
UNIVERSITATEA "POLITEHNICA" TIMIȘOARA
FACULTATEA DE ELECTRONICĂ ȘI TELECOMUNICAȚII
DEPARTAMENTUL DE ELECTRONICĂ APLICATĂ

Ing. IOAN LIE

**CONTRIBUȚII LA OPTIMIZAREA METODELOR ȘI
ECHIPAMENTELOR ELECTRONICE DE
INVESTIGARE CU ULTRASUNETE**

TEZĂ DE DOCTORAT

CONDUCĂTOR ȘTIINȚIFIC: Prof. Dr. Ing. TIBERIU MUREȘAN

TIMIȘOARA

2006

Cuprins

Introducere

Capitolul 1 Imagistica Ultrasonică

1.0 Scurt istoric	1-1
1.1 Principiul imagisticii ultrasonice	1-2
1.2 Evoluția sistemelor imagistice medicale	1-2
1.2.1 Modul A	1-2
1.2.2 Modul M	1-3
1.2.3 Modul B	1-4
1.3 Structura sistemelor ultrasonice moderne	1-8
1.4 Suportul Tehnologic	1-10
1.5 Comparația cu alte tehnici imagistice medicale	1-12
1.6 Concluzii	1-15

Capitolul 2 Formatoare de fascicul

2.1 Ecuațiile generale pentru sinteza fasciculelor ultrasonice	2-1
2.1.1 Cazul câmpului îndepărtat	2-1
2.1.2 Geometria ariei liniare	2-3
2.1.3 Cazul câmpului apropiat	2-3
2.2 Formatoare de fascicul ultrasonic tip întârziere – însumare	2-5
2.2.1. Întârzierea pentru dirijare și focalizare	2-6
2.2.2 Întârzierea și însumarea la emisie și recepție	2-8
2.2.3 Structura fasciculului	2-10
2.2.4 Eșantionarea în domeniul timp și în domeniul spațiu	2-12
2.2.5. Metode de control ale fasciculului de unde	2-13
2.3 Formatoare digitale de fascicul ultrasonic	2-14
2.3.1 Istoricul evoluției formatoarelor de fascicul	2-14
2.3.2 Formarea digitală a fasciculelor receptoare	2-16
2.4. Concluzii	2-23

Capitolul 3 Cuantificarea

3.1. Convertoare Rapide Multi – bit	3-1
3.2. CAN $\Delta\Sigma$	3-4
3.2.1 Tehnici de îmbunătățire a rezoluției	3-4
3.2.2 Supraeșantionarea	3-4
3.2.3 Formarea zgomotului	3-4
3.2.4.CNA pe un bit	3-5
3.2.5 CAN cu supraeșantionare	3-5
3.3 Modulatori $\Delta\Sigma$ de ordinul 1	3-6
3.4 Modulatori $\Delta\Sigma$ de ordinul 2	3-8
3.5 Arhitecturi cu o singură buclă	3-9
3.6 Arhitecturi interpolative	3-10
3.7 Modulatori $\Delta\Sigma$ în cascadă	3-10
3.8 Erorile de cuantificare a fazei	3-12
3.8.1 Lobii laterali într-un sistem nefocalizat	3-14
3.8.2 Lobii laterali într-un sistem focalizat	3-15
3.9 Cerințe de proiectare	3-18
3.10 Alegerea arhitecturii	3-19
3.11 Concluzii	3-20

Capitolul 4 Formator de fascicul ultrasonic cu modulație delta – sigma

4.1 Locul formatorului de fascicul în structura sistemului imagistic	4-1
4.1.1 Mecanismul de scanare	4-2
4.1.2 Alegerea tipului de focalizare	4-5
4.1.3 Metoda de eșantionare	4-5
4.1.4 Conversia formatului de scanare	4-6
4.2 Formatorul de fascicul cu modulație delta – sigma	4-7
4.3 Emularea software a formatoarelor de fascicul	4-9
4.4 Simularea unui formator de fascicul – varianta convențională	4-11
4.5 Simularea formatorului cu modulație delta – sigma	4-12
4.6 Formator cu modulație delta - sigma neuniformă	4-18
4.6.1 Evitarea repetiției eșantioanelor în procesul de formare a fasciculului ultrasonic	4-19
4.7 Discuție	4-22
4.8 Concluzii	4-26

Capitolul 5 Implementarea formatorului cu modulație delta - sigma neuniformă

5.1. Structura secțiunii digitale	5-1
5.1.1 Soluții de Implementare: FPGA sau ASIC	5-2
5.1.2 Caracteristicile familiei Flex 10K	5-3
5.2. Generatorul tactului de eșantionare neuniformă	5-5
5.2.1 Necesarul de memorie pentru formatorul de fascicul	5-5
5.2.2 Algoritmul punctului median	5-5
5.2.3 Implementarea generatorului	5-15
5.3. Memoria FIFO și Sumatorul	5-18
5.3.1 Dimensionarea memoriei FIFO	5-18
5.3.2 Sumatorul – optimizarea structurii	5-18
5.3.3 Descriere VHDL	5-18
5.4. Filtrul decimator	5-19
5.4.1 Filtru FIR cu fază liniară: structură și proprietăți	5-19
5.4.2 Efectul cuantificării coeficienților	5-24
5.4.3 Proiectarea în MATLAB a filtrului decimator	5-27
5.4.4 Conceptul hardware	5-31
5.4.5 Simulare + sinteza Flex10K	5-34
5.5. Implementarea unui formator cu 128 canale	5-36
5.6. Rezultate experimentale	5-37
5.7. Concluzii	5-40

Capitolul 6 Concluzii generale

6.1. Contribuții personale	6-1
6.2. Perspective de viitor	6-4

Bibliografie

Anexa 1

Anexa 2

INTRODUCERE

I see an important application of vision by ultrasound in medical diagnostics where it could not only replace X rays, but score above them by making visible fetuses, clogged veins and arteries, and incipient tumors.

- Dennis Gabor, 1970

Mobilul prezentei cercetări a fost dorința de a contribui la eforturile de realizare a unor sisteme de investigare ultrasonică (scanere) performante, ieftine și compacte. Cu ani în urmă scanerele ultrasonice erau mașini voluminoase și scumpe iar sistemele portabile nu apăruseră pe piață. Într-o viziune actuală dispozitivele pentru diagnosticul ultrasonic devin la fel de obișnuite ca și stetoscopul. Deși probabil va fi necesară o școlarizare de bază pentru a putea fi folosite astfel de dispozitive și a realiza un diagnostic corect, pe termen lung prezenta pe piață a unor sisteme ultrasonice ieftine va fi benefică pentru pacienți.

Scopul acestei lucrări a fost de a investiga și eficientiza metodele de formare a fasciculelor ultrasonice care permit implementarea unui număr mare de canale într-un singur circuit integrat. O sinteza de fascicul economică și compactă constituie un aspect cheie în dezvoltarea imagisticii ultrasonice medicale. Actualmente 64 sau 128 de canale sunt folosite pe scară largă în scanerele ultrasonice, sistemele performante folosesc 256 de canale, iar diagnoza ultrasonică tridimensională necesită un număr și mai mare de canale. Pe de altă parte există cerința de dispozitive portabile ieftine pentru utilizare în afara spitalelor în condiții de teren, unde consumul de putere și gradul de compactizare sunt factori importanți. S-a identificat ca fiind o abordare promițătoare folosirea convertoarelor analog numerice cu supraeșantionare în structura formatoarelor de fascicul deoarece acestea constituie o soluție simplă și ieftină de conversie și în plus permit o implementare mult mai compactă a formatorului comparativ cu situația în care se folosesc convertoare convenționale multibit.

Convertoarele analog digitale cu supraeșantionare au fost utilizate cu succes în domeniul audio și în telecomunicații. Folosirea lor împreună cu alte inovații tehnologice a facilitat utilizarea largă a conversiei de date de înaltă calitate conducând la disponibilitatea de soluții audio Hi-Fi și de telecomunicații ieftine. Ceea ce face convertoarele cu supraeșantionare utilizabile pe scara largă este posibilitatea acestora de a fi implementate în tehnologie CMOS. Acestea pot fi implementate împreună cu logica digitală pe un singur chip conducând la soluții uni-chip pentru interfațarea cu mediul analogic. Este de așteptat ca în câțiva ani să fie produse pe scara largă circuite care să conțină integrate pe un singur chip multiple modulatatoare delta-sigma a căror utilizare în scanerele ultrasonice vor conduce la simplificarea interconectării și a formării fasciculelor. Factorul care limitează folosirea convertoarelor cu supraeșantionare este raportul de supraeșantionare ridicat necesar și care împreună cu complexitatea modulatatoarelor determină calitatea conversiei. Acesta este motivul pentru care nu există multe exemple de convertoare cu supraeșantionare pentru semnale video sau de înaltă frecvență.

Abordarea domeniului formării digitale de fascicul în prezenta lucrare are în vedere tocmai evaluarea potențialului tehnicilor de formare prin supraeșantionare și decelarea posibilităților de perfecționare prin: identificarea unei structuri de convertor convenabilă din punct de vedere al raportului complexitate / performanță, implementarea unui mecanism de generare “on line” a întârzierilor pentru reducerea capacității de memorare necesare și optimizarea procedurii de sumare – filtrare – decimare din punct de vedere al vitezei de operare și al necesarului de resurse.

În primul capitol sunt indicate principiile de funcționare ale sistemelor imagistice ultrasonice și câteva moduri distincte de operare caracteristice pentru diferite generații de sisteme imagistice. Se prezintă de asemenea modul în care dezvoltările în electronică, tehnica de calcul și tehnologia materialelor au influențat perfecționarea aparaturii de investigare ultrasonică în diverse etape. În continuare pe baza unei arhitecturi generice se identifică locul și rolul fundamental al formatorului de fascicul în structura sistemelor de investigare ultrasonică. O succintă analiză comparativă a tehnicilor imagistice moderne, axată pe domeniile de aplicabilitate și pe avantajele/dezavantajele fiecăreia, a permis în finalul capitolului evidențierea locului și rolului imagisticii ultrasonice ca metodă paraclinică de bază în stabilirea diagnosticului.

Capitolul 2 constituie o abordare unitară a fundamentelor formării fasciculelor și a evoluției tehnicilor de formare. Evoluția tehnicilor de formare a fasciculelor ultrasonice este determinată de necesitatea îmbunătățirii calității imaginilor rezultate în urma scanării ultrasonice. Pornind de la cele afirmate mai sus sunt prezentate bazele matematice ale formării fasciculelor în general și a celor ultrasonice în special. Se realizează apoi o evaluare a tehnicilor analogice și digitale de formare a fasciculelor în scopul identificării posibilităților de optimizare sub raport complexitate performanță. Ca rezultat al evaluării se identifică în finalul capitolului două direcții de abordare pentru realizarea unei arhitecturi compacte de formator de fascicul: eșantionarea neuniformă și modulația delta-sigma.

