

ETAPA DE ELABORARE 3

RAPORT PRIVIND ACTIVITATEA 2.3 / 2.4

Part 2.3 / Part 2.4

Implementarea software a algoritmilor și testarea acestora

- ☐ **Raport științific în extenso**
- ☐ **Proces verbal de avizare internă**

RAPORTUL ȘTIINȚIFIC ȘI TEHNIC ÎN EXTENSO

CUPRINS

1. Introducere.....	4
2. Arhitectura sistemului.....	4
3. Configurația host-target.....	5
4. Programul target.....	6
4.1. Interfațarea cu programul CCS.....	6
4.2. Conectarea cu procesorul de semnal TMS320C6211.....	7
5. Programul host.....	8
5.1. Analiza EEG ca proces variabil în timp.....	9
5.2. Predicția.....	10
5.3. Analiza timp-frecvență.....	17
Bibliografie.....	20
Anexa 1	21
Anexa 2	32
Anexa 3	34
Anexa 4	35
Anexa 5	37



MONITORIZAREA NIVELULUI DE OBOSEALĂ MENTALĂ
A PERSONALULUI CARE ACTIVEAZĂ INFRASTRUCTURA CRITICĂ

Colectiv de realizare :

Colectiv de realizare :

Echipa Coordonator – Universitatea Ovidius din Constanta

Echipa Partener – UTI GRUP SA

FUSURI DE SOMN

contact@fusuridesomn.ro; www.fusuridesomn.ro
0735 150 657 0722 604 071

CONTRIBUȚII LA PRELUCRAREA SEMNALELOR EEG PENTRU DETECȚIA UNDELOR TRANZITORII IMPLEMENTAREA ALGORITMILOR SOFTWARE ȘI TESTAREA ACESTORA

1 Introducere

În acest referat este prezentată realizarea tehnologică, pentru implementarea algoritmului de detecție a *complexului K* și *fusuri de somn*, cu ajutorul procesorului de semnal *TMS320C6211* produs de Texas Instruments. În *anexa 1* este prezentată interfața DSK C6211 (Digital Starter Kit).

2 Arhitectura sistemului

Această aplicație se bazează pe arhitectura sistemului, de tipul *host-target*, unde target-ul este DSK-ul (*Digital Starter Kit*) care realizează procesarea în timp real transmisă de la host-ul realizat software pe calculatorul de proces.

În *figura 1* este prezentată arhitectura sistemului.

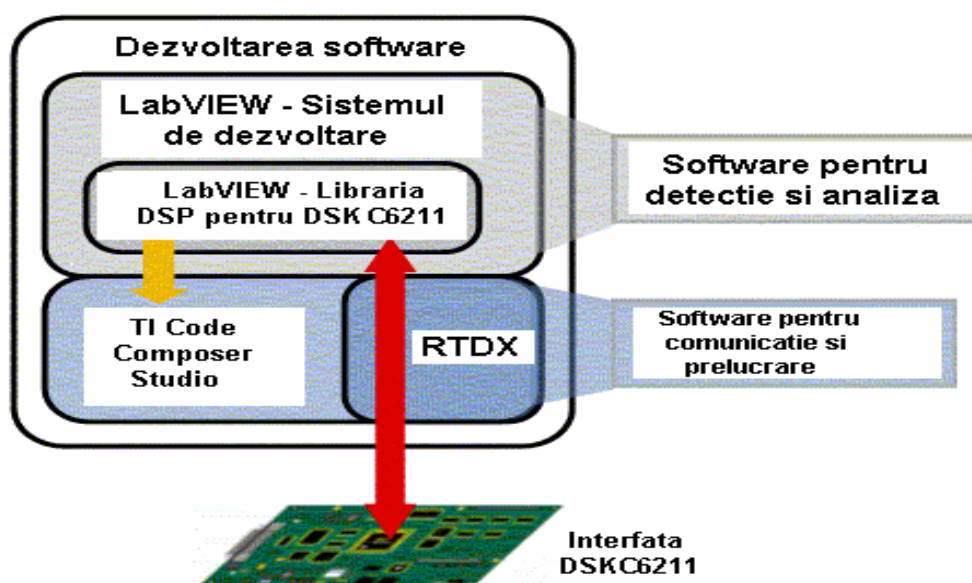


Figura 1. Arhitectura sistemului

Arhitectura sistemului utilizează protocolul RTDX (*Real Time Data Exchange*), dezvoltat de Texas Instruments, pentru realizarea schimbului de date bidirecțional cu procesorul de semnal TMS320C6211. Acest protocol este implementat în programul CCS (*Code Composer Studio*) și permite transferul de date în timp real, fără să fie influențat de întreruperile generate de program.

În colaborare cu Texas Instruments, firma National Instruments a realizat librăria *DSP Test Integration* pentru programul LabVIEW, realizând astfel interfațarea cu procesoarele de semnal din familiile TMS320C2000, C5000 și C6000, utilizând funcția RTDX.

Librăria DSP LabVIEW cu ajutorul căreia se realizează gestionarea funcțiilor CCS și comunicația RTDX este prezentată în *anexa 2*.

Aplicația transferă date pe portul paralel sau pe portul de mare viteză IEEE 1149.1 JTAG (*Joint Test Action Grup*), prin intermediul unei librării dinamice de alocare, *dsk 6211.dll*, realizată de fabricant.

3 Configurația Host – Target

Portul de interfațare cu un sistem gazdă (*Host Interface Port*) este un port paralel intrare / ieșire care permite conectarea la un procesor gazdă. Prin intermediul HIP, un TMS320C6211 poate fi văzut ca un periferic mapat în memoria gazdă. HIP operează paralel și asincron față de nucleul procesorului și memoria internă. HIP constă dintr-un set de regiștrii prin intermediul cărora se realizează comunicația cu procesorul gazdă. HIP poate fi configurat pentru magistrală pe 16 biți sau pe 32 biți, magistrală de date și adrese multiplexate sau separate, și semnale separate de citire și scriere sau semnal citire / scriere și semnale de date.

Configurația host – target este prezentată în *figura 2*.

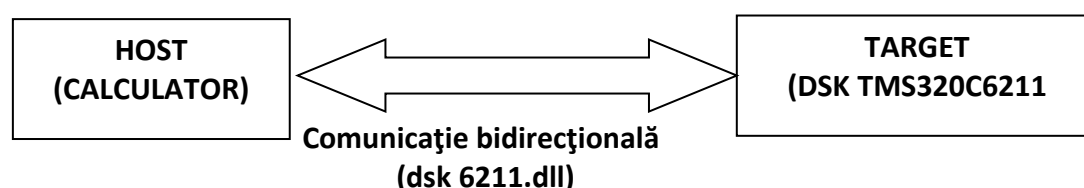


Figura 2. Configurația host – target.

Viteza de operare a HIP este similară cu cea a magistralei de date procesor. O operație de citire sau de scriere se poate efectua într-un singur ciclu. Portul de interfațare gazdă este complet asincron cu restul operațiilor procesorului de semnal. Gazda poate scrie sau citi date de la HIP în timp ce procesorul TMS320C6211 operează normal.

4 Programul target

Acest program are rolul de a stabili comunicația host – target, de realizare a schimbului de date cu ajutorul funcției RTDX, achiziționarea și filtrarea semnalului în timp real.

4.1 Interfațarea cu programul CCS

În cadrul librăriei LabVIEW DSP Test Integration au fost dezvoltate funcțiile de gestionare ale programului CCS (Code Composer Studio). Aceste funcții de gestionare CCS sunt: de deschidere, închidere, prelucrare și download a fișierelor care intră în compunerea proiectului dezvoltat în CCS.

Librăriile specificate anterior și arhitectura programului dezvoltat cu LabVIEW DSP de conectare cu CCS este prezentat în figura 3.

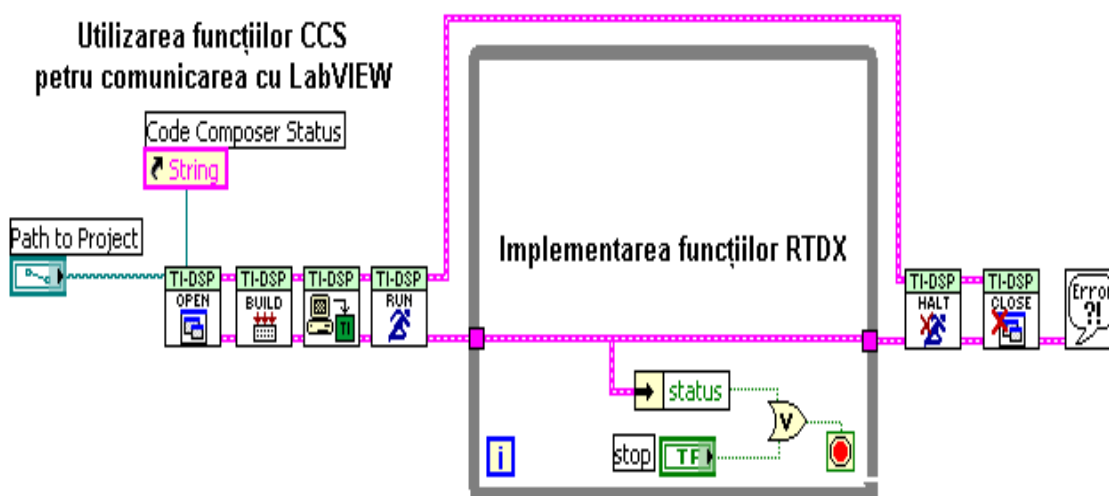


Figura 3. Arhitectura programului LabVIEW de comunicare cu CCS.

4.2 Conectarea cu procesorul de semnal TMS320C6211

Conectarea se realizează cu ajutorul funcțiilor RTDX. Aceste funcții permit transmiterea și recepția datelor spre și de la procesorul de semnal, de vizualizare și salvare a datelor, fără a opri aplicația. Înainte de a dezvolta secvențele de calcul în LabVIEW, este necesară definirea funcțiilor în CCS.

Implementarea funcțiilor RTDX pentru achiziția semnalului EEG în timp real și realizarea filtrării este prezentată în *figura 4*.

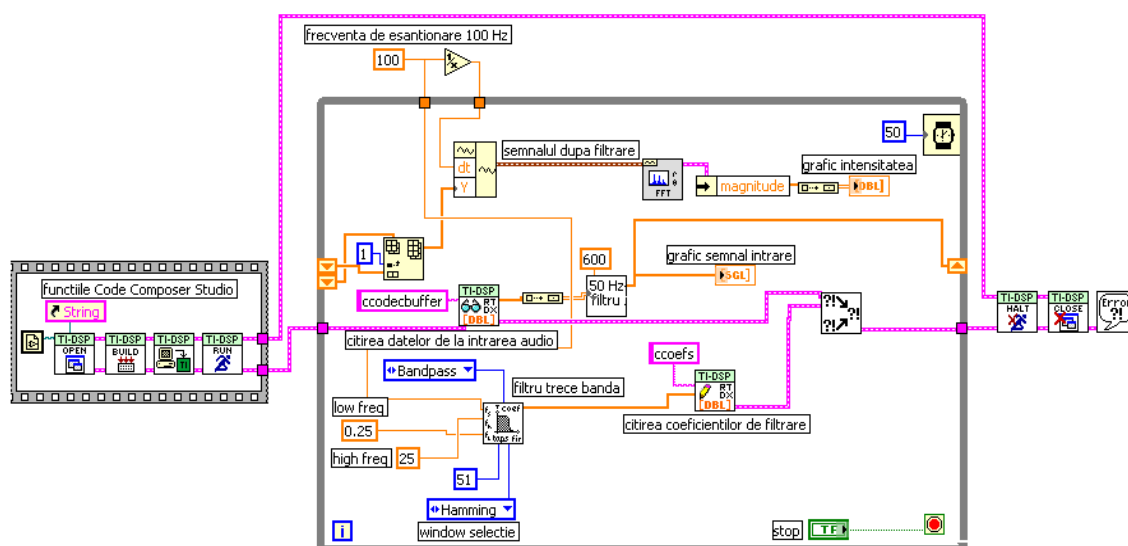


Figura 4. Arhitectura programului de achiziție și filtrare a semnalului EEG.

