ iar la ieșire trebuie să obținem secvențe numerice ce prezintă o instabilitate între valorile $11 \dots 10$ și $11 \dots 11$ (toți biții pe 1). Se are în vedere caracteristica de transfer a unui CAN bipolar, în CBD.



9. Circuit de eșantionare și memorare – structura de principiu a unui CEM cu buclă de reacție globală (un comutator), rolul reacției globale, caracteristici ale AO de intrare și de ieșire.

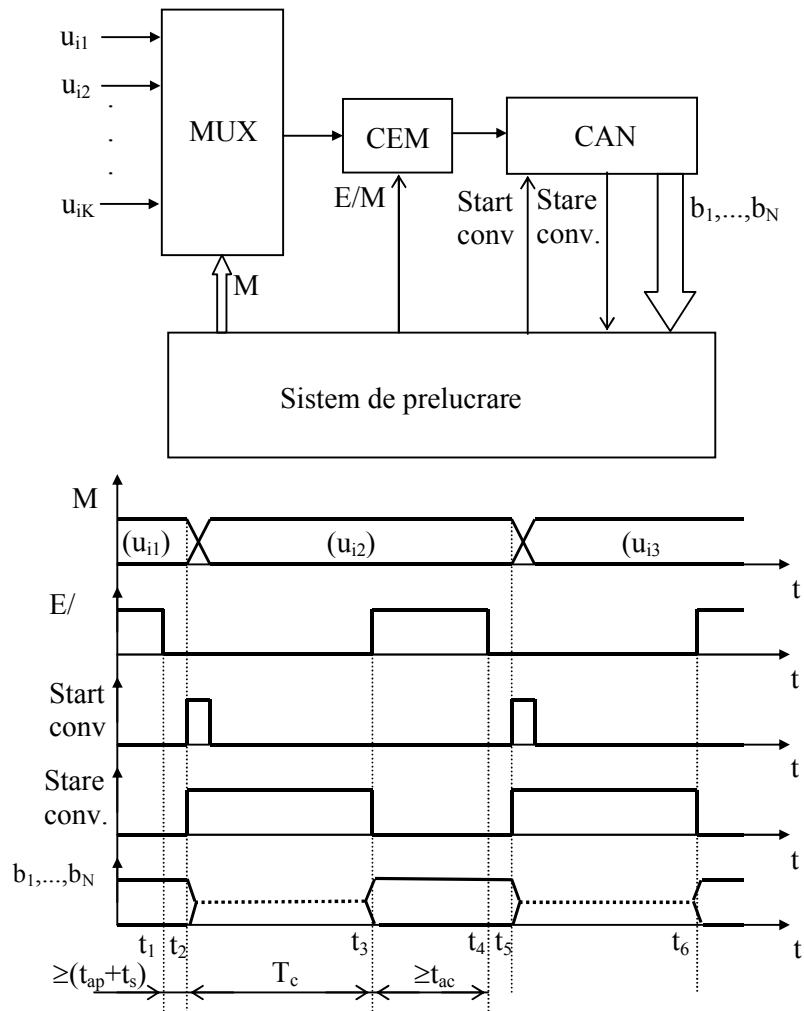


Reacția globală asigură eliminarea erorilor de decalaj corespunzătoare amplificatorului operațional de la ieșire AO2. În cazul structurii de principiu din figură, erorile de decalaj ale CEM sunt date de AO1, care trebuie ales cu derivă redusă a tensiunii de decalaj.

Amplificatorul operațional AO1 din circuitul de intrare al CEM trebuie să asigure o impedanță mare de intrare pentru a nu se perturba circuitul din care se prelevează tensiunea u_i . De asemenea, trebuie să aibe o impedanță mică de ieșire pentru a putea încărca sau descărca rapid condensatorul C astfel încât timpul de achiziție să fie cât mai mic.

Amplificatorul operațional AO2 de la ieșirea CEM trebuie să fie ales astfel încât curenții de polarizare a intrărilor să fie foarte mici pentru ca, în starea de memorare, descărcarea condensatorului C să fie foarte lentă. Din acest motiv se utilizează amplificatoare operaționale ce au în circuitul de intrare tranzistoare cu efect de câmp.

10. Sistem de achiziție de date cu multiplexarea semnalelor analogice de intrare – structura de principiu, explicarea funcționării pe bază de diagrame de timp ale semnalelor de comandă și control.