Aspectele privind conversia, între mediul real, fundamental analogic și domeniul numeric, unde se concentrează în cea mai mare parte procesarea constituie obiectul capitolului 3. Pentru identificarea unei structuri de convertor $\Delta\Sigma$ convenabilă din punct de vedere al raportului complexitate / performanță se prezintă o analiză unitară a problematicii legate de cuantificarea semnalelor recepționate de o arie de traductoare ultrasonice. Analiza a fost orientată pe două direcții principale: identificarea și evaluarea efectului erorilor de cuantificare a fazei și determinarea dependenței complexitate performanță pentru convertoarele $\Delta\Sigma$. Concluziile desprinse pe cele două direcții de analiză permit alegerea arhitecturii potrivite pentru convertorul delta-sigma din structura formatorului de fascicul ultrasonic pentru satisfacerea cerințelor privind : formarea cu acuratețe a fasciculului și păstrarea complexității în limite rezonabile pentru a conserva șansa de integrare într-un singur circuit a convertoarelor pentru întreaga arie de traductoare.

Capitolul 4 tratează problematica formatoarelor de fascicul de tip întârziere-însurare care au la bază modulația delta-sigma. Sunt supuse analizei comparative variantele de realizare cu eșantionare uniformă și respectiv neuniformă. Având ca referință formatorul convențional multibit se urmărește alegerea unei soluții constructive care să asigure performanțe comparative cu cele ale formatorului clasic dar cu avantajul major al simplificării structurii hardware. Analiza care se bazează pe modelarea Matlab cu date de intrare reale a celor trei variante constructive de formator: clasic multibit, cu modulație delta-sigma uniformă și cu modulație delta-sigma neuniformă, cuantifică performanțele fiecărui tip de formator. Alegerea soluției constructive pune în discuție la performanțe similare complexitatea hardware a fiecărei variante. Din acest punct de vedere rezultă că soluția cu modulație neuniformă este mai simplă deoarece nu presupune modificarea modulatorului ci doar asigurarea raportului de frecvențe pentru a se evita repetiția, suportul de memorie necesar pentru generarea și controlul întârzierilor fiind practic același pentru ambele variante.

Capitolul 5 tratează din punct de vedere structural secțiunea digitală a formatorului de fascicul cu modulație delta-sigma neuniformă. Primul aspect abordat a fost alegerea tehnologiei de implementare a secțiunii digitale pe baza unei analize comparative privind avantajele și dezavantajele tehnologiilor de implementare cu structuri logice programabile –

FPGA, respectiv cu circuite integrate dedicate - ASIC. Determinarea întârzierii necesare fiecărui element traductor spre a asigura focalizarea dinamică pentru fiecare punct dintr-un cadru imagine necesită efectuarea cu rapiditate a unui mare număr de calcule complexe în timp real sau un volum de stocare considerabil pentru memorarea întârzierilor calculate “off-line”. O soluție pentru reducerea necesarului de memorie constă în compresia informației pentru focalizare. În acest sens pentru generarea tactului de eșantionare neuniformă s-a investigat posibilitatea calculării iterative în timp real, “în circuit”, a informației de întârziere pe baza algoritmului punctului median aplicat unei ecuații care derivă din geometria de focalizare. Sunt prezentate arhitecturile blocurilor care constituie structura digitală a formatorului, modelarea VHDL și sinteza într-un FPGA din familia Flex. În finalul capitolului sunt evaluate șansele de implementare a întregii secțiuni digitale într-un circuit programabil de capacitate adecvată

Capitolul 6 prezintă sinteza contribuțiilor personale, redă concluziile finale și sugerează posibile direcții de cercetare ulterioară în domeniul abordat.

Lucrarea conține 6 capitole și 2 anexe cuprinzând 199 pagini și 143 referințe bibliografice dintre care 25 personale ca unic sau prim autor.

* * *

Teza de doctorat a fost elaborată sub atenta îndrumare a domnului prof. dr. ing. Tiberiu Mureșan, căruia autorul dorește să îi mulțumească în mod deosebit pentru coordonare, pentru sugestiile și sfaturile deosebit de utile, pentru sprijinul acordat și nu în ultimul rând pentru înțelegerea de care a dat dovadă.

Autorul își exprimă gratitudinea față de cadrele didactice ale Universității Politehnice din Timișoara, care au contribuit decisiv la formarea sa profesională, atât în timpul studenției, cât și în timpul activității de cadru didactic la Facultatea de Electronică și Telecomunicații, Departamentul de Electronică Aplicată.

Nu în ultimul rând, autorul dorește să mulțumească soției pentru înțelegerea și sprijinul acordat pe parcursul elaborării tezei.

* * *

Lucrarea prezintă căutările proprii în domeniul formării fasciculelor ultrasonice și al căilor de obținere a unor soluții compacte de formatoare. Sper ca ea să constituie o lectură utilă și o bună bază pentru cercetarea viitoare în acest domeniu.

CAPITOLUL 1

IMAGISTICA ULTRASONICĂ

Folosirea ultrasunetelor în medicină ca unealtă de vizualizare datează de la începutul anilor 50. Primele sisteme imagistice operau cu un singur traductor. De atunci performanțele sistemelor imagistice ultrasonice s-au îmbunătățit continuu în special datorită progresului în electronică și procesarea semnalelor. În zilele noastre imagistica bi-dimensională și maparea color a curgerii sangvine sunt facilitate obișnuite pentru sistemele imagistice ultrasonice, provocarea actuală fiind imagistica tridimensională.

Acest capitol indică principiile de funcționare ale sistemelor imagistice ultrasonice și câteva moduri distincte de operare caracteristice pentru diferite generații de sisteme imagistice. Se prezintă de asemenea modul în care dezvoltările tehnologice au influențat perfecționarea aparaturii de investigare ultrasonică în diverse etape. În final s-a realizat o succintă analiză comparativă a celor mai populare tehnici imagistice medicale axată pe domeniile de aplicabilitate și pe avantajele/dezavantajele fiecăreia.

1.0 SCURT ISTORIC

Istoria utilizării ultrasunetelor are ca punct de plecare anul 1880, când frații Pierre și Jacques Curie descoperă efectul piezoelectric. Timp de peste 30 de ani descoperirea fraților Curie nu și-a găsit aplicații practice. După scufundarea Titanicului, la 14 aprilie 1912, mulți oameni de știință și-au pus problema detectării la distanță a obstacolelor submarine. Primul care a propus folosirea în acest scop a ultrasunetelor a fost englezul Richardson într-o cerere de brevet redactată în 1913 dar rămasă neimplementată în absența unui suport tehnologic adecvat. Fizicianul francez Paul Langevin, beneficiind de disponibilitățile tehnologice ale piezoelectricității și ale tuburilor amplificatoare cu vid, este autorul unui prim rezultat practic. După doi ani de experimentări în laborator și în porturi acesta a brevetat în 1916 sub titlul “Dispozitiv în vederea producerii de semnale submarine dirijate și a localizării la distanță a obiectelor submarine” rezultatul cercetărilor sale. [Biquad72] Ulterior, a avut loc o rapidă perfecționare a aparaturii, iar ultrasunetelor li s-a dat o largă utilizare în marină și în defectoscopia metalelor.

Primele încercări de aplicare a ultrasunetelor în medicină au fost făcute de Wood și Loomis în 1927, urmați de Sokolov în 1937, Pohman, Richter și Parow în 1938 care au experimentat utilizarea ultrasunetelor în scop terapeutic. Încercările de utilizare a ultrasunetelor în diagnostic datează din deceniul al 5-lea. În 1942, Karl Dussik detectează tumori intracraniene cu ajutorul a două traductoare piezoelectrice plasate pe cap. George Ludwig măsoară viteza de transmitere a ultrasunetelor prin țesuturile moi. Douglas Howry înregistrează ecourile ultrasunetelor la interfața dintre grăsime și mușchi.

Aplicarea în diagnostic a ultrasunetelor a luat un mare avânt după anul 1955 când John Wild și John Reid au detectat tumori la sân și tumori ale intestinului gros. În continuare s-a trecut la utilizarea ultrasunetelor în obstetrică și ginecologie. În 1958, Jan Donald a pus în evidență chisturi ovariene și a reușit să măsoare diametrul biparietal al capului fătului, iar în 1960, Donald și Brown au pus în evidență sarcini multiple. [Szabo 04] Utilizarea largă a metodelor ultrasonice în medicina clinică datează din 1970, și ca urmare a unei experiențe bogate și a perfecționării aparaturii s-a extins foarte mult cuprinzând diverse ramuri medicale ca: medicină internă, urologie, endocrinologie, cardiologie, obstetrică și ginecologie, oftalmologie, chirurgie. [Erikson74]

Paralel cu extinderea utilizării ultrasunetelor în diagnoza medicală s-au efectuat studii fundamentale privind modul și căile de interacțiune, precum și efectele ultrasunetelor asupra materialului biologic. Ultrasunetele fiind un fenomen oscilator elastic și purtând o energie radiantă, contactul lor cu un mediu biologic produce toate fenomenele caracteristice interacțiunii dintre un câmp radiant și substanță.

1.1. PRINCIPIUL IMAGISTICII ULTRASONICE

Ultrasunetele sunt vibrații mecanice ale particulelor unui mediu elastic ale căror frecvențe de oscilație sunt superioare pragului de percepție al urechii umane – 20KHz și care se propagă în mediu sub formă de unde elastice. Undele ultrasonice pot fi direcționate și focalizate într-un punct specificat în ipoteza că mediul în care se propagă este omogen și izotrop. Într-un sistem imagistic traductorul piezoelectric este excitat cu un impuls electric și produce modificare de presiune în mediul învecinat. Pe măsură ce unda emisă se propagă orice discontinuitate din mediu creează unde reflectate care sunt mai apoi detectate de către traductor. Semnalul receptionat conține informație referitoare la poziția și caracteristicile elastice ale discontinuităților (structuri difuzante, difuzori = scatterers).

Pe durata propagării unda este atenuată atât datorită distanței parcurse cât și prin absorbția energiei de către mediul în care are loc propagarea. Întrucât în mediile biologice (corpul uman) atenuarea este semnificativă aceasta va trebui compensată. Se folosesc pentru aceasta amplificatoare cu câștig variabil în timp (TGC – Time Gain Control) a căror amplificare este minimă în momentul transmisiei și crește în funcție de distanța parcursă. Procesul poartă numele de compensare temporală a câștigului. Rezoluția care se poate obține cu astfel de sisteme imagistice este cu atât mai bună cu cât sunt utilizate frecvențe mai ridicate. În prezent sistemele imagistice de uz general operează în domeniul 3-12 MHz.