Pentru achiziția semnalului EEG care conține semnalele tranzitorii complex K și fusuri de somn este utilizat canalul F_z și C_z, conform sistemului internațional 10-20. Deoarece frecvențele de interes conținute în semnalul EEG sunt mai mici de 30 Hz, a fost aleasă o frecvență de eșantionare de 100 Hz pentru a respecta frecvența Nyquist. Înainte de aplicarea algoritmilor de detecție, semnalul a fost filtrat în două etape.

În prima etapă semnalul a fost filtrat cu un filtru rejector de bandă îngustă (filtru notch) pentru înlăturarea perturbațiilor de la rețeaua electrică. Pentru realizarea lui am ales o frecvență de eșantionare de 600 Hz și am dispus uniform 12 zerouri și 6 poli pe cercul unitate din planul z .

Această dispunere are ca efect anularea zerourilor ce corespund la frecvențele 0, 100, 200 Hz.

Funcția de transfer și ecuația cu diferențe finite ale filtrului notch sunt:

$$F(z) = \frac{1 - z^{-12}}{1 - z^{-6}}$$

$$y(n) = \frac{1}{2} [x(n) + y(n-6) - x(n-12)] \quad (1)$$

Diagrama de implementare a filtrului notch este prezentată în *anexa 3*.

Pentru a doua filtrare am ales un filtru trece bandă (FTB), pentru a realiza o izolare a componentelor de frecvență pe care le conține complexul K. Complexul K are două componente de frecvență, o componentă cuprinsă între 0.5 – 1.50 Hz și alta între 5 – 10 Hz. Din aceste considerente pentru FTB au fost alese următoarele frecvențe:

$$f_l = 0.25 \text{ Hz și } f_h = 25 \text{ Hz}$$

În *figura 5* sunt prezentate graficele semnalelor nefiltrate și filtrate utilizând cele două filtrări.

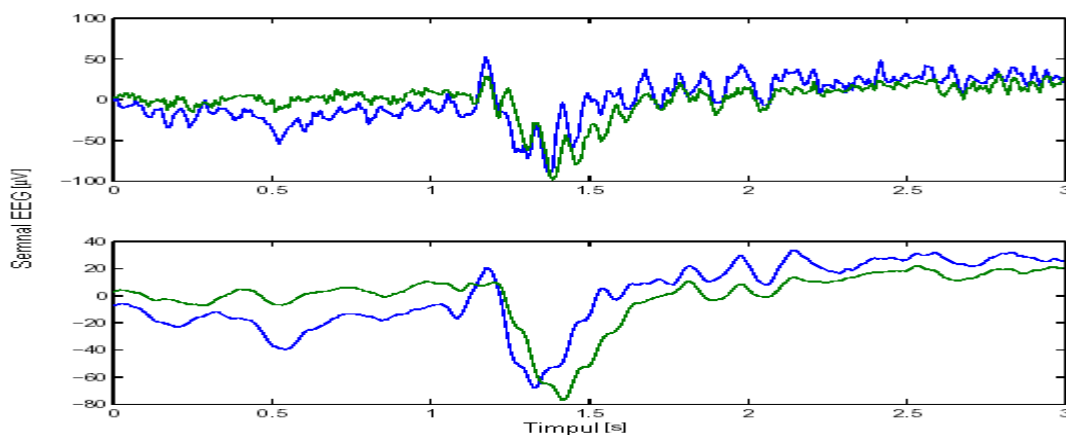


Figura 5. Semnal EEG care conține complex K, nefiltrat și filtrat.

5 Programul host

Programul host este realizat cu ajutorul limbajului de programare G, LabVIEW, și are ca obiectiv realizarea unui detector automat pentru complexul K, din electroencefalograma somnului.

Automatizarea detecției complexului K, trebuie să stabilească un *criteriu de verificare* și un *criteriu de predicție*.

5.1 Analiza EEG ca proces variabil în timp

Primul succes în analiza automată a EEG a fost introducerea algoritmului transformatei Fourier rapidă (FFT) în anul 1965, care este o dezvoltare a transformatei Fourier (FT).

Transformata Fourier îndeplinește criteriul predicției și determină o categorie de informații – distribuția spectrală a energiei semnalului. Cu toate acestea, FT poate duce la erori statistice și este influențată sever, datorită presupunerii că semnalul este ori infinit, ori periodic în afara ferestrei de măsurare. Metodele parametrice, cum ar fi modelul autoregresiv, nu prezintă efectul de “ferestruire”, și dau estimări cu proprietăți statistice mai bune, deoarece nu mai fac presupuneri asupra semnalului, în afara ferestrei de măsură. Totuși, analog ca în cazul FT, este necesar ca semnalul să fie staționar. Metodele spectrale, cum ar fi transformata Fourier și modelele autoregresive, au anumite limitări naturale. Ele oferă caracteristici globale ale întregului segment analizat, iar structurile de semnal cu durată mai scurtă decât fereastra de măsurare nu pot fi identificate. Conform informațiilor actuale, *informația prelucrată de creier este codificată de schimbările dinamice ale activității electrice în timp, frecvență și spațiu*. O descriere completă a acestui fenomen ar necesita o rezoluție timp-frecvență foarte bună, care nu poate fi obținută cu FFT sau modelul autoregresiv. Într-adevăr, există destule cazuri când sunt necesare caracteristicile globale ale întregului segment analizat. De asemenea, în unele cazuri, analizele timp-frecvență nu pot furniza tipul de informație obținută de modelul autoregresiv multicanal, de exemplu, direcția fluxului de informații între electrozi. Deoarece nici una din aproximările prezentate, nu satisfac toate cerințele, vom introduce în procesarea semnalelor biomedicale o nouă metodă – *criteriul de adaptare folosind algoritmi adaptivi*.

O îmbunătățire a rezoluției reprezentării timp-frecvență se poate obține dacă se utilizează filtre adaptive bazate pe metoda celor mai mici pătrate (*Recursive Least Squares*) și filtre cu estimarea erorii în sensul celor mai mici pătrate (*Least Mean Square*).

În acest scop realizăm un algoritm de predicție realizat cu ajutorul filtrului de netezire *Savitzky-Golai* și algoritmul adaptiv *LMS*.

Arhitectura programului de detecție a complexului k este prezentată în *figura 6*.

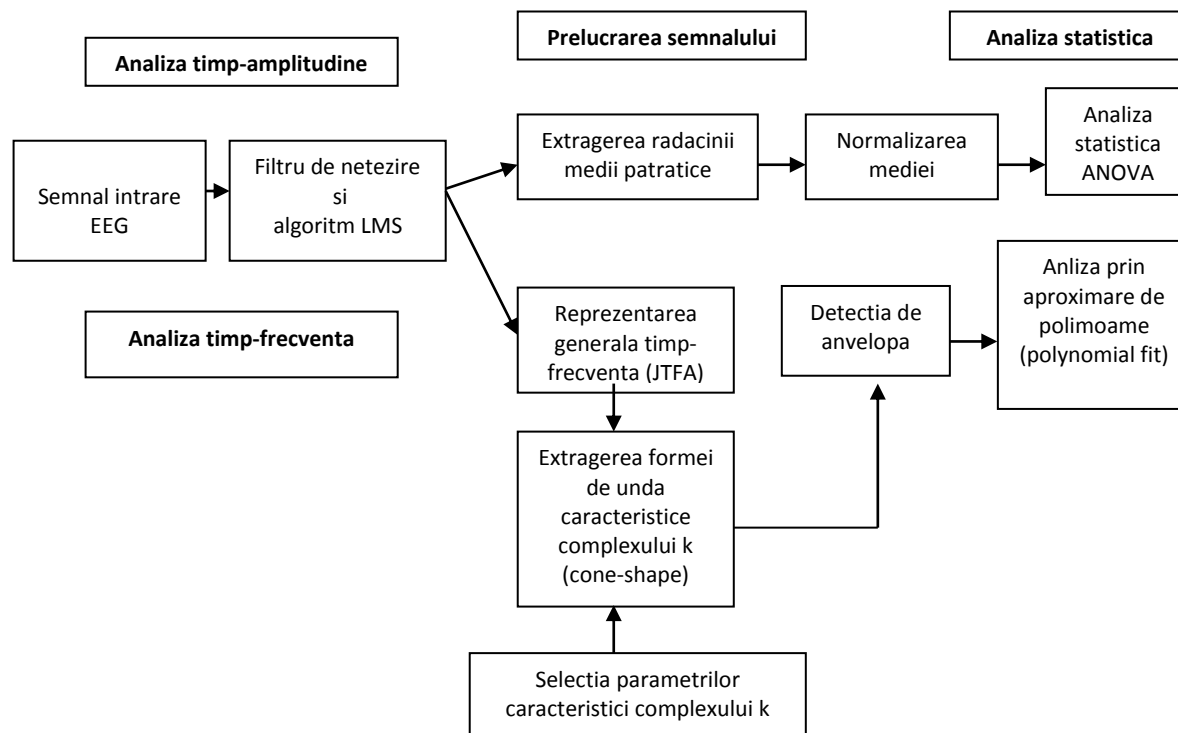


Figura 6. Arhitectura programului de detecție a complexului k și fusuri de somn.

Valorile obținute după realizarea predicției sunt analizate statistic (ANOVA) și corelate cu rezultatele statistice obținute prin analiza timp – frecvență a complexului k, utilizând librăriile JTFA (*Joint Time-Frequency Analysis*) din LabVIEW, urmată de extragerea anvelopei cu ajutorul unui filtru de ordonare statistică.

5.2 Predicția

Prin *definiție*, această operație *presupune o estimare a unei valori viitoare a semnalului aplicat*.

Rolul filtrului adaptiv este de a aproxima cât mai bine valoarea unui semnal la un moment dat pe baza unui număr finit de valori *anterioare* ale acestuia. Ideea fundamentală care justifică atingerea unui asemenea obiectiv constă în ipoteza, ca valorile succesive ale semnalului analizat respectă în mod obiectiv o dependență funcțională (în cazul cel mai simplu, liniară) dependentă de un număr limitat de parametri, ale căror valori pot fi estimate folosind un algoritm adaptiv adecvat.

În cazul liniar, modelele considerate se aleg de regula dintre următoarele 3 variante: autoregresiv (AR), cu medie alunecătoare (MA), respectiv combinația acestora (ARMA). Numărul de parametri care descriu modelul (și care definesc *ordinul* acestuia) se estimează folosind criterii statistice consacrate [Sil98]. În unele situații informația de ieșire este dependentă nu numai de valorile anterioare ale semnalului analizat, ci și de ale altor semnale. În plus, natura acestei dependențe poate varia în timp, sistemul adaptiv fiind forțat să asigure pe de o parte convergența rapidă a valorilor parametrilor și pe de altă parte urmărirea modificărilor apărute în procesul fizic analizat.

Schema predictorului este prezentată în figura 7.

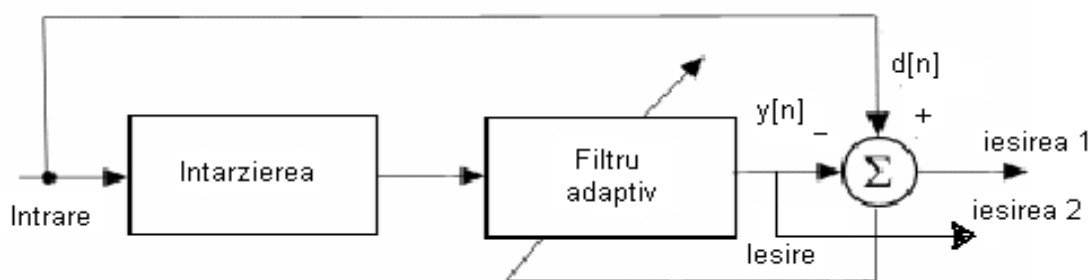


Figura 7. Predicția.

Predictorul ideal contravine principiului cauzalității, deci nu se poate realiza fizic. Pentru a evita acest obstacol, filtrului i se aplică secvența de intrare întârziată (x), iar ieșirea acestuia este comparată cu eșantionul prezent. Se poate utiliza fie *ieșirea 1*, realizându-se „*filtrul erorii de predicție*”, fie *ieșirea 2*, caz în care sistemul lucrează ca *predictor*. Eventual, un filtru având coeficienții identici cu cei al filtrului adaptat, poate fi utilizat pentru a filtra direct semnalul inițial.