Bibliografie

- [1] Liviu Toma – Sisteme de achiziție și prelucrare numerică a semnalelor, Editura de Vest, Timișoara, 1997
- [2] <https://intranet.etc.upt.ro/> An 3, Sisteme de achiziții de date, Curs
- [3] <https://intranet.etc.upt.ro/> An 3, Sisteme de achiziții de date, Laborator

Testarea echipamentelor pentru EA

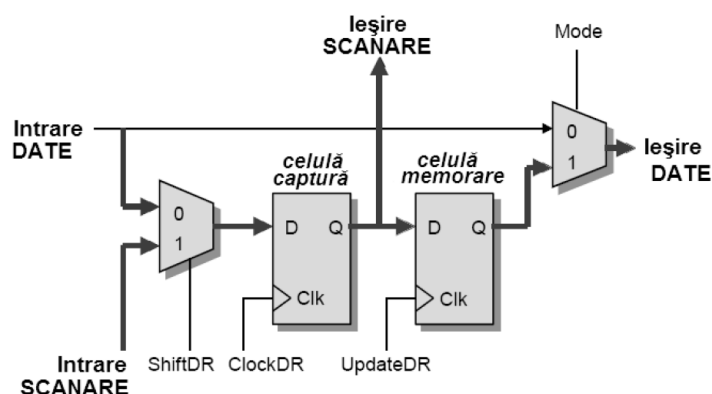
Anul IV

1. Celula de scanare pe frontieră: Arhitectura, semnale, moduri de operare.

Cursul 3 paginile 9-11

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

Celula de scanare – arhitectura, semnale



- **Intrarea de DATE** se conectează la terminalul circuitului integrat iar **ieșirea de DATE** la intrarea/ieșirea blocului funcțional, sau invers, după cum celula de scanare este conectată la un terminal de intrare sau de ieșire.

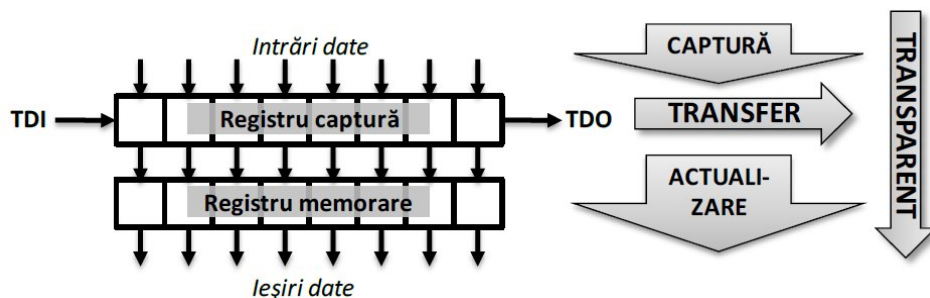
- **Intrarea de SCANARE** se conectează la ieșirea celulei anterioare (sau la terminalul TDI) iar **ieșirea de SCANARE** la intrarea următoarei celule (sau la multiplexorul pinului TDO), formând un registru de deplasare cu încărcare paralelă.

- Semnalele **ShiftDR**, **ClockDR**, **UpdateDR** și **Mode** sunt generate de către controlerul TAP care gestionează transferul datelor la nivelul fiecărei celule.

Celula de scanare – moduri de operare

Mod	Mode	ShiftDR	ClockDR	UpdateDR	Descriere
Transparent	0	x	x	x	Datele se transferă de la intrare la ieșire; circuitul funcționează normal, ca și când celulele de scanare ar fi inexistente
Captură	1	0	↑	0	Conținutul intrării de date se transferă în celula de captură
Actualizare	1	x	0	↑	Conținutul celulei de captură se transferă la ieșire și se memorează în celula de memorare
Transfer	1	1	↑	0	Datele se transferă de la intrarea de scanare la ieșirea de scanare, prin celula de captură. Acest mod nu afectează informația stocată în celula de memorare

- Modul **captură**: intrările de date sunt transferate în registrul de captură
- Modul **transfer**: se realizează deplasarea serială a datelor prin registru de captură
- Modul **actualizare**: registrul de captură se transferă în registrul de memorare și la ieșirile de date.
- Modul **transparent**: intrările de date sunt conectate direct la ieșirile de date, registrele de captură și de memorare sunt invizibile, fără a fi afectat conținutul lor.



2. Tehnici de inspecție optică automată: enumerare, scurtă caracterizare

Cursul 5 paginile 8-10

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau

<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

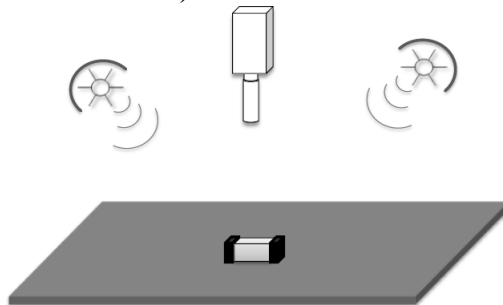
- Metoda referinței (Template Matching):

- Prin compararea cu o imagine de referință se decide dacă produsul inspectat corespunde sau nu specificațiilor.
- Imaginea de referință poate fi obținută fie în urma unei reprezentări CAD fie prin memorarea imaginii unei plăci de referință care în mod cert este conformă specificațiilor (Golden Board).
- Principalul dezavantaj: admite o toleranță redusă a procesului tehnologic, abateri minore de la referință, ce nu au impact asupra calității sau fiabilității sistemului inspectat, declanșând adeseori alarme false.