Principalul obiectiv al investigării ultrasonice constă în studierea modificărilor pe care le suferă undele ultrasonice atunci când străbat structuri cu proprietăți acustice diferite. La granița dintre astfel de structuri ia naștere un fenomen de reflexie proporțional cu diferența de impedanță acustică dintre cele două medii. Ca urmare undele sonore sunt parțial reflectate, dând naștere la ecouri, iar parțial sunt transmise mai departe prin mediu. Pe baza acestui fenomen s-au dezvoltat trei categorii de tehnici de investigare ultrasonică:

- Tehnicile tip "Puls-Ecou" – care permit localizarea structurilor difuzante și analiza densității lor pe baza ecourilor ultrasonice
- Tehnicile Doppler și tehnicile corelative – cu ajutorul cărora se determină caracteristicile mișcării prin evaluarea frecvenței și a gradului de corelare al ecourilor.
- Tehnicile "prin transmisie", în care ultrasunetele traversează complet zona examinată iar semnalele rezultate oferă informații privind timpul de tranzit și gradul de atenuare datorat structurii mediului prin care s-au propagat.

1.2 EVOLUȚIA SISTEMELOR IMAGISTICE MEDICALE

În cele ce urmează se prezintă o trecere în revistă a sistemelor imagistice de la introducerea lor până în prezent. Sistemele de investigare ultrasonică de tip "Puls-Ecou" realizează trei moduri fundamentale de reprezentare: modul A; modul TM (M) și modul B. Actualmente aparatele sofisticate pot realiza toate aceste moduri de funcționare.

1.2.1 MODUL A.

Primele sisteme imagistice ultrasonice au fost cele cu operare în modul A (Amplitudine). Funcționarea acestora era similară cu a unui osciloscop și se prezenta la scară

logaritmică evoluția funcție de timp (adâncime) a semnalelor ecou. Pentru un astfel de sistem deflexia verticală pe ecran reprezintă amplitudinea ecoului, iar cea orizontală timpul. Presupunând viteza de propagare în mediu constantă, axa timpului este totodată și axă a distanței. Informația afișată nu permite evaluarea internă a structurii difuzante dar mișcarea poate fi ușor detectată datorită ratei ridicate de reîmprospătare a imaginii. Imaginea este unidimensională cu privire la mediul investigat dar depinde de timp. Figura 1.1 prezintă schema bloc pentru realizarea acestei reprezentări. Traductorul pentru acest mod de funcționare este de tip monoelement sau multielement, comandat în așa fel încât să radieze un fascicul ultrasonic cu poziție fixă în spațiu. Blocul de control declanșează generatorul de impulsuri pentru comanda emisiei, generatorul de rampă pentru deflexia pe orizontală și generatorul semnalelor de marcare, care etalonează desfășurarea pe orizontală în distanță.

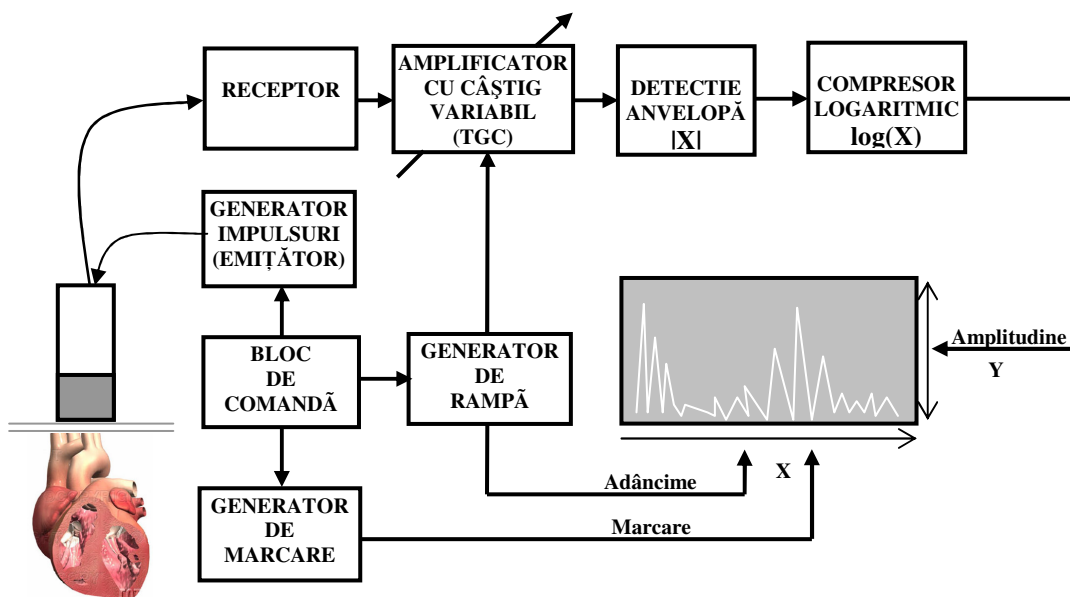


Figura 1.1 Arhitectura unui sistem care operează în Modul A

Semnalul generat de traductorul piezoelectric se propagă în mediu și este parțial reflectat și împrăștiat pe interfețele acustice întâlnite. Ecoul este recepționat de același transductor și apoi este amplificat într-un amplificator cu corecția temporală a câștigului (TGC – Time Gain Compensation). Generatorul de rampă acționează asupra amplificatorului receptor modificându-i câștigul în timp pentru a compensa atenuarea ultrasunetelor în mediu, proporțional cu distanța parcursă. Semnalului rezultat i se detectează anvelopa care apoi este supusă unei compresii logaritmice spre a permite afișarea unei game largi de ecouri.

Avantajele principale ale echipamentelor care realizează modul A sunt simplitatea și livrarea informației practic în timp real. Caracteristica de timp real are la bază faptul că frecvența de repetiție a impulsurilor ultrasonice este suficient de mare în comparație cu vitezele structurilor pe care are loc reflexia. Datorită faptului că nu dă informații bidimensionale traductorul este greu de orientat, utilizarea acestui mod restrângându-se la oftalmologie, cardiologie și encefalografie precum și în testarea nedistructivă (NDT).

1.2.2 MODUL M.

Pentru vizualizarea mișcării s-au dezvoltat sistemele care operează în modul M (Motion). În aceste sisteme amplitudinea semnalului ecou este reprezentată prin gradul de strălucire, axa verticală reprezintă adâncimea iar ecourile corespunzătoare emisiilor consecutive pe aceeași direcție sunt afișate pe orizontala unul după altul.

În acest mod se prelucrează ca și la modul A ecourile recepționate de la un traductor care emite și recepționează într-o direcție fixă în spațiu. Spre deosebire de modul A în modul M ,

figura 1.2, prezența și intensitatea ecourilor este materializată prin modulația intensității spotului, pe o rampă A aplicată circuitului de deflexie pe verticală a tubului de afișare, în punctele care corespund poziției în spațiu a surselor ecoului. Simultan, rampa T aplicată circuitului de deflexie pe orizontală poziționează următoarea rampă A. Astfel pe ecran rezultă evoluția în timp a poziției surselor de ecou situate pe direcția examinată. Durata rampei T reprezintă scara de timp a structurilor examinate. Acest mod de reprezentare este la fel ca modul A, unidimensional în raport cu țesutul examinat dar depinde de timp.

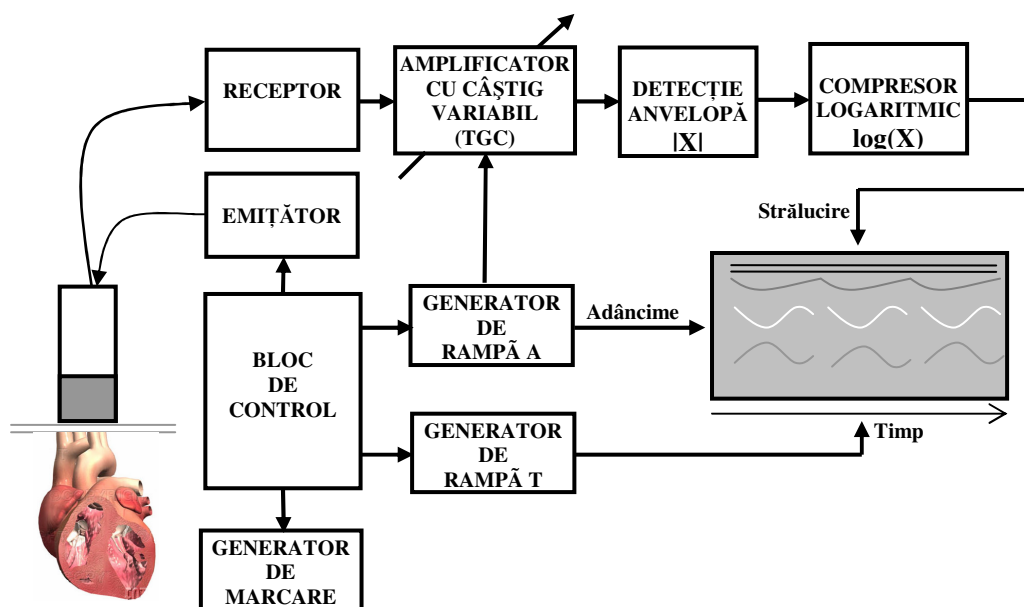


Figura 1.2 Arhitectura unui sistem care operează în Modul M

1.2.3 MODUL B

Un pas semnificativ în evoluția sistemelor imagistice a fost introducerea sistemelor în mod B. În acest mod dispozitivul de afișare reconstituie în două dimensiuni o secțiune în mediu pe baza ecourilor rezultate din explorarea regiunii respective cu o serie de fascicule ultrasonice generate de către traductor. Fiecare linie din imagine corespunde unei poziții a traductorului. Punctele care dau ecouri sunt marcate prin modularea intensității spotului de unde și numele modului „B” - Brightness = Strălucire. [Wells77]

Ceea ce diferențiază sistemele în mod B de cele anterioare este modalitatea de baleiere a regiunii de interes cu ajutorul traductorului. Cronologic au fost adoptate următoarele două soluții:

- manevrarea liberă a traductorului peste regiunea de interes și detecția poziției acestuia
- controlul mișcării traductorului peste regiunea de interes.