Metoda de predicție liniară permite estimări foarte bune ale parametrilor semnalului, precum și posibilitatea de a obține viteze relativ mari de calcul. Analiza cu ajutorul predictorului se bazează pe faptul, că un eșantion de semnal poate fi aproximat ca o combinație liniară a eșantioanelor precedente. Minimizând suma diferențelor pătratice, pe un interval finit, între eșantioanele reale de semnal și cele obținute prin predicție liniară, se poate determina un set unic de coeficienți numiți coeficienți de predicție.

Ideea predicției liniare au fost folosite de multă vreme în teoria informației sub numele de *estimarea sistemelor și identificarea sistemelor*. Ultima denumire este ilustrată în ceea ce privește

metoda predicției liniare, în sensul că odată obținuți coeficienții de predicție, sistemul este unic identificat și poate fi modelat de un sistem liniar cu o structură care conține numai poli.

Din cauza naturii nestaționare a semnalului electroencefalografic, estimarea coeficienților de predicție se face pe segmente mici de timp, urmărindu-se minimizarea erorii medii pătratice de predicție.

Estimarea parametrilor modelului conform acestui principiu conduce la un set de ecuații liniare, ce pot fi rezolvate cu eficiență pentru obținerea coeficienților de predicție [Sil98], [Sil98].

Calculul câștigului

Pornind de la relațiile (6.2) și (6.3):

$$x(n) = \sum_{k=1}^P a_k x(n-k) + Gs(n) \quad (2)$$

$$H(z) = \frac{G}{A(z)} \quad (3)$$

unde: $A(z) = 1 - \sum_{k=1}^P \alpha_k z^{-k}$, se observă că este posibil să se stabilească o legătură între constanta de câștig, G , semnalul de excitație și eroarea de predicție. În cazul când $a_k = \alpha_k$, coeficienții predictorului real și cei ai modelului sunt identici: $e(n) = G s(n)$.

Aceasta înseamnă că semnalul de intrare este proporțional cu semnalul de eroare. Practic, se face presupunerea că energia semnalului eroare este egală cu aceea a semnalului de intrare:

$$G^2 \sum_{m=0}^{N-1} s^2(m) = \sum_{m=0}^{N-1} e^2(m) = \overline{e_n^2} \quad (4)$$

Trebuie menționat însă că pentru *complexul k* și fusuri de somn presupunând că $s(n) = \delta(n)$, se impune ca ordinul p al predictorului să fie suficient de mare pentru a lua în considerare toate efectele, eventuala apariție a *undelor delta*. În cazul semnalelor care nu conțin complexe K , semnalul $s(n)$ este presupus a fi zgomot alb gaussian, cu variație unitară și medie zero.

Pentru realizarea predictorului se utilizează *filtrul Savitzky – Golai*. Filtrul Savitzky – Golai, este un filtru de netezire în sensul celor mai mici pătrate folosind regresia liniară a unui polinom de grad k cu $(k+1)$ date, fiind folosite uzual în reducerea zgomotului și pentru evidențierea unei anumite frecvențe dintr-un semnal afectat de zgomot.

Pentru acest tip de aplicație filtrul de netezire Savitzky – Golai, execută o mediere mult mai bună decât filtrele FIR, care realizează o filtrare doar pe o porțiune a semnalului de frecvențe înalte afectat de zgomot.

În cazul predictorului, pentru estimarea coeficienților se utilizează algoritmul *LMS* (*Least Mean Square*).

Algoritmul LMS este cel mai important membru al familiei algoritmilor adaptivi de gradient stohastic pentru filtrele transversale. Prezentarea algoritmilor LMS este realizată în capitolul 6 din [Sil99].

Implementarea algoritmului LMS utilizând LabVIEW este prezentată în *anexa 5* [Dum].

Un element important al algoritmului LMS este alegerea constantei de adaptare μ . Alegerea lui μ se face luând în considerare cerințe contradictorii. Astfel, μ trebuie să fie mai mic pentru a atinge convergența mediei vectorului pondere și stabilitatea matricii de autocorelație a vectorului diferență. Pe de altă parte, este convenabil ca μ să fie ales cât mai mare, pentru a crește viteza de convergență.

Interpretând media vectorului pondere drept semnal de control:

$$c(k) = E\{\hat{w}(k)\} = (I - \mu R)^k E\{\hat{w}(0)\} \quad (5)$$

convergența algoritmului (în acest sens) este asigurată în măsura în care este îndeplinită condiția:

$$0 \leq \mu \leq \frac{2}{\lambda_{\max}}, \quad (6)$$

unde $\lambda_{\max}$ este valoarea proprie maximă pentru matricea R de autocorelație a semnalului de intrare.

Comportarea algoritmului LMS în medii nestaționare

De cele mai multe ori, în aplicații practice statistica mediului este nu numai necunoscută dar și variabilă în timp. Suprafața de cost este acum o suprafață de parametri variabili. De aceea filtrul adaptiv trebuie să-și găsească minimumul și apoi să fie capabil să-l urmărească.

Deoarece așa cum este arătat în [Sil99], convergența algoritmului LMS este asimptotică (implică un anumit timp), în mod intuitiv putem presupune o legătură strânsă între modul și viteza de variație a statisticii mediului de dezadaptare.

Astfel, pe lângă dezadaptarea datorată zgomotului de estimare M (noise misadjustment), apare un termen suplimentar, notat M_I și care reprezintă dezadaptarea datorată variației statisticilor mediului [Wid86]. Dezadaptare totală M_t devine: $M_t = M + M_I$.

Rezultatul final important este că, *în total, dezadaptarea (misadjustment) conține un termen proporțional cu μ datorat estimării gradientului, unul invers proporțional cu μ datorat variației statisticii mediului și unul independent de constantă de adaptare datorat faptului că $x(k)$ și $d(k)$ nu sunt albe.*

În final se pot formula câteva concluzii utile pentru utilizatorul unui sistem adaptiv bazat pe algoritmul LMS. *Algoritmul este relativ simplu de implementat, însă alegerea constantei de adaptare μ și a ordinului M pentru filtrul discret, necesită unele precauții. Ordinul filtrului și constanta de adaptare sunt singurele mărimi la îndemâna proiectantului prin intermediul cărora acesta poate controla atât stabilitatea, cât și viteza de convergență sau calitatea adaptării.*

Performanțele algoritmului depind în exclusivitate de constanta de adaptare μ . Pentru a asigura stabilitatea, μ trebuie să satisfacă în cazul general condiția:

$$0 \leq \mu \leq \frac{2}{\text{tr}(R)} = \frac{2}{\text{puterea totală a vectorului de intrare}} = \frac{2}{M\sigma_x^2} \quad (7)$$

O atenție deosebită trebuie acordată și statisticii semnalului de intrare. Este de dorit ca el să aibă caracteristici cât mai apropiate de zgomotul alb, astfel se înrăutățește considerabil viteza de convergență și dezadaptare [Dum].

Utilizând *metoda de detecție predictivă*, prin modificarea constantei de adaptare μ și a ordinului filtrului M , cu ajutorul algoritmului LMS, sunt obținute următoarele rezultate.

În *figurile 8 și 9* sunt prezentate semnalele predicționate în funcție de modificarea ordinului filtrului (M) și μ constant.

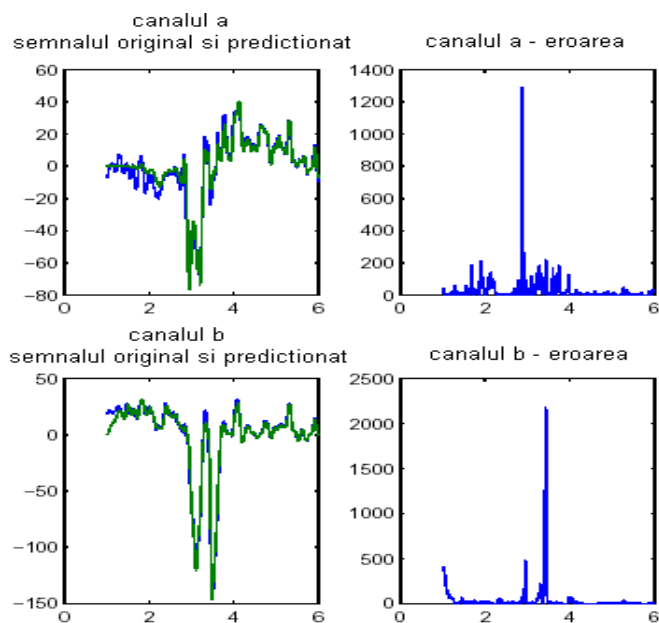


Figura 8. Complexul k original, predictionat și eroarea pentru μ constant și M variabil.

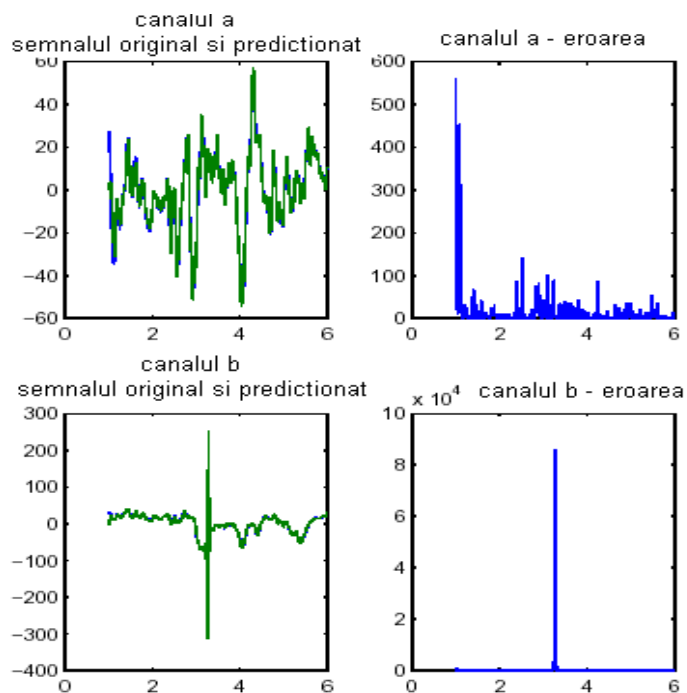


Figura 6.9. Complexul k original, predictionat și eroarea pentru valoarea maximă a lui μ (0.10) și M variabil.

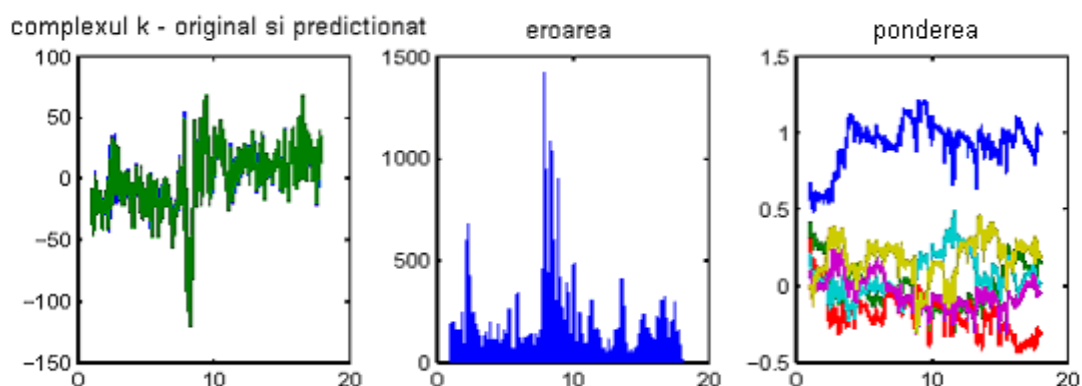


Figura 10. Complexul k – original și predicționat pentru μ variabil și M constant.