- Metoda comparației statistice (Statistical Pattern Matching):

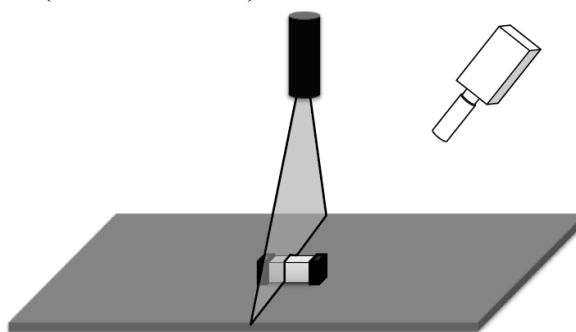
- reprezintă o îmbunătățire a metodei referinței în scopul de a atribui mai multă flexibilitate în luarea automată a deciziei CONFORM/NECONFORM.
- se utilizează mai multe imagini de referință obținute în urma inspecției unor produse atât conforme cât și neconforme. Decizia se ia pe criterii statistice în urma analizei similarității sistemului inspectat cu cele două categorii de referință.

- Măsurarea bidimensională (2D Measurement):



- utilizează una sau mai multe camere montate perpendicular pe planul de inspecție care este iluminat din multiple unghiuri.
- Imaginea obținută permite efectuarea de măsurători pe direcțiile X și Y, în urma unui proces de calibrare.
- Metoda se folosește pentru inspecția plăcilor neechipate în scopul detecției întreruperilor și scurtcircuitelor, pentru a verifica poziționarea componentelor și alinierea pastei de lipire pe padurile SMD.

- Măsurarea tridimensională (3D Measurement):



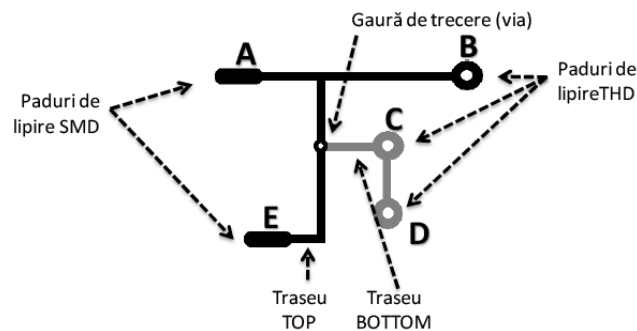
- Utilizează o sursă de lumină structurată, de exemplu o diodă laser, pentru a proiecta un șablon, cel mai frecvent o linie orizontală, pe suprafața suprapusă inspecției. Lumina structurată proiectată pe sistemul testat urmărește secțiunea acestuia într-un plan perpendicular pe planul inspecției.
- Camera este poziționată oblic față de planul inspecției și captează conturul luminii structurate, a cărui analiză permite efectuarea de măsurători tridimensionale.
- Tehnica se folosește în principal pentru a determina volumul pastei de lipire depus pe padurile componentelor SMD.

3. Testarea continuității

Cursul 5 paginile 24-26

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

- Testele de continuitate au ca scop detecția defectelor de tip întrerupere și se aplică între extremitățile rețelelor conductoare ale circuitului imprimat.
- O **rețea conductoare** (*net*) reprezintă o structură metalică continuă formată din trasee și găuri de trecere (*via*), care asigură o cale de circulație a curentului între mai multe paduri de lipire a componentelor sau paduri de test.



- Testarea continuității unei rețele se realizează prin măsurarea rezistenței dintre un punct de extremitate al acesteia și toate celelalte și compararea acesteia cu un prag.
- Standardul IPC-ET-652 solicită utilizarea unui prag de continuitate de 50Ω pentru echipamentele electronice de uz general și 20Ω pentru echipamentele de mare fiabilitate, cele de uz medical sau aeronautic.

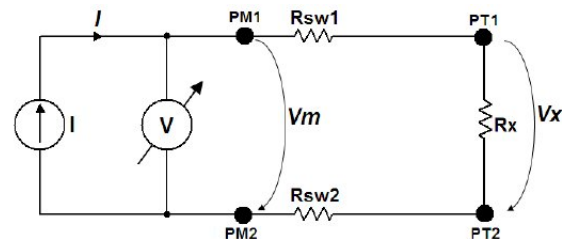
Tehnici pentru măsurarea rezistențelor dintre punctele de test situate la extremitățile unei rețele conductoare:

A. Măsurarea în două puncte

Este tehnica uzuală de măsurare a rezistenței utilizând o sursă de curent și un voltmetru.

Sursa de curent injectează curentul I între punctele de test PT1 și PT2 iar un voltmetru măsoară căderea de tensiune pe rezistența R_x dintre acestea. Prin compararea acestei tensiuni cu un prag se ia decizia de conformitate sau neconformitate.