Sistemul cu braț de scanare – a fost sistemul care a realizat imagini bidimensionale ale structurilor explorate folosind prima soluție, fiind folosit pe scară largă cu începere din anii 50 și până în anii 70. Traductorul utilizat era atașat unui braț articulat, ușor manevrabil, care asigura codificarea în coordonate $x(t)$, $y(t)$ a poziției sale și comunicarea informației la display. Manevrarea se făcea manual de către operator iar aparatul memora ecourile rezultate de la fasciculele ultrasonice emise în pozițiile succesive ale traductorului. În acest mod prin alegerea unui set suficient de extins de linii de scanare se putea obține o imagine detaliată a regiunii explorate. De altfel în această perioadă obiectivul scanării era obținerea celei mai bune „fotografii” a regiunii explorate. Deplasarea fiind manuală informațiile necesare construirii unei imagini se acumulau într-un timp relativ lung, de ordinul secundelor. Din acest motiv imaginea obținută era statică.

Baleierea prin controlul mișcării a condus la dezvoltarea sistemelor cu scanare liniară la care traductorul executa o mișcare de translație și respectiv a sistemelor cu scanare sectorială la care traductorul executa o mișcare de rotație. În ambele situații controlul poziției se realiza sincron cu startul unei noi emisii. Sistemele cu scanare liniară afișează un set de linii orizontale dispuse succesiv pe verticală și a căror dimensiune indică adâncimea de scanare. Din combinarea celor două traiectorii ale traductoarelor au rezultat câteva metode de scanare a căror utilizare a fost dictată de particularitățile zonei investigate.

- scanarea liniară – este metoda cea mai simplă constând în deplasarea traductorului pe o suprafață plană sau după o linie dreaptă - figura 1.3 - a
- scanarea sectorială (unghiulară) – se realizează prin deplasarea traductorului pe un arc de cerc fără translație - figura 1.3 - b.
- scanarea compusă – corespunde situației în care traductorul aflat în translație execută și o mișcare de rotație în jurul axei sale în fiecare dintre punctele traiectoriei liniare
- scanarea mixtă – în situația când traductorul trece succesiv din rotație în translație și înapoi în mișcare de rotație - figura 1.3 - d.

Ultimele doua metode au fost impuse de situațiile în care suprafața de scanat era curbă sau circulară.

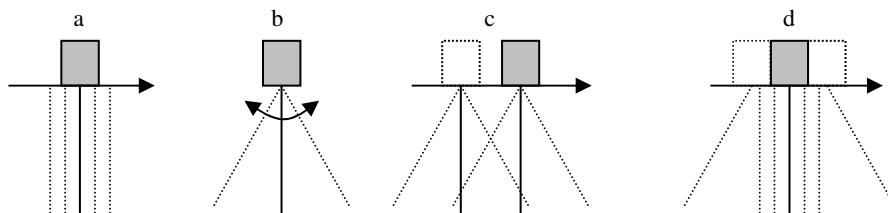


Figura 1.3 Metode de scanare

Sistemul cu scanare sectorială mecanică

Folosește un traductor monoelment presupus a avea dimensiuni foarte mici și care este rotit în plan pentru a emite fascicule ultrasonice care vor acoperi un sector circular. Din punct de vedere constructiv există mai multe posibilități de implementare a deflexiei mecanice a fasciculului ultrasonic:

- un traductor monoelment este cuplat printr-un mecanism adecvat la un motor de antrenare ;
- traductorul monoelment este fix în focarul unei oglinzi rotite de un motor;
- trei traductoare monoelment plasate la 120° sunt rotite de un motor și sunt comandate secvențial pentru emisie și recepție când trec prin dreptul unei ferestre. Unghiul de deschidere al sectorului este în jur de 90° pentru cardiologie și pentru explorarea cavității abdominale sau 360° în cazul explorării urologice.

În situația folosirii unui traductor monoelment al cărui element sensibil poate fi rotit, aria explorată are o formă sectorială fiind necesară conversia formatului de scanare înaintea afișării informației. Prin conversia formatului informația corespunzătoare direcțiilor de scanare este plasată într-o matrice rectangulară astfel încât geometria regiunii investigate sa poată fi afișată fără distorsiuni pe display.

Numărul de linii de scanare depinde de: adâncimea de explorare , viteza de propagare a ultrasunetelor în mediu și de numărul de imagini pe secundă. Apar probleme de execuție datorită prezenței pieselor în mișcare și a unui lichid de transmisie pentru cuplarea la mediu. Electronica aferentă se complică datorită necesității conversiei de la coordonatele polare în care se scanează la coordonatele rectangulare necesare pentru afișare. Arhitectura unui astfel de sistem este indicată în figura 1.4.

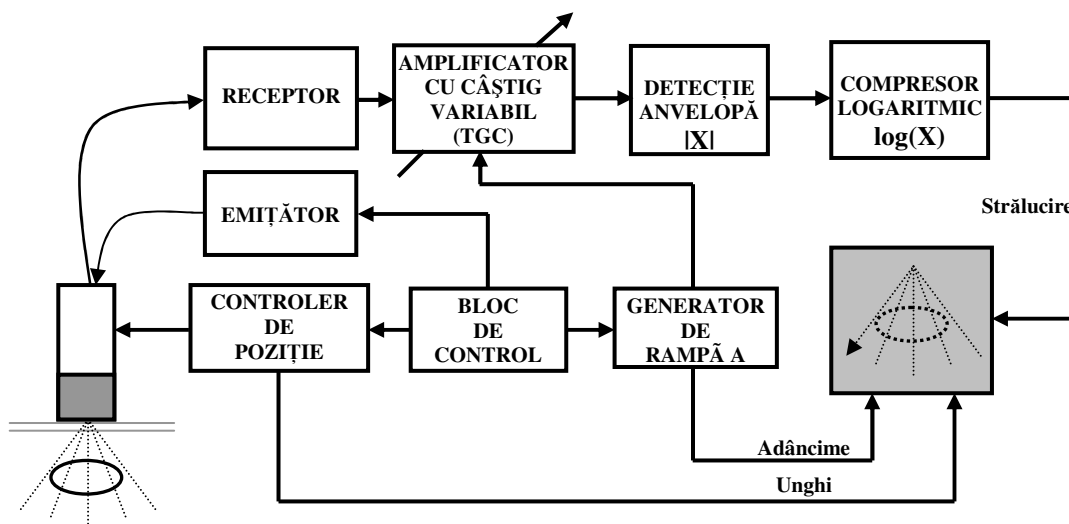


Figura 1.4 Arhitectura sistemului cu scanare sectorială mecanică

Sistem cu traductor multielement liniar.(figura 1.5)

Mișcarea de scanare a unui singur traductor după o linie dreaptă poate fi simulată electronic prin utilizarea traductorului multi-element liniar comandat în mod corespunzător. În principiu fiecărui element i se poate asocia o linie de scanare, fiecare element putând fi emițător și receptor, apoi următorul etc. Din motive de focalizare, liniile de scanare se asociază unui grup de elemente comandate într-un mod corespunzător. Rezultatul este o imagine bidimensională care la viteze mari de scanare se obține practic în timp real. Se pot astfel examina și structuri în mișcare.

Un dezavantaj al primelor sisteme imagistice care implementau acest concept era timpul lung de achiziție și neclaritatea imaginii datorată structurilor în mișcare. Tehnologiile de realizare a traductoarelor au cunoscut un ritm susținut de dezvoltare, începând cu anii 80. Au fost realizate arii liniare și arii fazate cu sute de elemente și frecvențe ridicate de operare. Introducerea traductoarelor multielement și perfecționarea tehnologiei digitale a permis trecerea la dirijarea și focalizarea electronică a fascicului ultrasonic.

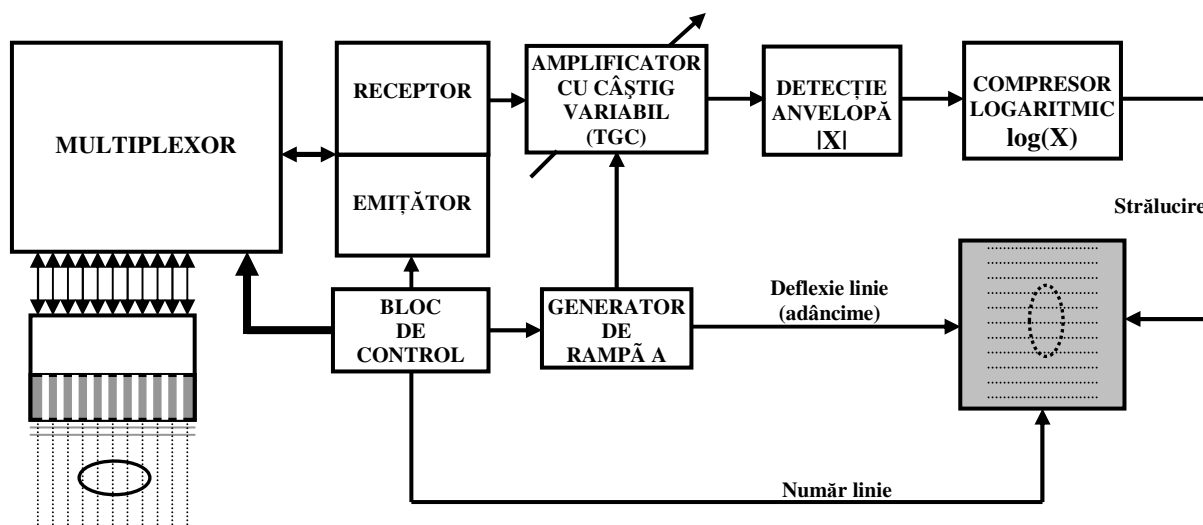


Figura 1.5 Arhitectura sistemului cu traductor multielement liniar

Sistemele imagistice moderne operează în modul B în timp real și afișează pe ecran un număr mare de imagini pe secunda. Informația corespunzătoare punctelor intermediare dintre direcțiile de scanare se obține prin interpolare. Figura 1.6 arată arhitectura unui sistem Mod B în timp real. Scanările moderne au un anumit număr de canale independente și folosesc traductoare multielement. Se utilizează două tipuri de astfel de traductoare: arii liniare și arii fazate.

Ariile liniare pot fi plate sau convexe și sunt operate folosind în același timp mai multe canale adiacente atât la emisie cat și la recepție. Rezultatul explorării este un set de linii paralele care, înaintea afișării, sunt interpolate în modulul de conversie a formatului de scanare. In ariile fazate toate elementele sunt utilizate împreună iar dirijarea fascicului se realizează prin întârzierea corespunzătoare a semnalelor de excitație pentru diferite elemente.