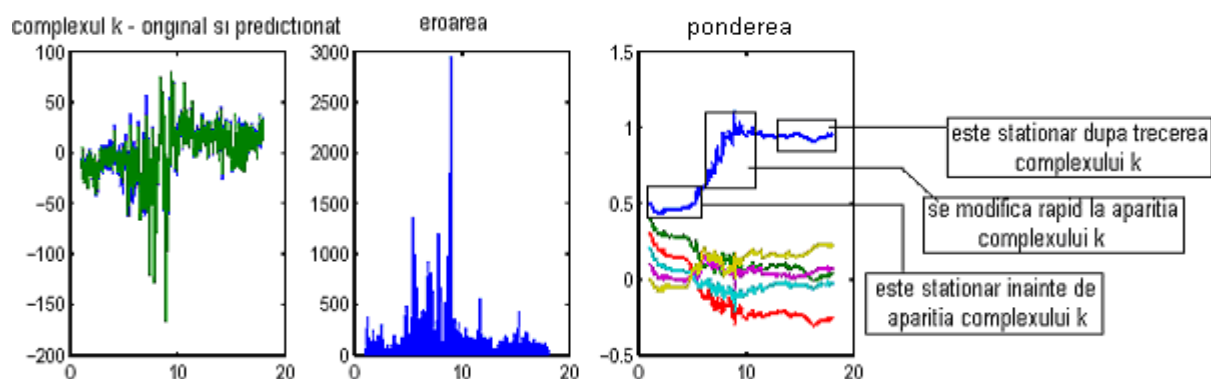


Figura 11. Variația ordinului M la apariția complexului k.

Se observă că ponderea este staționară înainte de apariția complex-ului K și fusuri de somn, se schimbă rapid în timpul apariției complex-ului K și devine din nou staționară după trecerea acestuia (fig. 11).

Utilizarea filtrării adaptive pentru predicția vârfurilor complexului k și fusuri de somn, a evidențiat existența unor praguri de detecție. Compararea cu detecția vizuală a ajuns la 80,1 % pentru detectarea complexului k și de 15,9 % pentru detectarea altor forme de undă, comparabile cu complexul k (undele delta). Pragurile de detecție au fost ajustate pentru a optimiza concordanța cu alegerea umană. În final am stabilit pragul de detecție cu amplitudinea $> 25 \mu V$ și lățimea în timp a ferestrei de analiză > 0.8 secunde. Astfel, se confirmă ipoteza formulată de Jankel și Niedermayer (1985) privind existența a două tipuri de vârfuri în timpul somnului: vârfuri lente, cu frecvența de 12

Hz, mai pronunțate în regiunea frontală, și vârfuri rapide, în jurul frecvenței de 14 Hz, localizate în regiunea parietală.

5.3 Analiza timp – frecvență

Pentru realizarea analizei timp – frecvență utilizăm librăriile JTFA din programul LabVIEW. Metoda JTFA implică selecția celei mai apropiate funcții nucleu care să corespundă formei fundamentale de undă care descrie complexul k și fusuri de somn. Astfel, noi trebuie să alegem forma nucleului bazându-ne pe valorile vârfurilor (localizarea) și pe amplitudinea unei funcții „martor”.

Rezoluția în domeniul frecvenței corespunzătoare analizei spectrului, care variază în timp, este egală cu frecvența Nyquist divizată cu 2^n ($n = 8$). Rezoluția în domeniul timp este 2^n ms ($n = 4$), fiind impusă de metoda aplicată.

Funcția nucleu este de forma:

$$C(t, w) = \frac{1}{4\pi^2} \iiint s^* \left(u - \frac{1}{2}u \right) s \left(u + \frac{1}{2}u \right) \rho(\theta, \tau) e^{-j\theta t - j\pi + j\theta t} du d\tau d\theta \quad (8)$$

unde $\rho(\theta, \tau)$ este o funcție bidimensională numită funcția nucleu [Shi96].

Selecția funcției nucleu pentru diferite distribuții timp – frecvență a EEG, a fost studiată de Devedeux și Duchene. Ei au definit funcția nucleu (ρ_c) astfel:

$$\begin{aligned} \rho_c(t, \tau) &= \xi(\tau) \text{ pentru } |\tau| \geq 2|t| \\ \rho_c(t, \tau) &= 0 \quad \text{în rest} \end{aligned} \quad (9)$$

Funcția nucleu stabilită de Zhao [Shi96], are expresia matematică:

$$\rho(\theta, \tau) = \xi(\tau) \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-j\theta t} dt = 2\xi(\tau) \frac{\sin\left(\frac{\theta\tau}{2}\right)}{\theta} \quad (10)$$

pentru care:

$$\xi(\tau) = \frac{1}{\tau} e^{-\alpha\tau^2}$$

$$\rho(\theta, \tau) = \frac{\sin\left(\frac{\theta\tau}{2}\right)}{\frac{\theta\tau}{2}} e^{-\alpha\tau^2} \quad (11)$$

Diagrama simplificată pentru analiza funcției nucleu din clasa lui Cohen este:

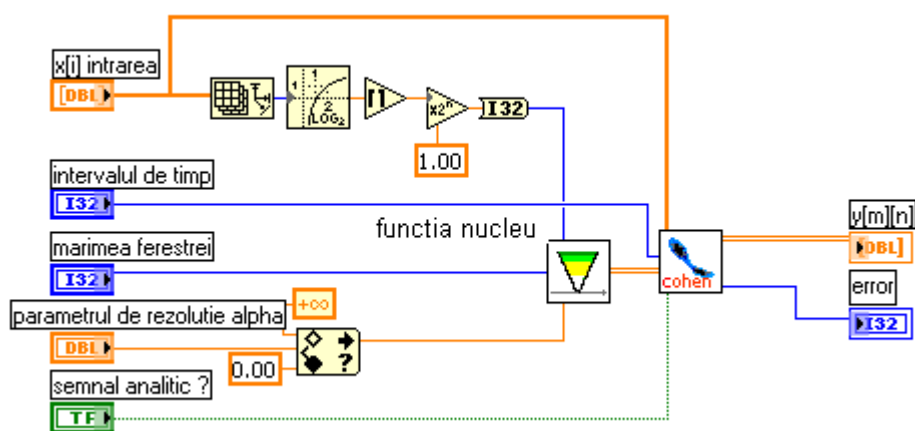


Figura 12. Diagrama simplificată a funcției nucleu (cone-shape kernel) și analiza din clasa Cohen.

Rezultatele analizei timp – frecvență cu ajutorul clasei Cohen, pentru detectarea complex-ului k și a undei delta sunt prezentate în *figura 13*.

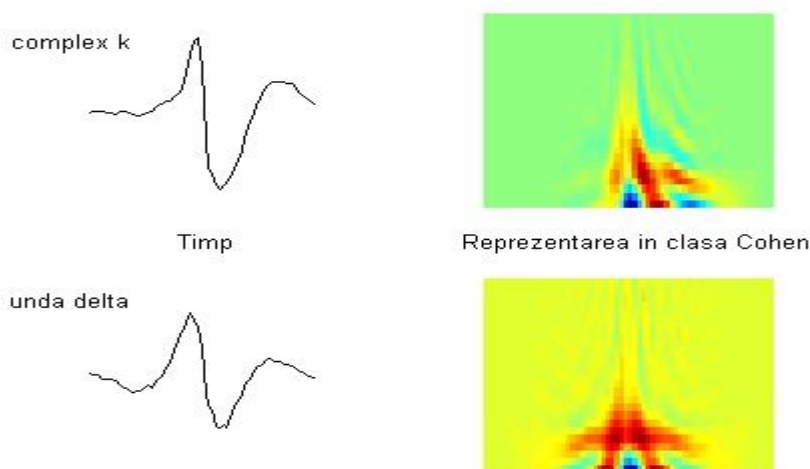


Figura 13. Analiza timp-frecvență pentru complex k și unde delta.

În *figura 14* este prezentată analiza comparativă timp-frecvență dintre stadiul 2 de somn (stadiu în care apare undele tranzitorii de tipul complex k și fusuri de somn), somnul paradoxal (Rapid Eye Movement) și stadiul de trezire. Din analiza rezultatelor prezentate în figurile 13 și 14, se observă *amprenta energetică caracteristică complex-ului k și fusuri de somn*.

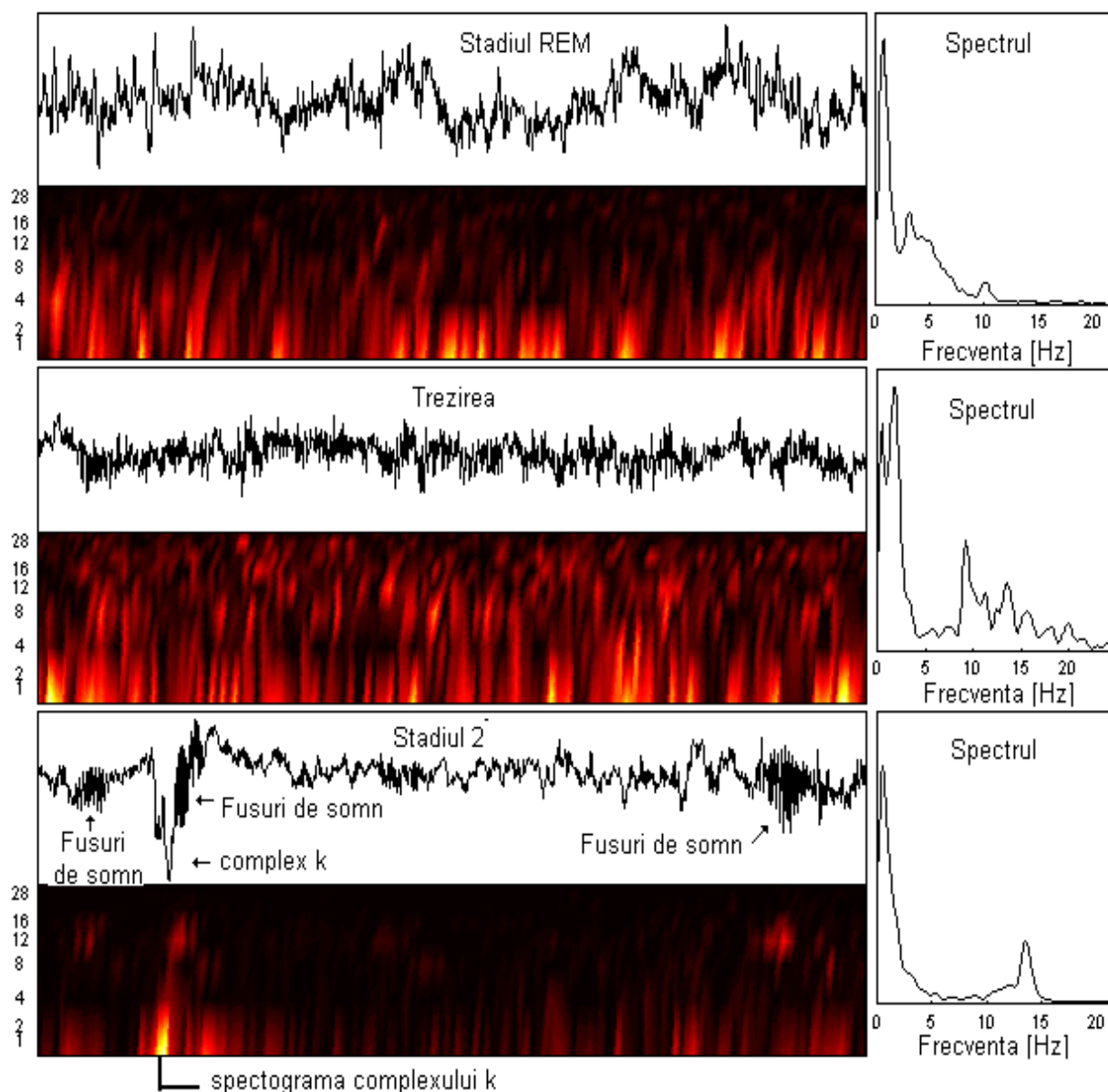


Figura 14. Spectograma rezultată în urma analiza timp-frecvență din clasa Cohen pentru diferite stadii ale somnului.

BIBLIOGRAFIE

TMS320C6000 Code Composer Studio Tutorial.

TMS320C6000 DSP/BIOS User's Guide.

TMS320C6000 Assembly Language Tools.

TMS320C6000 Peripherals Reference Guide.

TMS320C6000 CPU and Instruction Set Reference Guide.

TMS320C6000 Programmer's Guide.

TMS320C6000 Optimizing C Compiler User's Guide.

A. SAMANT, L. MAHESH. LabVIEW Signal Processing, *Prentice Hall*, 1998.

B. MIHURA. LabVIEW For Data Acquisition. *Prentice Hall*, 2001.

J. OLANSEN. Virtual Bio-Instrumentation Application in LabVIEW. *Prentice Hall*, 2002.

NATIONAL INSTRUMENTS. LabVIEW manuale de utilizare. *Nat. Instr. Press* 1997.

Resurse INTERNET.