Datorită rezistențelor comutatoarelor situate între punctele de măsură și punctele de test, notate în figura de mai sus cu R_{sw1} și R_{sw2} , tensiunea măsurată va fi:



$$V_m = I \cdot (R_x + R_{sw1} + R_{sw2}) = V_x + V_{err}$$

adică tensiunea măsurată V_m conține atât tensiunea de interes V_x cât și o tensiune de eroare V_{err} . Rezistența măsurată R_m se obține prin raportarea V_m la I :

$$R_m = R_x + R_{sw1} + R_{sw2}$$

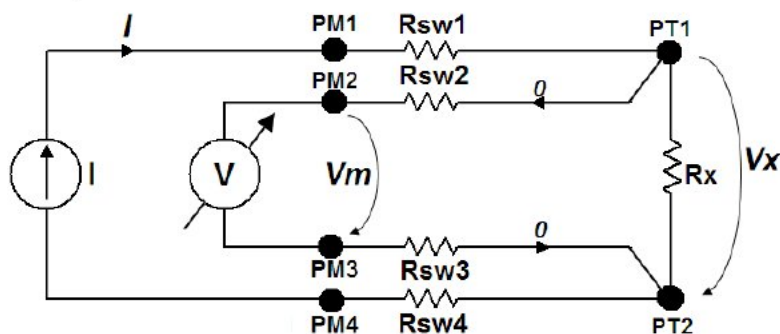
, ceea ce conduce la o eroare relativă de măsurare a rezistenței:

$$\varepsilon_{R_x} = \frac{R_m - R_x}{R_x} = \frac{R_{sw1} + R_{sw2}}{R_x}$$

Având în vedere că la testarea continuității rezistența R_x are valori de ordinul ohmilor chiar dacă se utilizează comutatoare cu rezistența în conducție mică eroarea de măsurare tot poate depăși cu ușurință pragul de 100%.

B. Măsurarea în patru puncte: utilizează o punte Kelvin formată din rezistențele în conducție ale comutatoarelor pentru a măsura rezistența de valoare mică dintre cele două puncte de test.

- Sursa de curent este conectată la punctele de test prin două puncte de măsură iar voltmetru prin alte două puncte de măsură.
- Se poate neglija curentul prin voltmetru pentru că impedanță de intrare a acestuia mult mai mare decât valorile de ordinul ohmilor pe care dorim să le măsurăm.
- Tensiunea măsurată este: $V_m = I \cdot R_x$
- Eroarea de măsură devine nulă indiferent de rezistențele comutatoarelor.
- Configurația de măsură elimină necesitatea calibrării și a compensării software însă utilizează de două ori mai multe puncte de măsură decât varianta prezentată anterior.



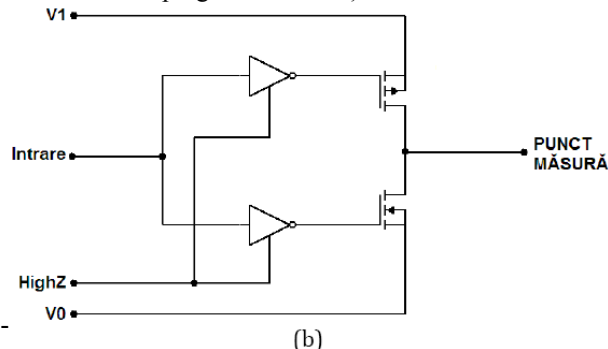
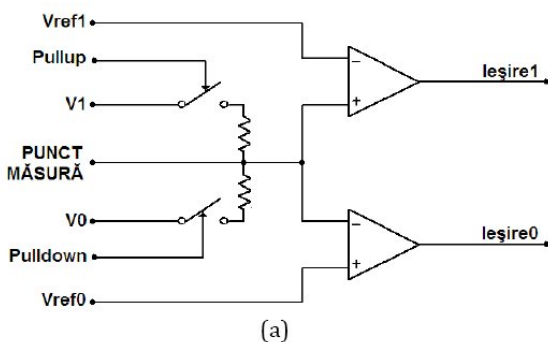
4. Resurse de testare digitală – Senzorul și Driverul: Scheme, funcționare

Cursul 6 paginile 16-17

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

Pentru testarea digitală se utilizează două tipuri de circuite: driver și senzor.

- **Driverul** (DD = *Digital Driver*) este un circuit care generează un stimul de test digital.
- **Senzorul** (DS = *Digital Sensor*) este un circuit care măsoară un semnal digital.
- Prin combinarea celor două se obține un circuit de test bidirecțional denumit **driver/senzor** (DDS = *Digital Driver /Sensor*).
- **Senzorul (a)** este un comparator dublu căruia îi sunt furnizate două praguri de referință independente, V_{ref1} și V_{ref0} , și care furnizează două ieșiri independente, una care semnalează starea logică „1” și una care semnalează starea logică „0”.
- Configurația permite nu doar detecția nivelurilor logice ci și detecția stării interzise, în care tensiunea de la intrare se află între cele două praguri de referință.



notată V_1 , sau tensiunea corespunzătoare nivelului „0”, notată V_0 .

- De obicei aceste două tensiuni sunt chiar tensiunile de alimentare ale circuitului supus testării, dar există și posibilitatea controlului prin software al acestora.