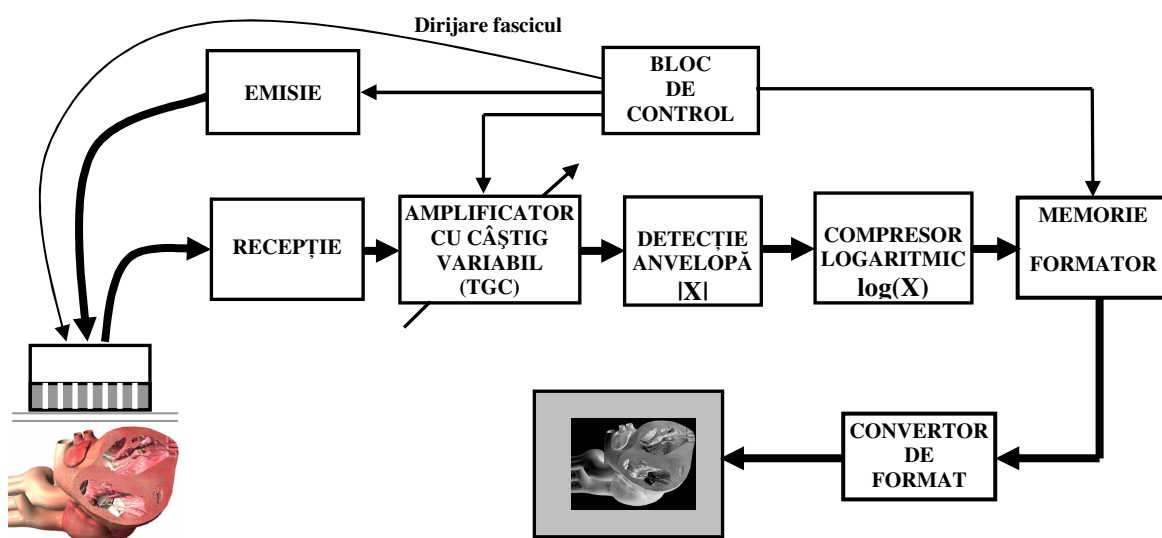


Figura 1.6 Arhitectura unui sistem Mod B în timp real

Figurile 1.7 și 1.8 indică modalitatea de formare a liniei de imagine pentru cele două tipuri de arii de traductoare. Se remarcă simetria întârzierilor în cazul ariilor liniare și faptul ca setul de întârzieri rămâne același pentru toate liniile de imagine, modificând-se elementele care participă la formarea liniilor succesive. Practic scanarea se realizează prin înlocuirea unui număr de elemente din apertura activă și se obține o imagine rectangulară alcătuită din linii paralele perpendiculare pe suprafața ariei. Dezavantajul este supradimensionarea ariei pentru a se scana regiuni extinse și necesitatea unei ferestre acustice de dimensiunea ariei. Acest tip de operare este potrivit pentru explorarea abdominală.

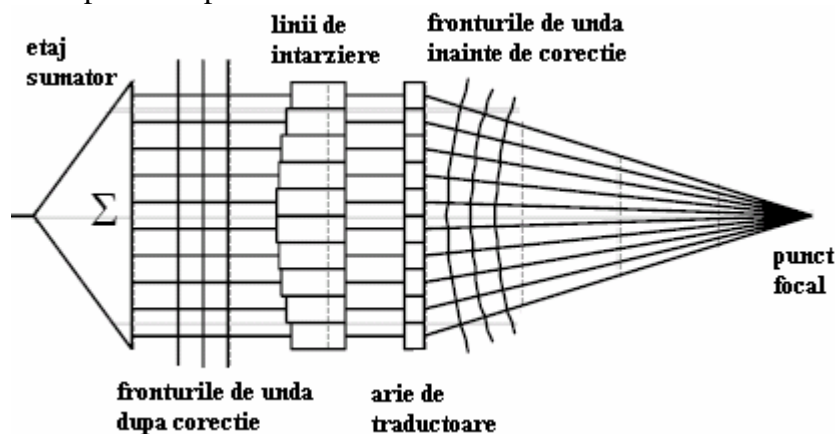


Figura 1.7 Formarea liniei de imagine cu arie liniară

În cazul ariilor fazate setul de întârzieri este univoc definit pentru fiecare linie de imagine care este formată prin aportul tuturor elementelor ariei – fig. 1.8. Se obține o imagine sectorială alcătuită dintr-un set de linii care converg în centrul ariei. Acest mod de operare s-a impus de oarece permite explorarea unor sectoare extinse prin ferestre acustice de dimensiuni mult mai reduse. Un exemplu tipic este explorarea cardiacă prin ferestre acustice relativ reduse din spațiul intercostal.

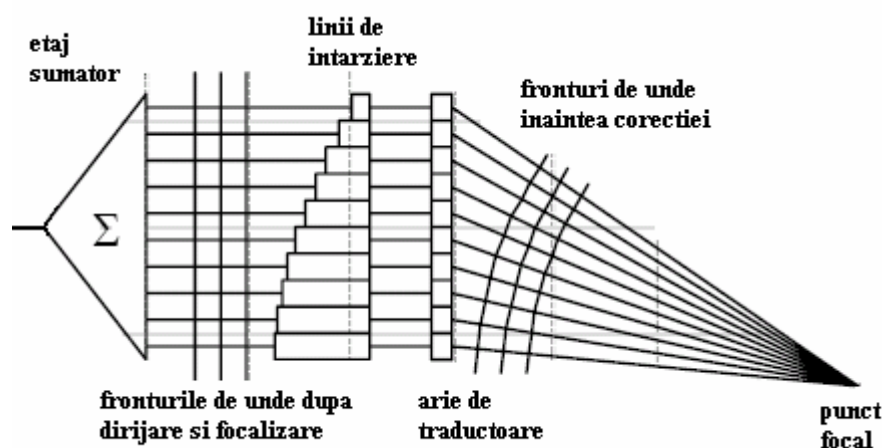


Figura 1.8 Formarea liniei de imagine cu arie fazată

1.3 STRUCTURA SISTEMELOR ULTRASONICE MODERNE

Tehnologia ultrasonică s-a îmbunătățit semnificativ în ultimii ani, datorită tehnicilor de fabricare a dispozitivelor semiconductoare integrate și a integrării pe scară largă. Aparare complete numerice cu arii de traductoare cu număr mare de canale, se găsesc pe piață, asigurând o rezoluție foarte bună imaginii și o mulțime de modalități de imaginare 2D și chiar 3D. Datorită costurilor relativ mari acestea coexistă momentan cu un set de aparare ultrasonice portabile cu performanță mai scăzută, care folosesc tehnici analogice de formare a fasciculului ultrasonic și traductoare cu număr mic de canale. Este doar o chestiune de timp până când aparare ultrasonice portabile, complete digitale să fie disponibile pentru ambulatoriu și clinici rurale în întreaga lume.

Diagrama bloc prezentată în figura 1.9 ilustrează structura de generică a sistemelor imagistice ultrasonice digitale actuale.

Interfața utilizator

Reprezintă acea parte a sistemului prin care utilizatorul poate configura sistemul pentru a opera în modul dorit. Sistemul afișează meniuri sort configurabile și elemente de control virtuale care operează în combinație cu elementele de control hard – potențioetre, comutatoare. Imaginea de pe monitorul principal oferă confirmarea vizuală a funcționării în modul dorit. Interfața utilizator oferă conexiunile adecvate pentru extragerea sau introducerea informației din/in sistem. Principalele conexiuni sunt cele la rețeaua locală sau la una specializată (DICOM), la dispozitive de stocare (CD-ROM, DVD) precum și la diverse periferice cum ar fi imprimanta.

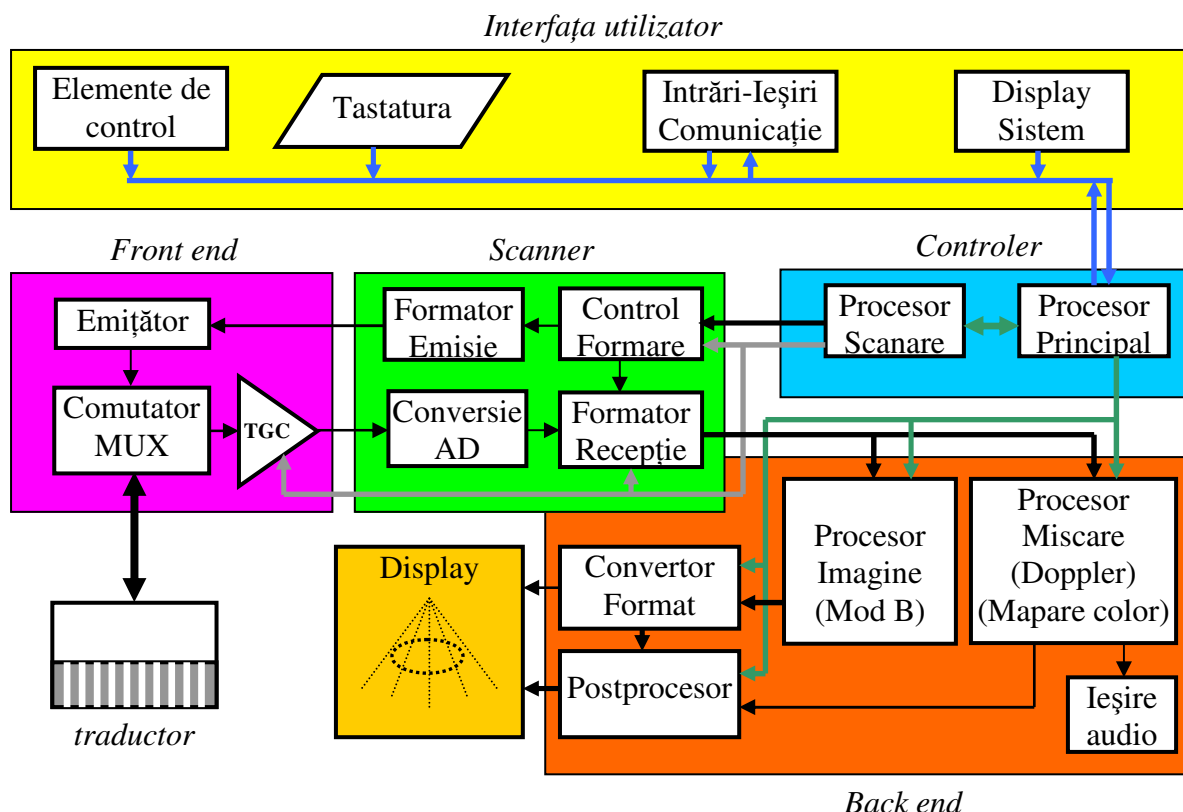


Figura 1.9 Arhitectura generică a sistemelor imagistice ultrasonice digitale

Controlerul

Un sistem tipic are unul sau mai multe microprocesoare sau PC-uri care conduc funcționarea întregului sistem. Controlerul preia setările furnizate prin dispozitivele de intrare și elementele de control și execută comenzile care pun structura hardware în modul de operare dorit. Totodată controlerul inițializează funcționarea formatoarelor de fascicul la emisie și recepție, a procesoarelor de semnal, a dispozitivelor de afișare și ieșire. Interacțiunile complicate dintre procesoarele din sistem necesită un protocol de comunicații de date robust și de viteză mare.