Anexa 1

Descrierea procesorului pentru prelucrarea numerică a semnalelor TMS320C6211

1. Descrierea arhitecturii, funcționării și programării

1.1 Descriere generală

Familia TMS320 este o clasă de procesoare de semnal care înglobează o arhitectură de bază, comună, optimizată pentru procesarea digitală a semnalelor (DSP) și alte aplicații de procesare numerică de mare viteză. Varietatea de procesoare din această familie diferă în general prin tipul de periferice on-chip care sunt adăugate la arhitectura de bază. Memoria on-chip, un timer, porturi seriale și porturi paralele sunt prezente la diferiți membrii ai familiei. În plus, procesoarele TMS320 includ o interfață analogică pentru conversia semnalului vocal.

Din familia procesorului de semnal TMS320 face parte și procesorul TMS320C6211. Acesta este un procesor în virgulă fixă, cu viteză mare de calcul, suportă cuvinte pentru instrucțiuni foarte mari (VLIW), arhitectura este dezvoltată de Texas Instruments și are aplicabilitate în achiziții multicanal și multifuncționale.

1.2 Unitățile funcționale

Principalele unități funcționale ale arhitecturii procesorului TMS320C6211 sunt:

- *Unitățile de calcul* – Procesorul conține șase unități aritmetice de lucru (ALU) pe 32 – 40 biți, două unități multiplicator-acumulator pe 16 biți cu obținerea rezultatului pe 32 biți și un bloc de deplasare. Unitățile de calcul procesează 16-32 biți odată și de asemenea prezintă suport hardware pentru calculele în dublă precizie.

- *Generatorul de adrese și secvențiatorul de program* – Două generatoare de adrese dedicate și un secvențiator de program asigură adresarea memoriei on-chip sau a memoriei externe. Secvențiatorul suportă salturi condiționate executate într-un singur ciclu și execută bucla program fără utilizarea explicită a instrucțiunilor de salt pentru rularea în buclă. Cei doi generatori de adresă permit extragerea

simultană de către procesor a doi operanzi. Împreună, secvențiatorul de program și generatorii de adresă mențin unitățile de calcul într-o continuă funcționare, maximizând performanțele procesorului.

- *Memoria* – Procesorul folosește o arhitectură Harvard în care memoria de date păstrează datele și memoria program păstrează și date și cod program. Procesorul conține RAM on-chip (L1P, L1D) care reprezintă de fapt memoria program și memoria de date. Viteza de răspuns a memoriei on-chip permite procesorului să extragă doi operanzi (unul din memoria de date și unul din memoria program) și o instrucțiune (din memoria program) într-un singur ciclu.

- *Porturile seriale* - Porturile seriale (McBSPs) asigură o interfațare completă, serială cu comandare hardware pentru compresia și expandarea dinamică a datelor. Sunt suportate ambele tipuri de compandări μ -law și A-law.

McBSPs se interfațează ușor și direct cu o mare varietate de dispozitive seriale. Fiecare McBSPs poate genera un ceas intern programabil sau să accepte unul extern. McBSPs include o opțiune de comunicație multicanal.

- *Timer* – Două timere programabile cu un prescalar pe 8 biți permite generarea periodică a întreruperilor.

- *Portul de interfațare gazdă* – Portul de interfațare gazdă (HPI) permite conectarea directă (fără interfațare logică) la un procesor gazdă. HIP prezintă 16 pini de date și 11 pini de control. HIP este extrem de flexibil și permite o simplă interfațare cu o varietate de procesoare gazdă.

- *Porturi DMA* – Portul DMA (EDMA) permite un transfer eficient de date la și de la memoria internă. Portul DMA are magistrala de date și adrese multiplexată pe 16 biți și suportă 24 de biți pentru memoria program. Portul mai permite și încărcarea și memorarea instrucțiunilor și datelor.

Arhitectura familiei TMS320 prezintă un grad înalt de paralelism, adaptat la cerințele DSP.

Într-un singur ciclu, orice tip de procesor din această familie poate:

- Genera următoarea adresă program.
- Extrage următoarea instrucțiune.
- Execută unul sau două transferuri de date.
- Reactualizează una sau două adrese de date (operanzi).
- Execută o operație (aritmetică sau logică).

În același ciclu, procesorul poate:

- Recepționează și / sau transmite date prin intermediul portului serial.
- Recepționează și / sau transmite date prin intermediul portului HPI.
- Recepționează și / sau transmite date prin intermediul portului DMA.
- Recepționează și / sau transmite date prin intermediul interfeței analogice dacă aceasta este implementată pe placa de dezvoltare.

1.3 Interfața sistem și a memoriei

În procesorul TMS320C6211 există patru magistrale interne care leagă memoria internă cu celelalte unități funcționale: magistrala de adrese a memoriei de date, magistrala de date a memoriei de date, magistrala de adrese a memoriei program și magistrala de date a memoriei program. Aceste sunt multiplexate în afara chip-ului rezultând magistrala externă de adrese și magistrala externă de date, aceste magistrale pot fi folosite pentru accesare memoriei program și date.

Dispozitivele externe pot obține controlul asupra magistrelor procesorului cu ajutorul semnalelor de cerere și cedare a magistrelor. Procesorul poate continua să ruleze în timp ce magistralele sunt cedate altor dispozitive, până la apariția unei operații cu memoria externă.

Procesoarele TMS320 suportă periferice mapate în memorie cu posibilitatea generării programabile a stărilor de wait.

Ansamblul circuitelor de boot-are permite încărcarea automată a memoriei program interne, după reset. Acest lucru poate fi făcut fie prin intermediul interfeței cu o memorie EPROM, prin intermediul HPI de la un procesor gazdă sau prin portul DMA al TMS320.

Familia de procesoare TMS320 diferă prin structura vectorului său de întreruperi, dar în toate cazurile secvențiatorul de programe permite procesorului să răspundă cu o întârziere minimă. Întreruperile pot fi „nest-ate”, adică o întrerupere poate fi la rândul său întreruptă de o altă întrerupere cu prioritate mai mare.

1.4 Setul de instrucțiuni

Familia TMS320 înglobează un set unificat de instrucțiuni proiectat pentru a fi compatibil cu dispozitivele cu densitate mai mare de integrare. Setul de instrucțiuni al familiei TMS320 permite un transfer de date flexibil. Instrucțiunile multiple combină unul sau mai multe transferuri de date cu

excepția unor operații aritmetice sau logice. Fiecare instrucțiune poate fi executată într-un singur ciclu procesor. Limbajul de asamblare folosește o sintaxă algebrică pentru ușurarea înțelegerii codului. Un set facil de instrumente software și hardware permite implementarea și testarea programelor.

1.5 Performanțele DSP-ului

Aplicațiile de procesare a semnalelor necesită anumite performanțe care disting arhitectura DSP de arhitectura altor microprocesoare și microcontrolere. Nu doar instrucțiunile trebuie executate rapid, ci, DSP-urile trebuie să răspundă la fel de bine la următoarele cerințe:

- *Aritmetică rapidă și flexibilă* – Arhitectura de bază execută într-un singur ciclu înmulțiri, înmulțiri cu adunări, deplasări cu număr variabil de biți în ambele direcții, operații logice și aritmetice standard. În plus, unitățile aritmetice permit orice secvență de calcul așa că un algoritm DSP poate fi executat fără a fi reformulat.

- *Extinderea dinamică a domeniului* – Extinderea sumei de produse, comună în algoritmii DSP, este suportată în unitățile înmulțire- adunare ale TMS320C6211.

Un acumulator pe 40 biți prezintă 8 biți împotriva depășirii în adunări succesive pentru a se asigura împotriva pierderii de date, pot apărea până la 256 de depășiri. Instrucțiunile speciale permit lucrul cu blocuri de numere scalate.

- *Extragerea într-un ciclu a doi operanzi* – În calculul sumelor de produse, este nevoie de doi operanzi pentru a efectua calculul. TMS320 este capabil să extragă doi operanzi, indiferent dacă aceștia sunt în memoria internă sau cea externă.

- *Buffere circulare hardware* – Un număr mare de algoritmi DSP, incluzând și filtrele digitale necesită buffere circulare de date. Arhitectura de bază a TMS320 include structuri hardware pentru dirijarea pointer-ului de adrese în jurul vectorului, simplificând implementarea buffer-elor on sau off-chip și reducând nevoia scrierii de cod în plus (creșterea performanțelor).

- *Bucle dedicate și salturi* – Majoritatea algoritmilor DSP prezintă bucle. Secvențiatorul de programe suportă coduri de ciclare fără a fi nevoiți să scriem alte instrucțiuni de salt în plus, combinând creșterea performanțelor cu eficiența în înțelegerea codurilor program. De asemenea salturile condiționate se execută fără introducerea unor întârzieri suplimentare.

2. Structura internă a procesorului TMS320C6211

Arhitectura internă a procesorului este alcătuită din următoarele unități:

- *Unitățile de calcul* – Fiecare procesor conține 6 unități aritmetice de lucru (ALU), două multiplicatoare pe 16 biți și o unitate de deplasare. ALU execută un set standard de operații aritmetice și logice având în plus și instrucțiunile de împărțire. Multiplicatorul execută operații de înmulțire într-un singur ciclu, operații de înmulțire – adunare și înmulțire – scădere. Toate unitățile conțin regiștrii de intrare și ieșire care pot fi accesați prin magistrala de date a memoriei de date. În general operațiile de calcul își iau operanzii din regiștrii de intrare și încarcă rezultatul în regiștrii de ieșire. Regiștrii acționează ca un punct de trecere, pentru date, între memorie și unitățile de calcul.

- *Generatorii de adresă* – Generatoarele de adresă dedicate și secvențiatorul de program permit folosirea eficientă a unităților de calcul. Generatoarele de adresă asigură adresarea memoriei la transferul de date între memorie și regiștrii de intrare sau ieșire. Fiecare generator memorează până la patru pointere de adresă. Secvențiatorul de program asigură adresarea instrucțiunii din memoria program. El este condus de registrul instrucțiune care reține instrucțiunea curentă (aflată în execuție).

- *Magistralele* – Procesoarele au cinci magistrale interne. Magistralele de adrese a memoriei program și date sunt folosite intern pentru adresarea memoriei de date și program. Magistralele de date a memoriei de date și program sunt folosite pentru transferul de date asociate cu spațiul de memorie. Magistralele sunt multiplexate într-o singură magistrală de adrese externă și o singură magistrală externă de date.

- *Porturile seriale* – Procesorul are două porturi seriale bidirecționale, dublu buffer-ate, pentru comunicație serială. Porturile sunt sincrone și folosesc semnale de cadre pentru controlul traficului de date. Fiecare port poate genera intern propriul său semnal de ceas sau îl poate primi din afară, de la un ceas extern. Lungimea cuvântului poate varia de la 3 la 16 biți. Unul din porturile seriale are posibilitatea de lucru în mod multicanal și permite recepția sau transmiterea, selectabilă, prin 24 sau 32 de canale.

- *Timer* – Timer programabil, cu calare PLL, generator de întreruperi periodice. Un registru prescalar pe 8 biți, permite ceasului să decrementeze registrul numărător pe 16 biți odată la un ciclu sau 256 de cicli.

- *Porturi DMA* – Portul DMA asigură o comunicație eficientă între un sistem gazdă și un procesor. Portul este folosit pentru a accesa memoria program și memoria de date on-chip ale DSP, transferul fiind întârziat cu un singur ciclu/cuvânt

- *HPI* – Portul de interfațare cu un sistem gazdă, este un port paralel care permite conectarea facilă la un procesor gazdă. Prin intermediul HPI procesorul poate fi văzut ca un periferic mapat în memoria gazdei. HPI operează în paralel și asincron față de nucleul procesorului și memoria internă. HPI constă dintr-un set de regiștrii prin intermediul cărora se realizează comunicația cu procesorul gazdă.

- *EMIF* – suportă interfețe de extindere cu plăci externe, pe care le apelează (sincronizare de burst pentru SDRAM, acces sincron DRAM).

- *BOOT Configuration* – Poate suporta alta dispozitive externe de reset a plăcii cu DSP. Poate accesa plăci externe de inițializare, include coduri pentru ROM-uri externe, pentru EMIF și extensii de magistrale pentru HPI de la gazdele externe.

- *Interrupt Selector* – Are rolul de a produce sursele de întreruperi, selectează întreruperile și le ordonează după gradul de prioritate.