Driverul (b) dispune, pe lângă intrarea de date, de două intrări analogice V_1 și V_0 la care se stabilesc nivelurile de tensiune corespunzătoare nivelurilor logice „1”, respectiv „0”, precum și de o intrare de validare a comenzii tranzistoarelor finale, notată HighZ, prin intermediul căreia se poate aduce ieșirea în starea de înaltă impedanță.

Suplimentar, driverul poate avea circuite de monitorizare a curentului de ieșire și de protecție la scurtcircuit.

Driverul trebuie să aibă impedanță de ieșire redusă și să fie capabil să genereze un curent ridicat, uneori până la 500 mA, datorită fenomenului de **comandă inversă** (*backdriving*) inerent testării în circuit.

Pentru protecția circuitului supus testării împotriva efectelor distructive ale comenzii inverse, driverele performante sunt prevăzute cu circuite de control al timpului de creștere.

5. Structura unui stand de testare funcțională

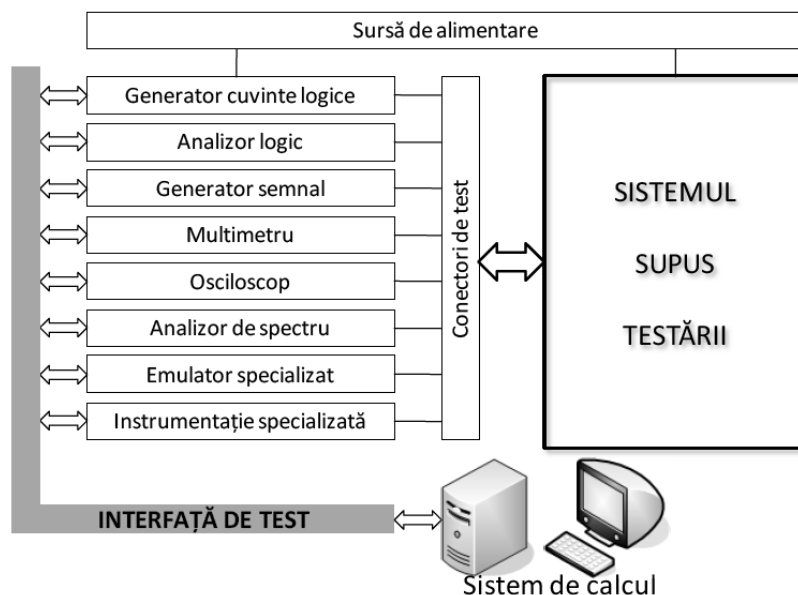
Cursul 7 paginile 22-24

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

Circuitele de complexitate redusă pot fi testate funcțional fără a necesita echipamente speciale; Pentru testarea sistemelor de complexitate redusă se pot utiliza testoare în circuit și monturi de test.

În acest scop a fost dezvoltată o categorie de testoare în circuit performante, dotate cu mai multe resurse de test decât cele prezentate anterior, denumite **testoare combinaționale (Combinational Tester)**.

- Sistemele complexe necesită însă echipamente de test dedicate, construite cel mai frecvent prin interconectarea mai multor instrumente de uz general și eventual a unor instrumente dedicate într-un **stand de test (test rack)**.



Structura generală a unui stand de testare funcțională, realizat prin interconectarea mai multor echipamente atât de uz general cât și specializate. Conceptul central al structurii este de **instrument modular controlat de calculator**

Din punct de vedere mecanic standul de test reprezintă un șasiu dotat cu un fund de sertar (*backplane*): o placă ce conține conectori de șasiu (*slot*) în care pot fi montate diverse module.

Fundul de sertar asigură conexiunea fizică între diferitele module ce pot implementa diferite funcții, inclusiv aceea de comunicație cu un sistem de calcul care poate controla în mod automat execuția testelor.

Prin standardizarea interfeței prin care comunică modulele, precum și a specificațiilor mecanice ale șasiului, devine posibil ca modulele se poată fi dezvoltate de diverși producători, oferind astfel o mare varietate de funcții de măsură.

6. Caracterizarea erorilor: Detectabilitatea și Redundanța – definiții

Cursul 8 paginile 16, 18

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

1. Detectabilitatea : Spunem despre un vector de test $X=[x_1, \dots, x_n]$ că detectează o eroare f dacă și numai dacă vectorul de răspuns conține cel puțin o ieșire z_k a cărei valoare este diferită în prezența, respectiv în absența erorii f :

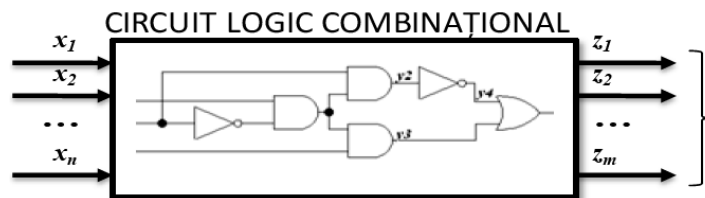
$$X \text{ detectează } f \Leftrightarrow \exists k = \overline{1, m} \text{ a. î. } z_k(X) \oplus z f_k(X) = 1$$

Cu $z_k(X)$ s-a notat funcția logică implementată de circuit la ieșirea z_k în funcționarea corectă, iar cu $z f_k(X)$ funcția logică implementată la aceeași ieșire în prezența erorii f .