Interfața cu procesul (Front end)

Această secțiune din interiorul sistemului reprezintă poarta semnalelor spre și dinspre elementul traductor selectat. Sub controlul microprocesorului responsabil cu generarea și formarea la emisie, semnalele de excitație sunt aplicare elementelor traductoare de către blocurile emițătoare. Semnalele ecou din mediu sunt recepționate de către elementele ariei și aplicate la intrările amplificatoarelor cu câștig variabil pentru compensarea atenuării în mediu și a difracției pe durata propagării. Semnalele rezultate sunt trimise apoi la formatorul receptor.

Dispozitivul de baleiere (Scanner)

Conține acele componente de pe calea de semnal care asigură funcția fundamentală de compunere coerentă a semnalelor livrate de fiecare element traductor pentru a crea fiecare linie a imaginii. Formatorul de emisie trimite pulsurile electrice spre elemente în secvența temporală necesară. Semnalele ecou convertite digital sunt dirijate spre formatorul receptor pentru sinteza digitală de fascicul. În plus dispozitivul de scanare realizează un prim nivel de procesare de semnal care include filtrarea, obținerea semnalelor în quadratură, extragerea informației de mișcare pentru maparea curgerii sangvine.

Procesarea Post Scanare (Back end)

Această grupare de funcții este asociată cu formarea (sinteza), afișarea și evaluarea cantitativa a imaginii. Intrarea acestei secțiuni este un set de anvelope de semnale ecou extrase din liniile sintetizate la nivelul formatorului receptor pe baza semnalele de RF returnate de traductoare. Formarea imaginii este asigurată prin convertirea acestor semnale într-un convertor de format de scanare (scanconvertor) pentru a se obține date în formatul afișabil pe video monitor sau PC. Pe calea de semnal, în acest bloc sunt realizate o serie de operații de postprocesare, compresie logaritmică, mapare color sau pe scara de gri. Peste imaginea de bază se pot suprapune imagini conținând caractere alfanumerice sau alte informații și se pot rula anumite programe de măsură pentru evaluări dimensionale la nivelul regiunii explorate. Sunt de asemenea prezente elemente de control virtuale pentru modificarea formatului informației afișate.

Dispozitivul de scanare este cel mai important bloc al arhitecturii generice reprezentând „inima” sistemului [Thomenius 98, 05]. Evident este și cel mai scump bloc din sistem, cu o pondere de 30 -50% din valoarea totală a componentelor și din volumul de muncă.

Formatorul emițător are sarcina relativ simplă de a genera secvențele de întârziere la emisie. Aceste secvențe pot fi ușor implementate într-un ASIC ca un set de numărătoare presetabile

Formarea la recepție presupune sarcini mult mai complexe ca:

- generarea întârzierilor pentru:
 - o Dirijarea fasciculului
 - o Focalizarea dinamică
- ponderarea dinamică a semnalelor recepționate

și care se regăsesc arhitecturi de implementare de evident mai complicate.

Plecând de la aceste constatări formatoarele de fascicul vor fi abordate pe parcursul capitolelor următoare atât în ceea ce privește fundamentele teoretice cât și posibile soluții de implementare. Orice dezvoltare pe această direcție este de așteptat să conducă într-un viitor apropiat la optimizarea raportului preț - performanță pentru sistemele imagistice ultrasonice

1.4 SUPORTUL TEHNOLOGIC

Deși adesea progresul imagisticii ultrasonice este analizat individual corelația cu disponibilitățile tehnologice trebuie avută în vedere în permanență întrucât progresele rapide în electronică, tehnica de calcul și tehnologia materialelor au modelat în fiecare etapă “viitorul” imagisticii ultrasonice. Alături de Radar și Sonar, Imagistica Ultrasonica a beneficiat de punerea la punct a suportului de tehnologie electronica pentru scanarea și focalizarea ariilor cu comandă prin fază utilizate pentru aplicațiile în câmp electromagnetic. Pe seama acestor realizări s-au introdus ulterior metodele de scanare electronică liniară și sectorială.

Descoperirea tranzistorului și realizarea calculatorului digital la sfârșitul anilor 40 a declanșat cum era firesc o serie de modificări în conceperea sistemelor ultrasonice al căror ritm a fost accelerat la sfârșitul anilor 50 o data cu realizarea primului circuit integrat. Predicția Moore privind creșterea exponențială a densității circuitelor integrate si-a găsit confirmarea în creșterea vitezei de miniaturizare și a puterii de calcul integrate pe chip. Consecința firească a fost apariția pe piață a calculatoarelor personale, a memoriilor digitale și a circuitelor programabile. Sistemele imagistice ultrasonice în Mod B static, cu braț de scanare, apărute la sfârșitul anilor 60, încorporau parțial elementele indicate.

Apariția la începutul anilor 80 a sistemelor imagistice cu arii fazate a fost posibilă datorită ultimelor dezvoltări în tehnica video, microprocesoare, memorii digitale, linii de întârziere și a miniaturizării oferite de circuitele integrate programabile. Performanta acestor sisteme a fost rapid îmbunătățită în anii 80 de apariția circuitelor integrate dedicate (ASIC), a

procesoarelor de semnal (DSP) și de proiectarea asistată de calculator (CAD) a circuitelor integrate pe scara largă (VLSI) astfel ca la sfârșitul acestei decade sistemul cu arie fazată a devenit unealta imagistică dominantă. Actualmente miniaturizarea accelerată și în special utilizarea circuitelor integrate pe scara largă dedicate – ASIC-uri, face posibilă realizarea de sisteme imagistice care folosesc arii cu capacități de înaltă calitate a imaginii.

Când a apărut primul sistem cu arie fazată acesta cântărea sute de kg. Predicția creșterii densității tranzistoarelor conform legii Moore indica un factor de reducere a ariei de 1000000 în intervalul 1980 - 2000. Între timp panta curbei Moore a suferit o anumită scădere astfel ca actualmente factorul de reducere a ariei este de 1290. Aceasta reflectă limitele fizice ale tehnologiei CMOS dar și costurile ridicate reclamate de miniaturizarea extremă. [Brenner01] Deși nu se poate evalua un factor de miniaturizare a sistemelor imagistice ultrasonice, faptul ca un astfel de sistem din generația 2003 are mult mai multe facilități decât sistemele cu arii fazate din prima generație și cântărește doar câteva kilograme este grăitor.

Realizări remarcabile de dată recentă sunt capul de scanare cu arie bidimensională și formator încorporat și sistemele portabile care includ astfel de capete de scanare. Datorită disponibilității sistemele portabile pot fi folosite ca dispozitive de triere în clinicile mici sau în locuri în care costul unui sistem imagistic este prohibitiv. Sistemele ultrasonice moderne se definesc printr-o densitate mult mai mare de componente și o putere de calcul cu mult sporită față de predecesorii lor. Pe viitor sunt de așteptat modificări pe direcția creșterii complexității și a reducerii costurilor. Suportul tehnologic și punctele de cotitură în dezvoltarea investigării ultrasonice au fost concentrate în tabelul 1.1

Tabelul 1.1 Cronologia dependentei tehnologice a tehnicilor de investigare ultrasonică

Perioada	Ultrasunete	Suport Tehnologic
anii '30	Detectia ecourilor	Piezoelectricitate Tuburi amplificatoare cu vid
anii '40	Imaginea creierului (Dussik) Imagini PPI (Position Plan Indicator) Terapie și chirurgie	Radar, Sonar Reflectoscop supersonic Calculatoare (ENIAC, Collossus) Tranzistorul
anii '50	Modul A de scanare Scanare compusă Efectul Doppler în ultrasunete Scanare în modul M-miscare	Circuite integrate Comanda în fază a ariilor de antene
anii '60	Scanarea statică în mod B Scanarea mecanică în Timp Real Echoencefalografia	Legea lui Moore Microprocesorul, VLSI Calculatorul portabil
anii '70	Imagistica în Timp Real Conversia formatului de scanare Scara de gri Arii liniare și arii comandate prin faza	Memorii: RAM, EPROM ASIC Calculatoare Stiintifice Primul PC (Altair)
anii '80	Sisteme comerciale bazate pe arii de traductoare Sisteme Doppler în impulsuri Imagistica color a curgerii sangvine Traductoare specializate sau de banda larga	Arii de porti Procesoare de semnal Tehnologia SMD Proiectarea asistata de calculator a circuitelor VLSI
anii '90	Sisteme digitale Arii 1.5D și arii matriciale Imagistica armonica Commercializarea sistemelor imagistice 3D	Convertoare A/D-solutii ieftine Powerful PCs Procesoare de imagine - 3D Tehnologii nanometrice
anii 2000	Sisteme portabile cu arii bidimensionale pentru imagistica 3D în Timp Real	Miniaturizarea

Cronologia expusă în tabel evidențiază un decalaj între apariția tehnologiei și manifestarea efectelor sale la nivelul sistemelor de investigare. Cele mai dramatice schimbări au apărut pe direcția miniaturizării în conformitate cu legea Moore. Existența decalajului menționat, conduce la concluzia ca ultimele dezvoltări tehnologice nu și-au manifestat în totalitate impactul asupra sistemelor imagistice. Potențialul de diagnostic prezis inițial imagisticii ultrasonice de către pionierii domeniului a fost atins și depășit. Combinarea progresului continuu în electronică, cu o mai bună înțelegere a interacțiunii ultrasunetelor cu mediul va conduce la sisteme imagistice de complexitate sporită. Este de așteptat ca în viitor principiile simple care stau la baza majorității sistemelor imagistice să fie înlocuite cu algoritmi sofisticăți de procesare de semnal.