- *Power- down* – Are rolul de a coordona reducerea ceasului corelat cu reducerea puterii consumate (disipația termică).

În *figura 1* este arătată schema bloc a structurii interne a procesorului de semnal TMS320C6211.

3. Placa de dezvoltare DSK TMS320C6211 (Digital Starter Kit)

Cu ajutorul procesorului pentru prelucrarea numerică a semnalelor TMS320C6211, firma Texas Instruments a realizat **placa de dezvoltare DSK C6211**. Această placă de dezvoltare este folosită pentru achiziția semnalelor analogice, prelucrare digitală și stocarea lor sub un anumit format pe calculatorul de analiză și proces.

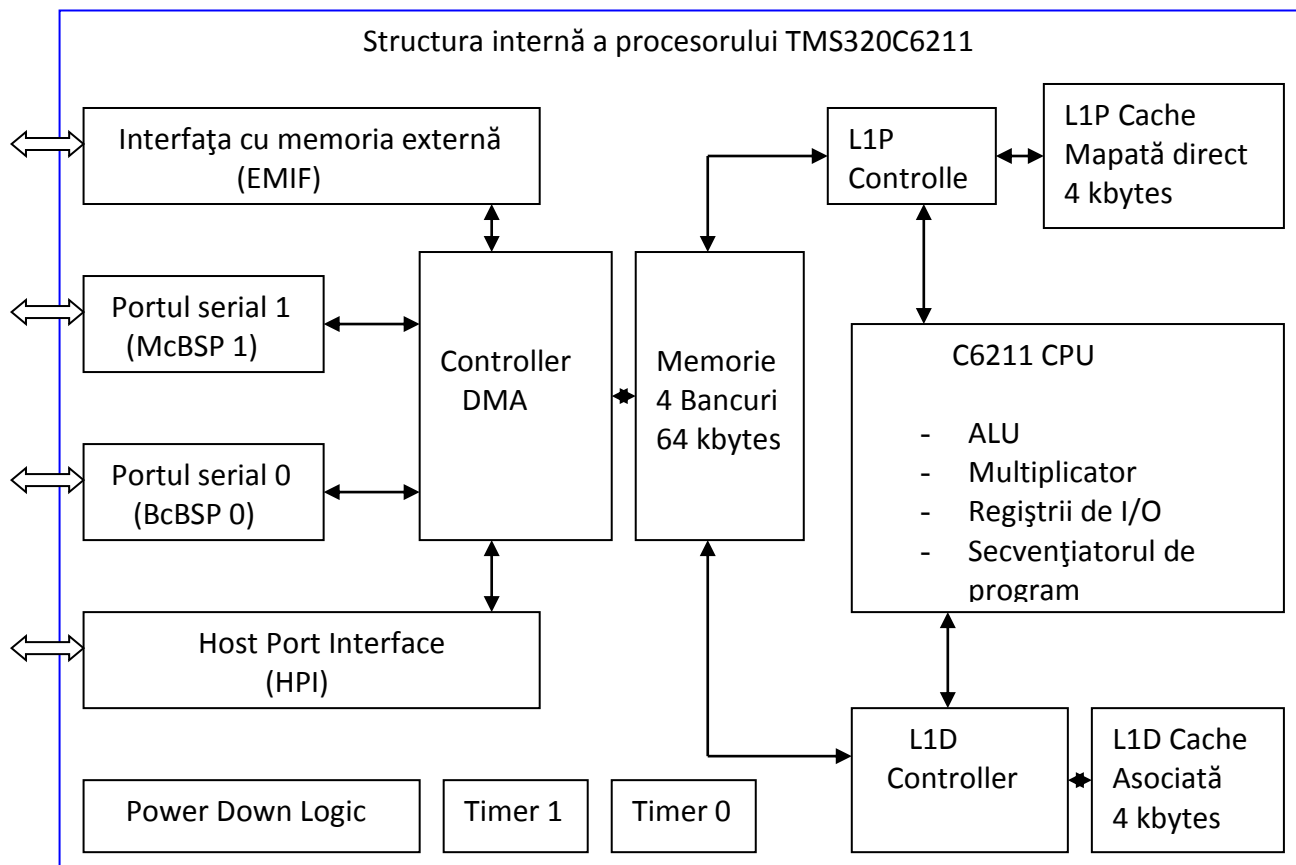


Figura 1. Structura internă a procesorului de semnal TMS320C6211.

3.1 Caracteristicile plăcii de dezvoltare DSK C6211

Caracteristicile plăcii de dezvoltare DSK C6211 sunt:

- conector pentru PC cu cablu paralel, viteză 150 MHz;
- SDRAM extern de 4 MB;
- flash ROM extern de 128 KB;
- convertor pe 16 biți tip TLC320AD535;
- controlul alimentării cu tensiunea de tipul TPS56100;
- controller pentru standard JTAG (acces multiplu de transfer);
- conectoare de extensie pentru interfețele aferente;
- intrare de semnal analogic (line level microphone);

- ieșire de semnal analogic (line level speaker) ;
- circuit de reset a procesorului și plăcii.

3.2 Descrierea interfeței analogice a plăcii DSK C6211

Procesorul TMS320C6211 include pe placa de dezvoltare DSK C6211 o interfață pentru semnal analogic constând dintr-un convertor A/D sigma-delta, un convertor D/A sigma-delta pe 16 biți și un set de regiștri de control și date mapați în memorie. Interfața analogică prezintă următoarele caracteristici:

- codec liniar sigma-delta ADC pe 16 biți ;
- codec liniar sigma-delta DAC pe 16 biți ;
- filtru anti-aliasing și anti-imaging on-chip;
- frecvența de eșantionare de 8 kHz;
- factor de amplificare programabil pentru DAC și ADC;
- referință de tensiune on-chip.

Interfața analogică este dedicată pentru lucrul cu semnal vocal în aplicații DSP. ADC și DAC lucrează la o frecvență fixă de 8 kHz. Includerea filtrelor anti-aliasing (*dacă viteza de eșantionare este prea mică, din datele achiziționate se va obține o formă de undă complet diferită și de frecvență mai mică. Acest efect este denumit **aliasing**. Dacă semnalul de măsurat conține componente cu frecvență mai mare decât jumătate din rata de eșantionare, se recomandă utilizarea unui filtru **anti-aliasing***) și anti-imaging, a convertoarelor sigma-delta pe 16 biți și a amplificatoarelor programabile, toate on-chip, prezintă o soluție de integrare înaltă pentru cerințele de procesare a semnalelor. Tehnologia de conversie sigma-delta elimină nevoia de filtre anti-aliasing off-chip și a circuitelor de eșantionare și reținere.

Interfața analogică operează folosind mai mulți regiștri de control și date mapați în memorie. Ieșirile și intrările ADC- ului și ale DAC- ului pot fi transmise și recepționate prin regiștri individuali mapați în memorie sau data poate fi autobuffer-ată direct în memoria de date a procesorului. Această autobuffer-are este similară cu cea a portului serial de la procesor .

Două întreruperi sunt dedicate convertoarelor ADC și DAC. O întrerupere este folosită pentru ADC, iar cealaltă pentru DAC. Întreruperile apar la rata de eșantionare sau când se execută o autobuffer-are complete.

3.3 Conversia A/D

Circuitul de conversie A/D a plăcii de dezvoltare DSK C6211 constă dintr-un multiplexor de intrare, un amplificator cu factor de amplificare programabil și un convertor analog-digital sigma-delta.

Intrarea analogică. Intrarea analogică este referențiată intern de către o referință de tensiune on-chip pentru a permite operarea cu o singură sursă de tensiune de +5V. Multiplexorul analogic selectează intrarea NORM sau AUX. El este configurat de către bitul 1 al registrului de control analogic al procesorului.

Amplificarea programabilă poate fi programată pentru creșterea nivelului semnalului cu +6dB, +20dB și +26dB. Nivelul semnalului de intrare la modulatorul sigma-delta nu trebuie să depășească V_{INMAX} valoarea specificată pentru procesor în catalog.

ADC. ADC-ul interfeței analogice constă dintr-un modulator analogic de ordinul patru sigma-delta, un filtru anti-aliasing cu decimare și un filtru digital trece-sus. Modulatorul sigma-delta elimină zgomotul din semnal și generează eșantioane de 1 bit la o frecvență de 1,0 MHz. Acest șir de biți, ce reprezintă de fapt semnalul analogic de la intrare, este băgat la intrarea filtrului anti-aliasing cu decimare.

Filtrul anti-aliasing cu decimare al ADC conține două nivele de procesare.

Primul este un filtru digital sinc⁴ care mărește rezoluția la 16 biți și reduce rata de eșantionare la 40 kHz. Al doilea nivel este un filtru trece-jos IIR.

Filtrul trece-jos IIR este un filtru eliptic de ordinul 10 cu nivelul benzii de trecere la 3,70 kHz și o atenuare a benzii de oprire de 65 dB la 4 kHz.

Acest filtru are următoarele specificații:

- tipul filtrului – filtru trece-jos eliptic IIR de ordinul 10.
- frecvența de eșantionare 40 kHz.
- frecvența de tăiere a benzii de trecere 3,7 kHz.

- riplul benzii de trecere $\pm 0,2$ dB.
- frecvența de tăiere a benzii de oprire 4 kHz
- riplul benzii de oprire -65 dB.

Filtrul trece-sus al ADC-ului extrage frecvențele joase din semnal, el atenuează din energia frecvențelor situate la baza benzii de trecere a convertorului.

Filtrul trece-sus este un filtru eliptic de ordinul 4 cu frecvența de tăiere a benzii de trecere la 150 Hz. Atenuarea benzii de oprire este de 25 dB. Acest filtru are următoarele specificații:

- tipul filtrului – filtru trece-sus eliptic IIR de ordinul 4.
- frecvența de eșantionare 8 kHz.
- frecvența de tăiere a benzii de trecere 150 kHz.
- riplul benzii de trecere $\pm 0,2$ dB.
- frecvența de tăiere a benzii de oprire 100 kHz
- riplul benzii de oprire -25 dB.

3.4 Conversia D/A

Circuitul de conversie D/A a interfeței analogice constă dintr-un convertor digital-analogic sigma-delta (DAC), un filtru analogic de netezire, un amplificator cu factor de amplificare programabil și un amplificator diferențial de ieșire.

DAC. DAC-ul interfeței analogice este implementat cu filtre digitale și un modulator sigma-delta ce au aceleași caracteristici ca și filtrele și modulatorul ADC-ului. DAC-ul este alcătuit dintr-un filtru trece-sus digital, un filtru anti-imaging cu interpolare și un modulator sigma-delta.

Filtrul trece-sus al DAC-ului are aceleași caracteristici ca și filtrul trece-sus al ADC-ului. Acesta extrage frecvențele joase din semnal, atenuează din energia frecvențelor situate la baza benzii de trecere a convertorului.

Filtrul de interpolare este un filtru anti-imaging care conține două nivele de procesare. Primul este un filtru trece-jos IIR care interpoalează rata datelor de la 8 kHz la 40 kHz și elimină erorile produse de acest proces. Rezultatul acestuia este din nou interpolat la 1 MHz și trecut printr-un filtru

digital sinc⁴ care atenează erorile produse de procesul de interpolare de la 40 kHz la 1 MHz. Acest filtru are aceleași caracteristici ca și filtrul trece-jos IIR de la ADC.