Operația SAU-EXCLUSIV transpune în algebra binară cerința „valoare diferită”, (vezi tabelul de adevăr al funcției XOR.)

Spunem despre o eroare f că este detectabilă dacă și numai dacă există cel puțin un vector de test care o detectează:

$$f = \text{detectabilă} \Leftrightarrow \exists X \text{ a. î. } X \text{ detectează } f$$



2. **Redundanța:** Redundanța erorilor se definește ca opusul detectabilității:

$$f = \text{redundantă} \Leftrightarrow \exists X \text{ a.î. } X \text{ detectează } f$$

O eroare ce nu poate fi detectată de nici un vector de test este o eroare redundantă. Întrucât orice combinație binară aplicată intrărilor poate reprezenta un vector de test, rezultă că doar într-un circuit redundant pot exista erori astfel de erori.

Circuit redundant → poate fi eliminată cel puțin o poartă logică fără a-i modifica funcția logică implementată.

Redundanța poate apărea nu doar din cauza unei proiectări suboptimale a circuitului, ci și în urma aplicării unor tehnici de eliminare a hazardului combinațional sau pentru obținerea toleranței la defecțiuni.

Erorile redundante nu reprezintă o problemă în funcționarea circuitului (acesta implementează aceeași funcție în prezența și în absența unei astfel de erori), însă afectează testabilitatea datorită fenomenului de **mascare**.

O eroare redundantă **g** maschează o eroare **f** dacă **f** este detectabilă însă în prezența simultană a erorilor **f** și **g**, aceasta devine redundantă:

$$g \text{ maschează } f \Leftrightarrow \begin{cases} f = \text{eroare detectabilă} \\ g = \text{eroare redundantă} \\ f \text{ și } g = \text{eroare redundantă} \end{cases}$$

7. Generarea deterministă a vectorilor de test. Principiul

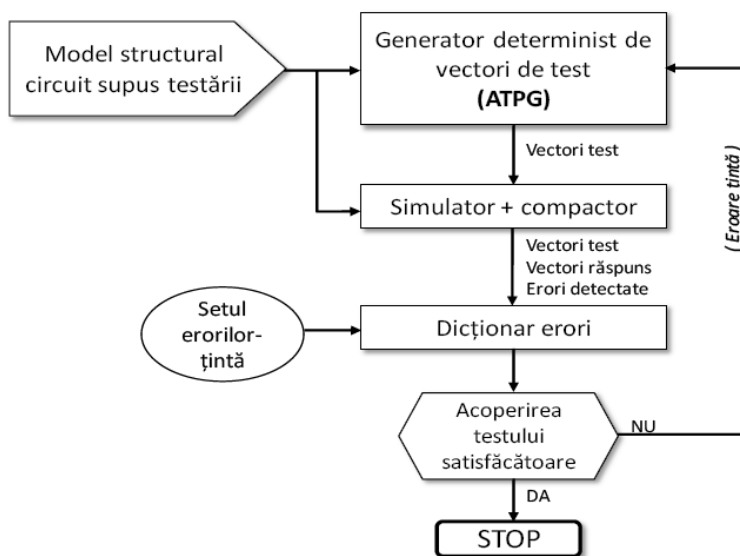
Cursul 9 pagina 17

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau

<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

Un **generator determinist de vectori de test** (ATPG = *Automatic Test Pattern Generator*) reprezintă un modul software care, în urma analizei unui model structural al unui circuit electronic, generează vectori pentru testarea acestuia. „**Analiza unui model structural**” este ceea ce conferă acestui tip de generator de vectori de test caracterul determinist, în opoziție cu generatoarele statistice care generează vectori aleatori în mod independent de structura circuitului supus testării.

- Pe baza modelului structural și eventual a unei erori țintă generatorul furnizează vectori de test unui modul de analiză a testului;
- Utilizând un simulator de erori și un algoritm de compactare a testului, acesta întreține un dicționar de erori ce conține vectori de test, vectori de răspuns și erorile detectate de fiecare dintre acestea.
- Dacă dicționarul de erori oferă o acoperire a testului satisfăcătoare, adică vectorii de test determinați pot detecta un procent minim impus dintre erorile țintă, atunci procesul de generare a testului este încheiat.
- Până la atingerea unei acoperiri a testului satisfăcătoare generatorul determinist este apelat în mod repetat, eventual cu specificarea unei erori țintă.