1.5 COMPARAȚIA CU ALTE TEHNICI IMAGISTICE MEDICALE

Secolul XX marcat de incredibilul avânt al descoperirilor tehnice pune la îndemâna medicinei o aparatură operațională din ce în ce mai diversă și mai perfecționată. Datorită eficacității și costurilor scăzute imagistica ultrasonică este la ora actuală tehnica imagistică preferată. S-a estimat că în anul 2000 la nivel mondial au fost efectuate săptămânal 5 milioane de investigații cu ultrasunete (figura 1.10). O succintă trecere în revistă a tehnicilor imagistice moderne va permite evidențierea locului și rolului imagisticii ultrasonice ca metodă paraclinică de bază în stabilirea diagnosticului.

Radiografia

Este cea mai veche dintre tehnicile imagisticii medicale și are la bază absorbția selectivă a radiațiilor X de către diverse structuri ale corpului uman. Ea furnizează informații despre starea fiziologică, localizarea și structura diferitelor formațiuni. Este o metodă ieftină, cu o bună rezoluție a determinării dar prezintă dezavantajul factorilor de risc asociați iradierii cu raze X, fapt ce conduce la o repetabilitate a expunerii la minim 6 luni. În plus nu permite evaluarea stării de mișcare.

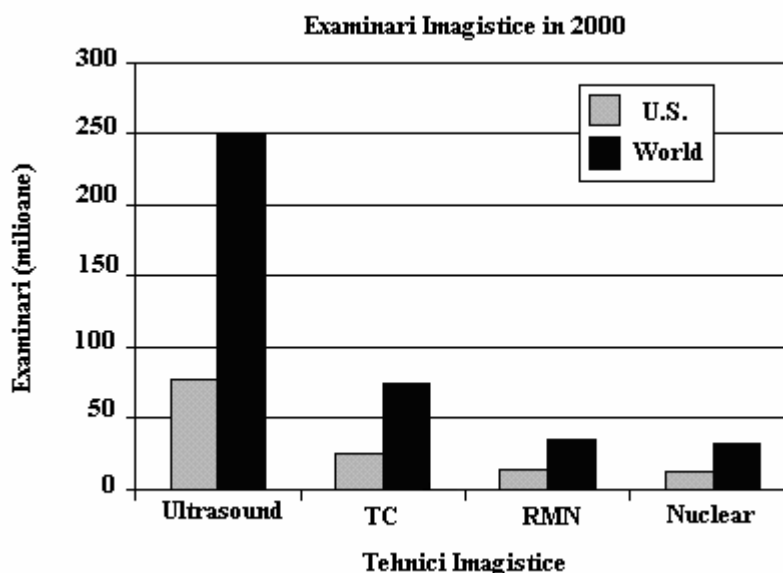


Figura 1.10 Statistica utilizării tehnicilor imagistice în anul 2000 [Cote01]

Tomografia computerizată(TC).

Absorbția razelor X în fiecare punct al unei secțiuni transversale a corpului poate fi calculată prin măsurarea mai multor fascicule de raze X care cad sub unghiuri diferite în

cadrul secțiunii transversale. Coeficienții de absorbție calculați sunt reprezentați ca densități radiografice prin mult mai multe nuanțe de gri decât este posibil printr-o radiografie obișnuită. Tomografia computerizată a fost dezvoltată la începutul anilor 70. Aceasta tehnica furnizează imagini ale țesuturilor moi, oaselor și vaselor de sânge. TC permite vizualizarea directă și diferențierea țesuturilor moi cum ar fi: ficatul, splina și țesutul adipos. TC este utilă în mod special pentru detecția leziunilor extinse, a tumorilor și metastazelor și permite evaluarea dimensională și localizarea spațială a acestora. TC a craniului și creierului poate detecta tumori, arată defecte ale vaselor de sânge, ventricule extinse cauzate de prezența lichidului cerebrospinal și diferite alte anomalități cum sunt cele ale mușchilor și nervilor oculari. Avantajele metodei sunt: o rezoluție de contrast foarte bună asociată cu o penetrabilitate bună, limitată, eventual, de un prag de iradiere. Ca dezavantaje sunt de menționat expunerea la radiații dar și costurile ridicate ale aparaturii și investigației propriu-zise. Expunerea prelungită la radiație care poate avea efecte cancerigene în special la copii sub 15 ani.

Rezonanța magnetică nucleară (RMN)

Imagistica prin rezonanță magnetică a fost cercetată la începutul anilor 80, primul prototip a fost testat clinic pe pacienți în 1980 iar comercializarea acestor sisteme se face din 1984.

Rezonanța magnetică nucleară are la bază o proprietate a nucleelor atomilor cu un număr impar de protoni, din care în organism cele mai numeroase sunt nucleele de hidrogen. În interiorul unui câmp magnetic foarte puternic, spinul nucleelor este orientat în direcția câmpului, prin suprapunerea unui semnal de radio frecvență, alinierea acestor nuclee se poate modifica iar viteza cu care nucleele revin la orientarea inițială (relaxarea) se poate măsura pe baza unui semnal. Există două tipuri de relaxare care depind de natura chimică și învelișul atomic al fiecărui nucleu, fapt ce face ca acestea să difere de la țesut la țesut.

Pe baza gradientului se pot măsura vitezele de relaxare, simultan în mai multe puncte din spațiu, ceea ce permite afișarea lor sub forma unui set de imagini în tonuri de gri, în oricare din planurile transversal, frontal sau sagital. Momentul suprapunerii semnalului de radiofrecvență determină care dintre viteze predomină și de ce aceea imaginile obținute au diferite nuanțe de gri pentru același țesut deși structura anatomică nu s-a schimbat. Secțiunile transversale RM seamănă foarte mult cu imaginile TC dar flexibilitatea în alegerea și folosirea celor două viteze de relaxare conferă RMN un contrast mai bun decât TC.

Avantajele metodei sunt: o rezoluție de contrast excepțională între țesuturile normale și cele patologice, lipsa radiațiilor ionizante și capacitatea de a obține imagini în oricare din planurile transversal, frontal sau sagital. RMN este o unealtă deosebit de eficientă în identificarea și diagnosticarea multor forme de cancer.

Dezavantajele metodei:

- Procedura nu poate fi aplicată persoanelor care au implantate stimulatoare cardiace sau care suferă de claustrofobie
- rata scăzută de înflorărire, conduce la imagini cu contur neclar datorită mișcărilor respiratorii și peristaltice
- Sistemele sunt foarte zgomotoase
- Pacientul trebuie să rămână liniștit o lungă perioadă de timp
- Prezența unui câmp magnetic puternic care face imposibilă investigarea persoanelor cu stimulatoare cardiace sau dispozitive metalice ortopedice (șuruburi, plăci, joncțiuni) Prezența acestora în aria scanată poate cauza distorsiuni majore ale imaginii
- Sistemele pentru MR sunt foarte scumpe și prin urmare și procedurile de investigare sunt foarte costisitoare

Imagistica ultrasonică

Ultrasunetele au deschis o nișă în câmpul imagisticii medicale, concurând sistemele imagistice cu raze X/CT și rezonanță magnetică (RMN). Spre deosebire de tehnicile cu raze

X/CT, imagistica ultrasonică nu expune pacientul radiațiilor ionizante, ceea ce constituie un avantaj important pentru examinările pediatrice și obstetrice și prezintă în avantajul unui preț relativ scăzut, portabilității și siguranței. Pe de altă parte rezonanța magnetică asigură calitate rafinată imaginii în timp non-real, fără radiație ionizată, dar cu un cost final foarte mare și timpi de achiziție lungi. Ultrasunetele se potrivesc bine aplicațiilor care cer structuri de imagini în timp real, unde costul și siguranța sunt de asemenea caracteristici semnificative.

În modul B de scanare cu ultrasunete, ecourile sunt afișate color sau sub forma unor nuanțe de gri permițând reproducerea de imagini în secțiuni transversale, longitudinale sau oblice ale organelor, în funcție de orientarea transductorului. În cazul scanării în timp real imaginea afișată se modifică în mod continuu permițând astfel studierea structurilor anatomice și observarea mișcărilor fiziologice ale țesuturilor. Ecografia Doppler ca variantă a ecografiei bazată pe folosirea efectului Doppler și a tehnicilor corelative este singura dintre metodele imagistice medicale care detectează fluxul sanguin și poate genera imaginea curgerii suprapusa peste imaginea în mod B.

Metodele care folosesc ultrasunetele sunt excelente pentru investigarea și diagnosticarea neinvazivă a unui număr important de organe. Obstetrica modernă – urmărirea sarcinii și a nașterii – se bazează în principal pe tehnica ultrasunetelor pentru a furniza imagini detaliate ale fătului și ale uterului. Imagistica ultrasonora poate arata dezvoltarea fătului și diferite funcții ale corpului cum ar fi respirația și mișcarea. Ultrasunetele sunt de asemenea larg utilizate pentru evaluarea rinichilor, ficatului, pancreasului, a inimii și vaselor de sânge din abdomen și din zona gâtului. Pot fi de asemenea folosite pentru ghidare la recoltarea biopsiilor pentru testele de laborator. Recent ultrasunetele sunt folosite pe scara largă pentru investigarea sânelor și pentru ghidare la prelevarea biopsiei în cancerul de sân

Cea mai mare dificultate la ultrasunete este prezența unei pete în imagine care umbrește structura și face interpretarea greu de realizat. De asemenea, ultrasunetele sunt limitate la acele aplicații care prezintă ferestre acustice adecvate pentru a aborda țesutul moale intern, deoarece undele sonore nu se propagă bine prin oase și pungi de aer. Cu aceste limitări, ultrasunetele reprezintă o tehnică de imagistică importantă pentru observarea inimii, a sistemului vascular periferic și central și de asemenea pentru țesutul moale abdominal și pelvian.

În tabelul 1.2.se prezintă o evaluare comparativă a principalelor tehnici imagistice medicale pe baza unui set de criterii care cupinde:

- Rezoluția spațială
- Timpul de investigare
- Accesibilitatea
- Riscuri asociate metodei – prezența radiației
- Portabilitate
- Penetrabilitate
- costurile aparaturii și ale procedurilor de investigare

Informațiile prezentate succint evidențiază locul și rolul distinct al diagnosticului ultrasonor în ansamblul metodelor paraclinice de diagnostic caracterizat prin:

- Competitivitatea cu alte metode în ceea ce privește informația obținută (rezoluție)
- Timpul redus de investigare (Scanare în timp real)
- Detecția mișcărilor fiziologice, a vitezei și debitului sangvin
- Păstrarea strictă sub limitele admise a dozei de iradiere ultrasonică
- Competitivitate în ceea ce privește costurile aparaturii și ale procedurilor de investigare.

Aceste caracteristici definesc în fapt direcțiile de cercetare pentru perfecționarea procedurilor de diagnostic ultrasonor și a aparaturii aferente acestuia.