3.5 Circuitul de reset

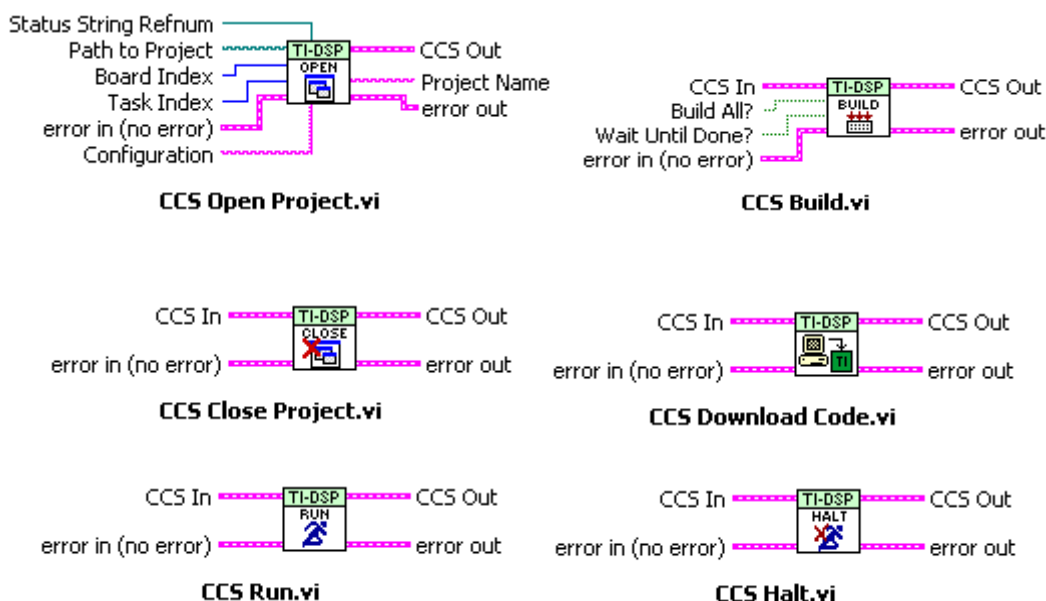
Butonul de reset oprește execuția programului și cauzează un reset hardware procesorului. Conținutul regiștrilor unităților de calcul și generatorul de adrese este nedefinit după reset.

Când semnalul reset este dezactivat, procesorul trece automat la operația de boot-are. Dacă procesorul este gazdă se poate iniția un reset sub control software.

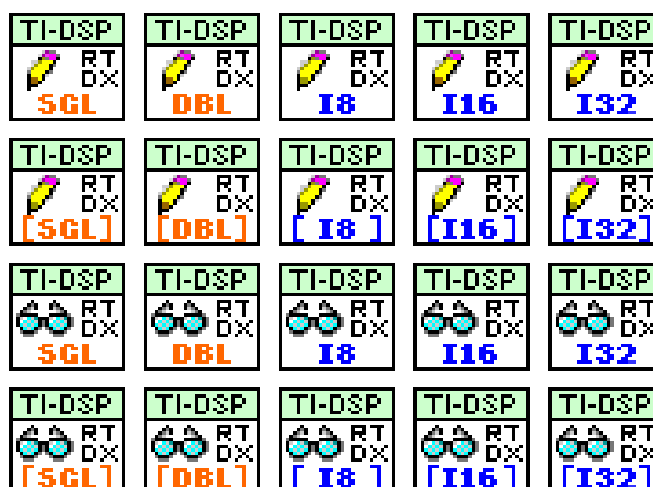
Anexa 2

Librăriile DSP și RTDX din LabVIEW – DSP Test System

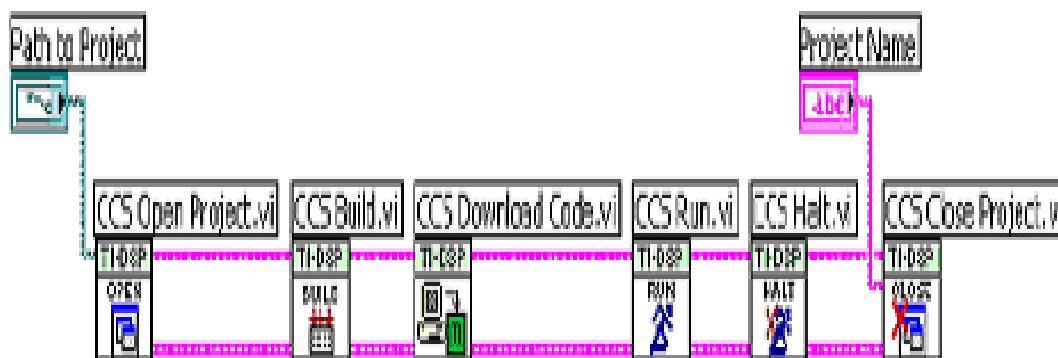
1. Funcțiile de bază ale Code Composer Studio implementate în DSP Test System



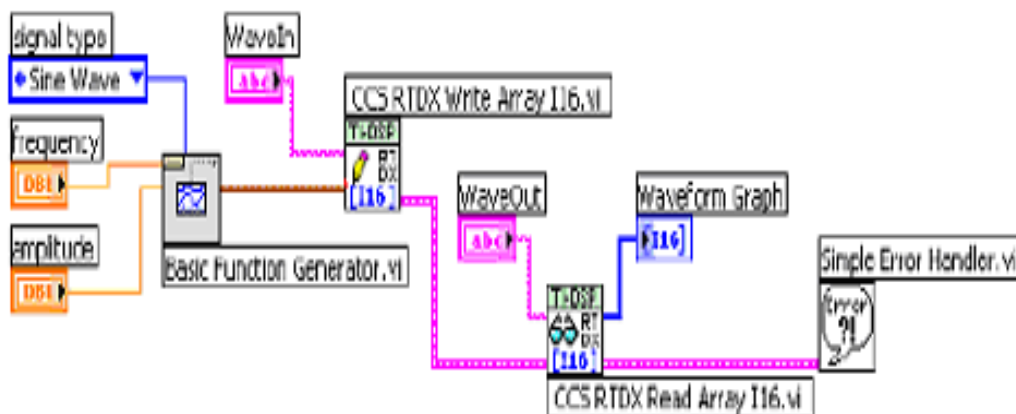
2. Funcțiile RTDX de comunicație implementate în DSP Test System



3. Diagrama programului de apelare în LabVIEW a programelor CCS



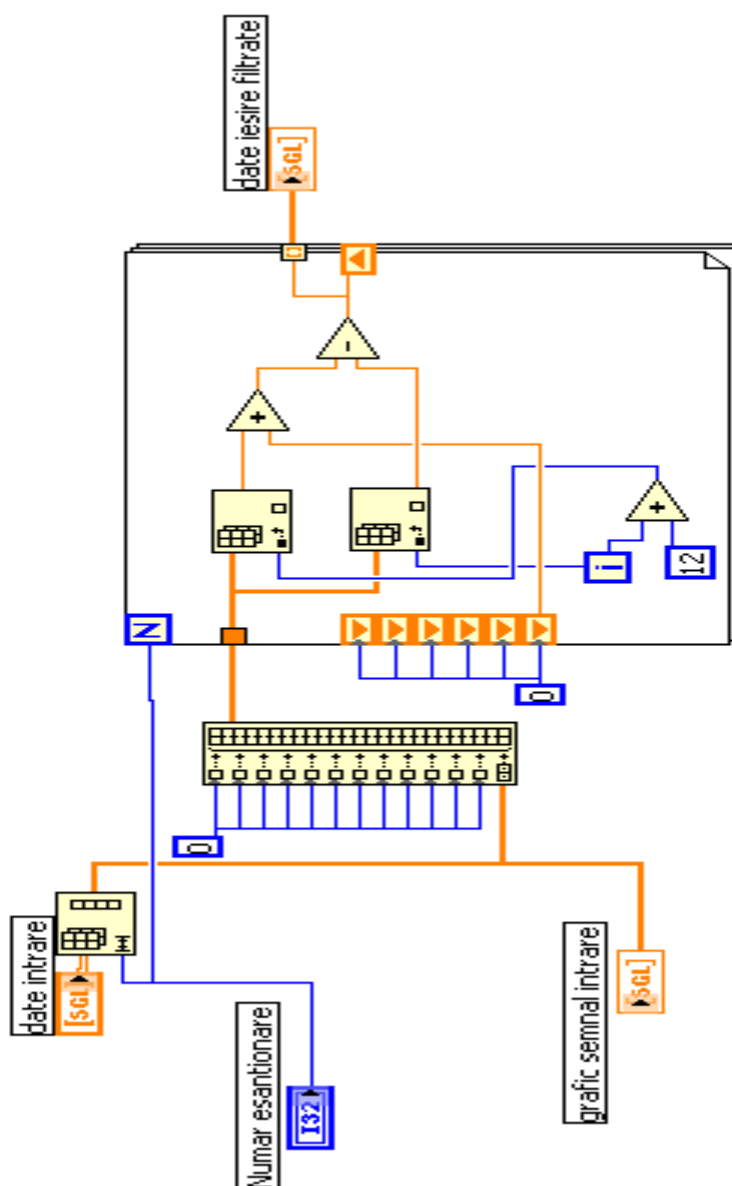
4. Diagrama programului de test pentru funcția RTDX în LabVIEW



Informații suplimentare despre „LabVIEW DSP Test Toolkit for TI DSP” se pot obține de pe sit-ul www.ni.com

Anexa 3

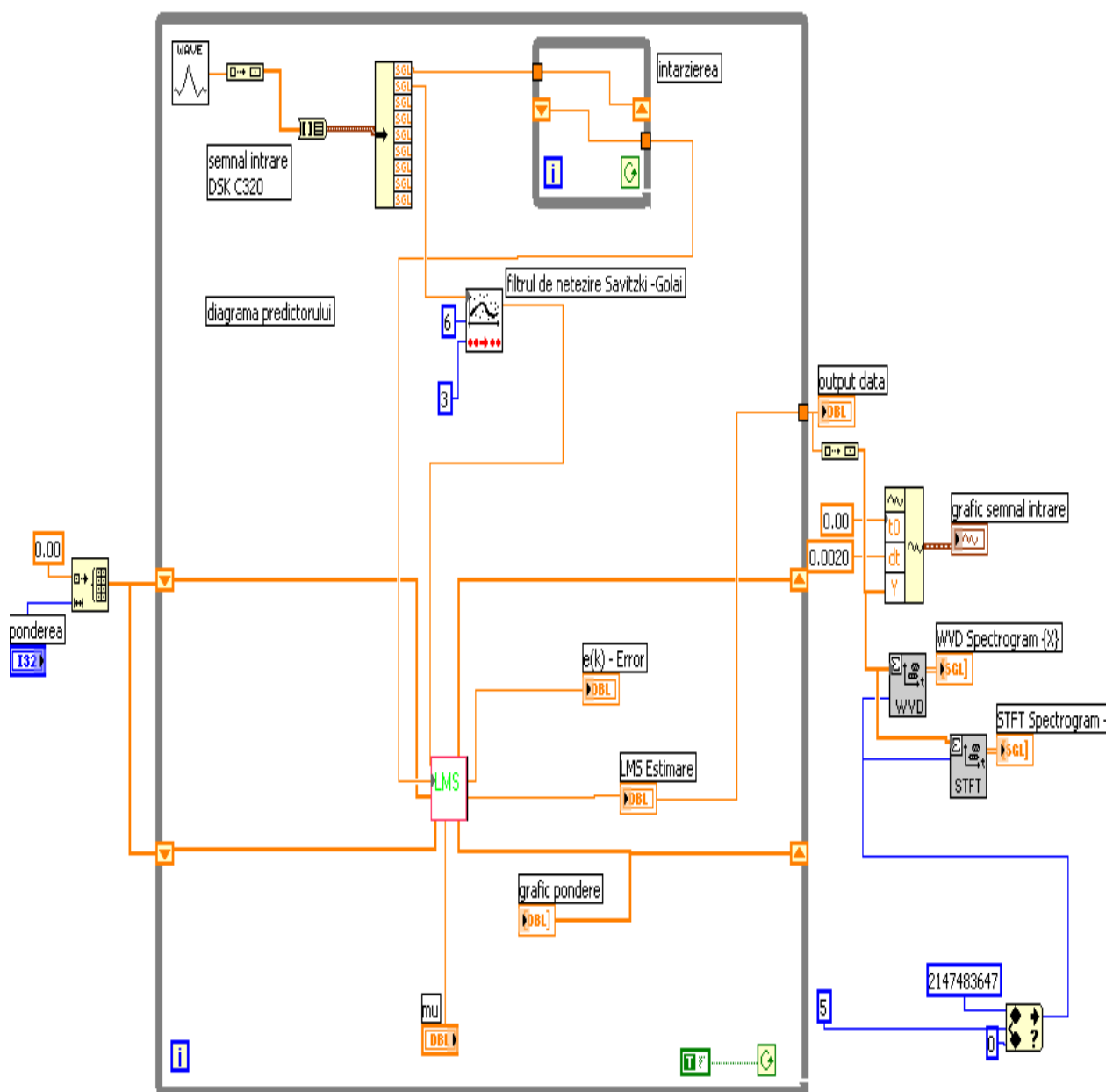
Diagrama de implementare LabVIEW a filtrului notch, conform ecuației sale cu diferențe finite