Utilizarea unui generator determinist de vectori de test, în conjuncție cu un simulator

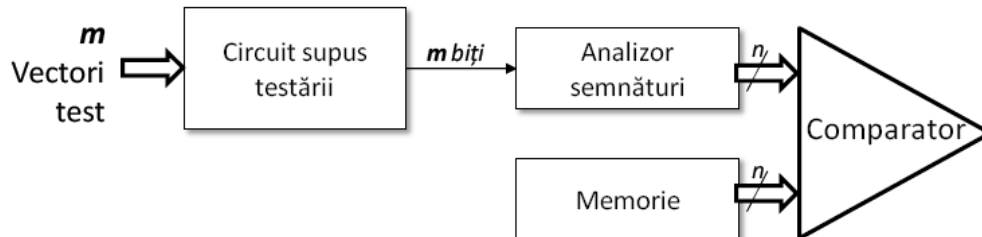
8. Principiul testării cu compresia datelor. Metode de compresie. Obiective.

Cursul 11 paginile 17-18

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

Principiul testării cu compresia datelor: în urma aplicării unei succesiuni de m vectori de test, la o ieșire a circuitului supus testării se obține o succesiune de m valori binare care poate fi caracterizată printr-o **semnătură** pe n biți. Calculând semnătura obținută în urma aplicării testului și comparând-o cu cea corespunzătoare funcționării corecte, stocată într-o memorie, se poate determina dacă circuitul este sau nu defect.

O semnătură caracterizează o secvență binară de lungime m , adică un șir de m biți care se succed în timp la intrarea analizorului. Semnătura se reprezintă pe un număr finit de n biți, de obicei 16, însă poate caracteriza o secvență binară de orice lungime, ceea ce o face atractivă pentru testarea circuitelor secvențiale complexe.



Metode de compresie:

- numărarea valorilor binare
- numărarea tranzițiilor
- determinarea parității
- testarea sindromului
- analiza de semnătură

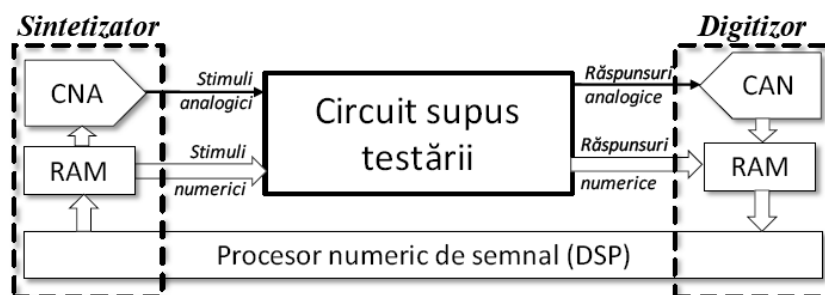
Obiective urmărite în implementarea metodelor de compresie:

- Metoda trebuie să poată fi implementată prin tehnici (circuite) cât mai simple.
- Tehnica utilizată nu trebuie să introducă întârzieri suplimentare în funcționarea circuitului sau să afecteze major timpul de test.
- Lungimea semnăturii trebuie să fie mult mai mică decât cea a răspunsului circuitului (comparația se face pentru o cantitate mult mai mică de informație). Această performanță este caracterizată de **gradul de compresie (GC)** definit ca raportul dintre lungimea secvenței de ieșire (răspunsul circuitului) și lungimea semnăturii.
- Metoda de compactare nu trebuie să piardă informația utilă din răspunsul circuitului, adică să nu mascheze manifestarea defectelor. Această performanță este caracterizată prin **probabilitatea de mascare (P)** a erorilor.

9. Testoare cu procesare numerică. Structura. Problema sincronizării.

Cursul 12 paginile 23-24

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau
<http://este2.etc.upt.ro/moodle/course/view.php?id=5>



Ansamblul DSP-RAM-CNA formează un **sintetizator**, capabil să genereze stimuli analogici: curenți, tensiuni, forme de undă, etc.

Ansamblul CAN-RAM-DSP formează un **digitizor**, capabil să achiziționeze și apoi să analizeze semnale analogice.

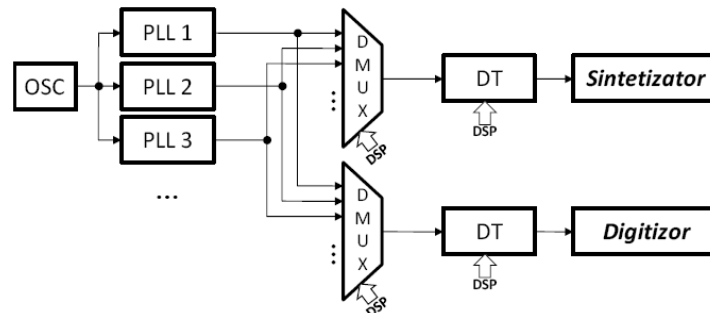
Sintetizatorul și digitizorul reprezintă emulatoare numerice ale unor instrumente analogice, care permit echipamentului de test să opereze cu vectori numerici în locul mărimilor analogice.

O caracteristică importantă a testoarelor cu procesare numerică este **sincronizarea** emulatoarelor pe baza unui tact unic.

Figura prezintă o structură tipică de distribuție a tactului într-un echipament pentru testarea circuitelor de semnal mixt.

Oscilatorul cu cuarț (OSC) produce tactul de frecvență medie (zeci de MHz), aplicat mai multor bucle cu calare pe fază (PLL = *Phase Locked Loop*) care generează multiple semnale de tact sincronizate, având diferite frecvențe, de ordinul

sutelor de megahertzi. Atât sintetizatoarele cât și digitizoarele pot opera cu oricare dintre aceste semnale de tact, a căror frecvență se poate reduce prin utilizarea unor divizoare de tact (DT). Factorii de multiplicare ai oscilatoarelor PLL și factorii de divizare ai blocurilor DT sunt aleși astfel încât între frecvențele cu care pot opera sintetizatoarele și digitizoarele să se stabilească rapoarte de numere întregi, de obicei prime între ele, necesare pentru eșantionarea coerentă. Sincronizarea în fază a tuturor semnalelor de tact și impunerea între acestea a unor rapoarte de numere întregi poartă denumirea de **sincronizare M/N**.



10. Testarea statică a CNA. Probleme. Masurarea indirectă

Cursul 13 paginile 7-9

https://intranet.etc.upt.ro/~TEE_EA/CURS/ sau

<http://este2.etc.upt.ro/moodle/course/view.php?id=5>

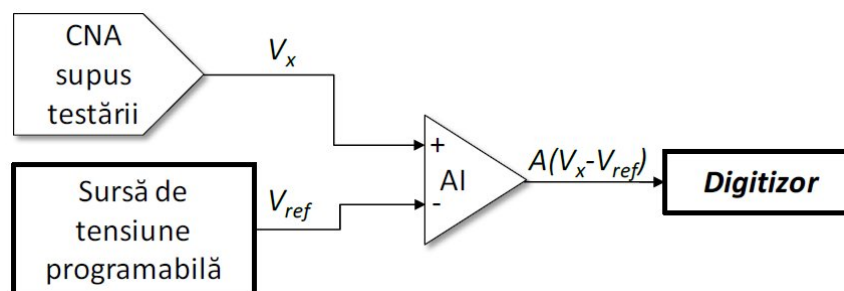
Testarea convertoarelor numeric-analogice pentru evaluarea caracteristicilor statice poate părea de domeniul evidenței: *se măsoară valorile de ieșire pentru fiecare cod aplicat la intrare și se compară cu valoarea ideală*. Există totuși două probleme:

(i) **Precizia digitizorului:** pentru testarea unui convertor pe n biți digitizorul trebuie să ofere o precizie de cel puțin $LSB/10$, ceea ce, ținând cont de zgomot și de erorile proprii digitizorului, solicită ca acesta să opereze pe cel puțin $n+4$ biți. Și atunci cum testăm un convertor pe 32 biți ?

(ii) **Numărul mare de coduri:** un convertor pe n biți necesită 2^n măsurători, adică 65.536 pentru convertoarele pe 16 biți și 16.777.216 pentru convertoarele pe 24 biți, ceea ce crește considerabil timpul de testare.

Soluția primei probleme este **măsurarea indirectă**, iar soluția celei de-a doua este **superpoziția erorilor**

Măsurarea indirectă se referă la evaluarea unei tensiuni prin măsurarea diferenței dintre aceasta și o tensiune cunoscută .



Programând sursa de tensiune pentru generarea unei valori V_{ref} apropiate de valoarea ce se dorește a fi măsurată (determinată de codul aplicat la intrarea CNA), se măsoară o diferență de tensiuni amplificată de un amplificator de instrumentație (AI). **Pentru măsurarea unor valori absolute ale V_x este necesar ca V_{ref} să aibă precizie ridicată, însă pentru măsurarea unor variații relative, cum e cazul neliniarității diferențiale, nu trebuie decât ca să fie stabilă în timp.**

Ex: măsurarea directă cu un digitizor ce operează pe 16 biți a tensiunii de la ieșirea unui convertor pe 12 biți, ambele având aceeași gamă de variație FSR . *Cuanta digitizorului va fi de 16 ori mai mică decât cuanta CNA, astfel încât, ignorând orice altă sursă de eroare, precizia de măsurare va fi $\pm 0.5 \cdot LSB_{digitizor} = 3,125\% LSB_{CNA}$.*

Să presupunem acum că se realizează măsurarea indirectă a erorilor de neliniaritate diferențială a CNA, iar gama de variație a digitizorului se stabilește la

$$FSR_{digitizor} = 4 \cdot LSB_{CNA}$$

Cuanta digitizorului va fi în acest caz de $2^{16}/4 = 16\ 384$ ori mai mică decât a CNA, astfel încât precizia de măsurare, considerând din nou doar eroarea de cuantizare a digitizorului, va fi de $0,003\% LSB_{CNA}$.

Pentru măsurarea DNL se vor face două măsurători consecutive:

$$V_{x1} - V_{ref} \text{ și } V_{x2} - V_{ref},$$

prin a căror diferențiere se obține rezultatul dorit fără a fi afectat de eroarea de programare a V_{ref} , cu singura condiție ca această tensiune să rămână constantă pe durata celor două măsurători.