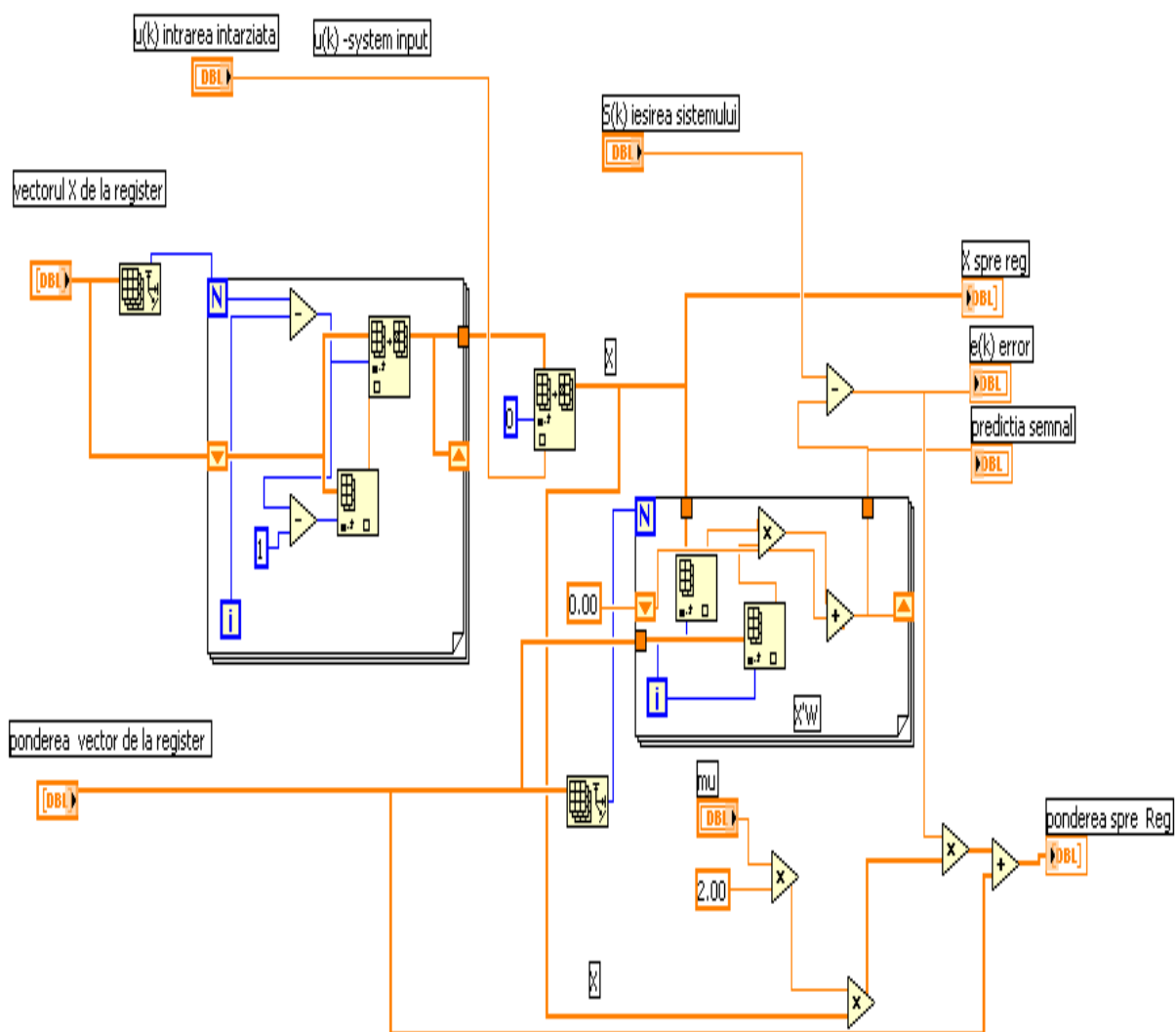
Anexa 4

Implementarea predictorului

1. Diagrama simplificată de implementare a predictorului



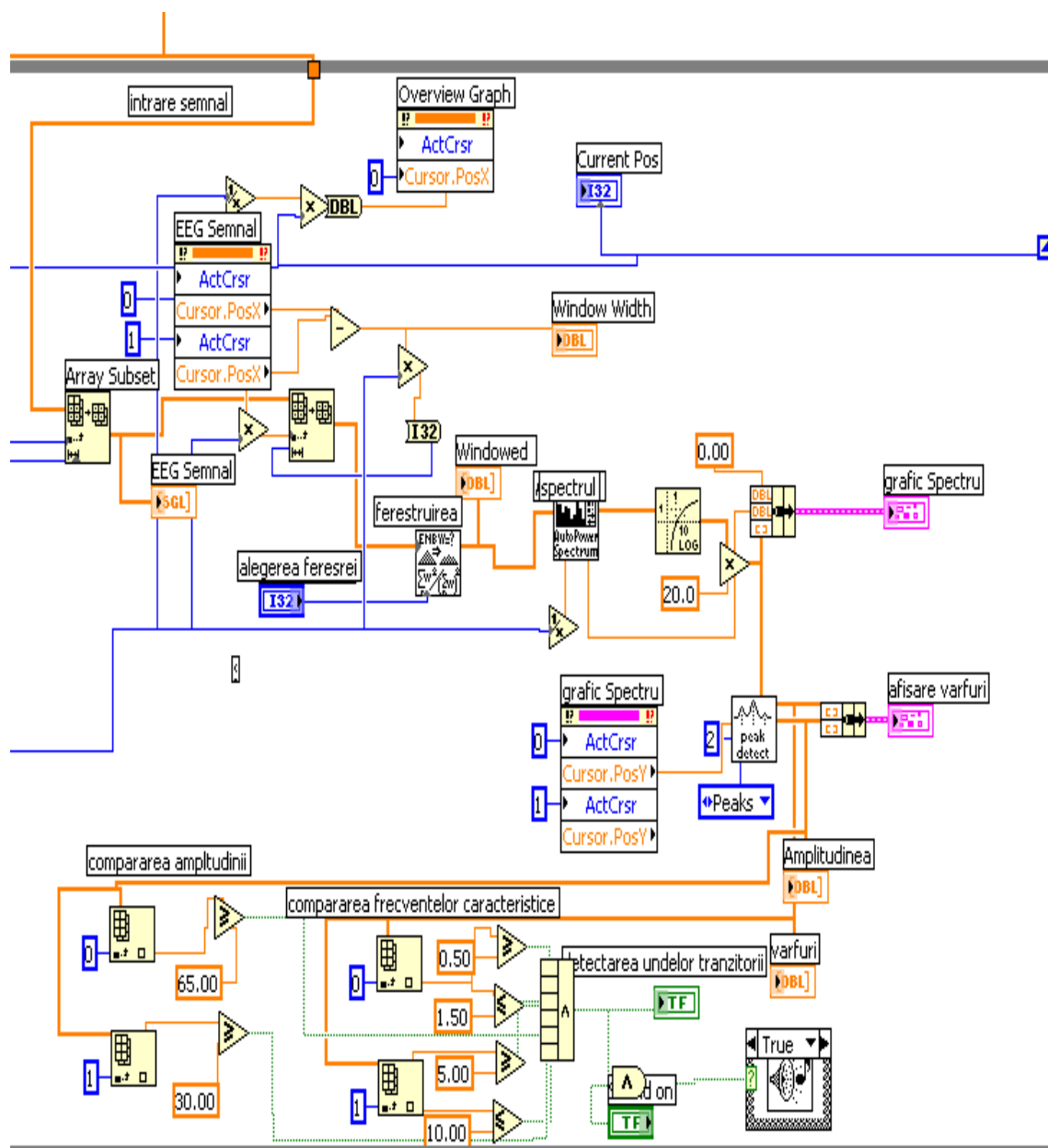
2. Diagrama algoritmului LMS



Anexa 5

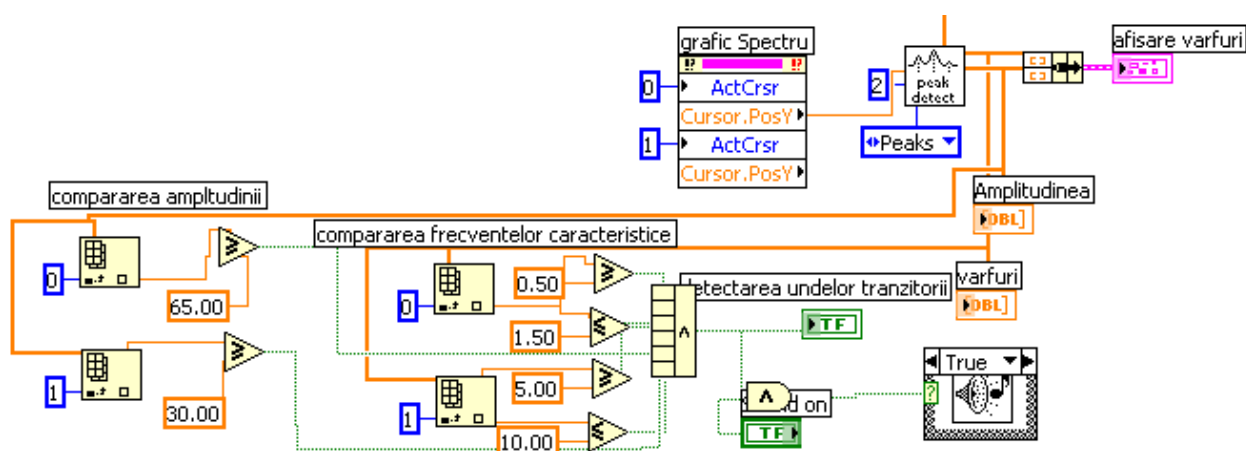
Implementarea programului test de detecție a undelor tranzitorii din EEG

1. Diagrama programului



2. Diagrama algoritmului de detecție pentru complexe k

Unda tranzitorie complex k se deosebește de fusurile de somn prin cele două vârfuri ale sale. Primul vârf are frecvența cuprinsă între 0.5 – 1.5 Hz, cu amplitudinea mai mare de 65 μ V, iar al doilea vârf are frecvența cuprinsă între 5 – 10 Hz, cu amplitudinea în jurul valorii de 30 μ V. Pragul de detecție a fost stabilit la 25 μ V, iar perioada fereastra de analiză a fost stabilită la 1.5 secunde, corespunzătoare perioadei de apariție a complexului k.



3. Rezultate

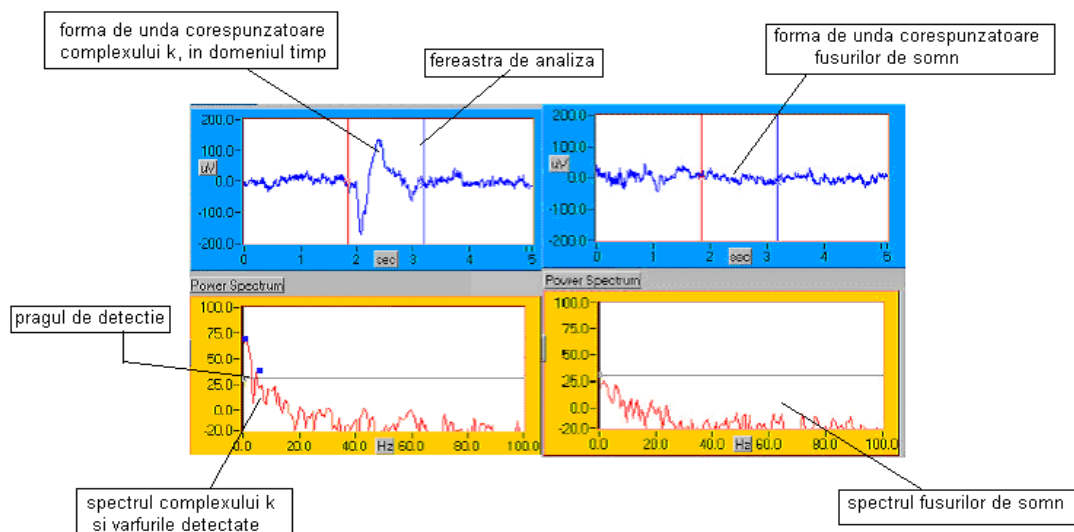


Figura 1. Detecția complex k și fusuri de somn.