Tabel 1.2 Tehnici imagistice medicale

METODA CRITERII		RADIOGRAFIA	ECOGRAFIA	TC	RMN
Mărimea măsurată		Absorbția medie în țesuturi	Proprietăți mecanice	Absorbția în țesuturi	Proprietăți biochimice (timp de relaxare)
Rezoluție spațială		~1mm	Dependentă de frecvență și orientare 0.3 - 3mm	~1mm	~1mm
Timp de investigare		minute	100 cadre/sec	30sec - minute	10 cadre/sec
Acces		Necesită acces din două părți	Prin ferestre acustice adecvate	Pe circumferință	Pe circumferință
Risc asociat (Iradiere)		DA	NU	DA	Nu
Costuri: aparatura + procedura de investigare		\$	\$	\$\$\$\$	\$\$\$\$\$\$\$\$
Portabilitate		Bună	Excelentă	Slabă	Slabă
Repetabilitatea expunerii		6 luni	Oricând	După un Δt	Limitată de costuri
Penetrabilitate		Excelentă	Dependentă de frecvență 3-25 cm	Excelentă	Excelentă
Domenii de utilizare	Pulmonar	X		X	X
	Cardiovascular	X	X	X	X
	Hepatobiliar	X	X		X
	Obstetrică		X		
	Ginecologie		X		
	Oftalmologie		X		
	Sistem nervos			X	X

1.6. CONCLUZII

Capitolul prezintă pe scurt principalele elemente privind evoluția structurală a sistemelor de investigare ultrasonică pentru atingerea obiectivelor unei investigații performante și locul distinct al investigației ultrasonice în cadrul metodelor imagistice medicale. Obiectivul principal a fost acela de a identifica în structura sistemelor imagistice ultrasonice blocurile funcționale care definesc performanța sistemului pentru a direcționa efortul de cercetare spre perfecționarea structurii acestora.

Pe baza unui studiu bibliografic extins în paragraful 1.2 s-a prezentat în sinteză evoluția sistemelor imagistice medicale de la introducerea lor până în prezent. Se evidențiază o structură relativ unitară la interfața cu traductorul și creșterea complexității sistemului prin introducerea mecanismelor de scanare mecanică și apoi electronică.

Introducerea traductoarelor multielement și perfecționarea tehnologiei digitale a permis trecerea la dirijarea și focalizarea electronică a fascicului ultrasonic. Structurile de procesare dezvoltate pentru traductoarele multielement au un anumit număr de canale independente, operează în modul B în timp real și afișează pe ecran un număr mare de imagini pe secundă. S-au impus două tipuri de scanare funcție de tipul de traductor folosit: arie liniară sau arie fazată.

Sistemele cu arie liniară folosesc în același timp mai multe canale adiacente la emisie cât și la recepție. Același set de întârzieri simetrice este utilizat pentru formarea tuturor liniilor de imagine, modificându-se doar elementele care participă la formarea liniilor succesive. În cazul ariilor fazate setul de întârzieri este univoc definit pentru fiecare linie de imagine care este formată prin aportul tuturor elementelor ariei. Se obține o imagine sectorială alcătuită dintr-un set de linii care converg în centrul ariei. Din punct de vedere structural folosirea ariilor de traductoare și a scanării electronice a condus la apariția în structura dispozitivului de scanare a blocurilor funcționale care sintetizează liniile de imagine.

Structura sistemelor ultrasonice moderne a fost prezentată în paragraful 1.3 cu scopul de a identifica locul și rolul dispozitivului de scanare. Este unanim acceptat [Thomenius 98, 05] că dispozitivul de scanare este cel mai important bloc al arhitecturii generice reprezentând „inima” sistemului. În același timp este și cel mai scump bloc din sistem realizarea sa necesitând 30 -50% din totalul componentelor și din volumul de muncă.

În timp ce formatorul emițător are o sarcină relativ simplă de a genera secvențele de întârziere la emisie formatorul receptor are sarcini mult mai importante:

- generarea întârzierilor pentru:
 - Dirijarea fasciculului
 - Focalizarea dinamică
- Ponderarea dinamică a semnalelor recepționate

Plecând de la aceste constatări formatoarele de fascicul vor fi abordate pe parcursul capitolelor următoare atât în ceea ce privește fundamentele teoretice cât și posibile soluții de implementare. Orice dezvoltare pe această direcție este de așteptat să conducă într-un viitor apropiat la optimizarea raportului preț - performanță pentru sistemele imagistice ultrasonice.

Progresul imagisticii ultrasonice trebuie analizat în corelația cu disponibilitățile tehnologice întrucât progresele rapide în electronica, tehnica de calcul și tehnologia materialelor au modelat în fiecare etapă “viitorul” imagisticii ultrasonice. Suportul tehnologic și punctele de cotitură în dezvoltarea investigării ultrasonice au făcut obiectul paragrafului 1.4. Realizările remarcabile de dată recentă: capul de scanare cu arie bidimensională și formator încorporat și sistemele portabile care includ astfel de capete de scanare confirmă concluziile referitoare la importanța formatoarelor de fascicul.

Cronologia expusă evidențiază un decalaj între apariția tehnologiei și manifestarea efectelor sale la nivelul sistemelor de investigare. Cele mai dramatice schimbări au apărut pe direcția miniaturizării în conformitate cu legea Moore. Existența decalajului menționat conduce la concluzia că ultimele dezvoltări tehnologice nu și-au manifestat în totalitate impactul asupra sistemelor imagistice. Potențialul de diagnostic prezis inițial imagisticii ultrasonice de către pionierii domeniului a fost atins și depășit. Combinarea progresului continuu în electronică, cu o mai bună înțelegere a interacțiunii ultrasunetelor cu mediul va conduce la sisteme imagistice de complexitate sporită. Este de așteptat că în viitor principiile simple care stau la baza majorității sistemelor imagistice să fie înlocuite cu algoritmi sofisticăți de procesare de semnal.

O succintă trecere în revistă a tehnicilor imagistice moderne a permis în finalul capitolului evidențierea locului și rolului imagisticii ultrasonice ca metodă paraclinică de bază în stabilirea diagnosticului. Informațiile prezentate evidențiază locul și rolul distinct al diagnosticului ultrasonor în ansamblul metodelor paraclinice de diagnostic caracterizat prin competitivitate în ceea ce privește rezoluția, costurile, timpul de investigare și riscul de iradiere. Aceste caracteristici definesc practic direcțiile de cercetare pentru perfecționarea imagisticii ultrasonice și a aparaturii aferente acestora.

CAPITOLUL2

FORMATOARE DE FASCICUL

Acest capitol prezintă într-o abordare unitară fundamentele formării fasciculelor și evoluția tehnicilor de formare. Sinteza de fascicul este un proces important în sistemele imagistice deoarece calitatea fasciculului determină rezoluția laterală și frecvența cadrelor. Evoluția tehnicilor de formare a fasciculelor ultrasonice este determinată de necesitatea îmbunătățirii calității imaginilor rezultate în urma scanării ultrasonice.

Pornind de la cele afirmate mai sus sunt prezentate bazele matematice ale formării fasciculelor în general și a celor ultrasonice în special. Se realizează apoi o evaluare a tehnicilor analogice și digitale de formare a fasciculelor în scopul identificării posibilităților de optimizare sub raport complexitate performanță

2.1 ECUAȚIILE GENERALE PENTRU SINTEZA FASCICULELOR ULTRASONICE

Formarea sau sinteza fasciculelor este o metodă de observare a semnalelor recepționate de o arie de traductoare de pe o direcție dorită simultan cu atenuarea răspunsului ariei pentru semnalele provenite de pe alte direcții. Procedura de sinteză permite o vizualizare multidimensională a mediului prin folosirea unei arii de traductoare adecvate și are multiple aplicații în medicină, astronomie și echipamentele militare. [Johnson93]

Se consideră pentru început aria tridimensională localizată prin vectorul de poziție $\mathbf{r}_m$, unde n este indexul elementului curent al ariei, care recepționează semnalul $x(t)$. Direcția semnalului recepționat $x(t)$ este specificată de vectorul unitate $\mathbf{u}$, în timp ce aria va fi dirijată pe direcția descrisă de vectorul unitate $\mathbf{u}_0$. Orientarea vectorilor direcționali ($\mathbf{u}$, $\mathbf{u}_0$) este definită de două unghiuri numite azimut (ϕ , ϕ_0) și elevație (θ , θ_0).

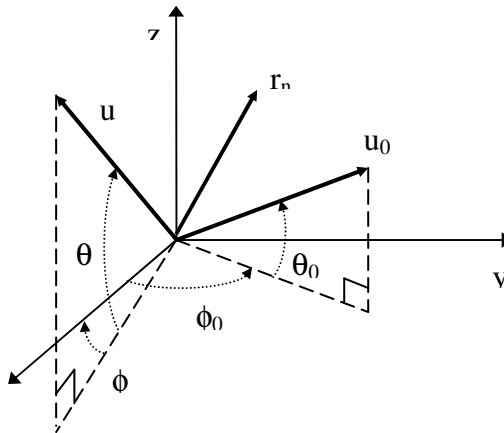


Fig. 2.1 Sistemul de coordonate tridimensional pentru formarea fasciculelor în câmp îndepărtat

2.1.1 CAZUL CÂMPULUI ÎNDEPĂRTAT

Semnalul complex $x(t)$ este în general un puls sinusoidal modulat care are informația spațială înscrisă în urma reflexiilor pe *interfețele și obiectele* care vor fi detectate și localizate. Se identifică prin denumirea de *interfață acustică* suprafața de separare a două medii cu impedențe acustice diferite.

$$x(t) = e^{j\omega t} \quad (2.1)$$

Semnalul se propagă prin mediu cu viteza c sub forma unei unde plane de pulsație ω , iar în cele ce urmează se va lucra în ipoteza câmpului îndepărtat pentru care distanța până la sursă se presupune suficient de mare astfel încât fronturile de undă sunt paralele cu aria. Faza undelor recepționate conține informația despre direcția spațială a sursei. Astfel semnalul recepționat de elementul n este:

$$x_n(t) = e^{j(\omega t + k r_n^T \cdot \mathbf{u})} = x(t) \cdot e^{j k r_n} \text{ pentru } n = 0, \dots, N-1 \quad (2.2)$$

$k = \omega/c$ este *numărul de undă* asociat undei, iar $r_n = \mathbf{r}_n \cdot \mathbf{u}$ proiecția vectorului $\mathbf{r}_n$ pe direcția $\mathbf{u}$ care indică distanța adițională pe care unda care sosește de pe direcția $\mathbf{u}$ o are de parcurs până la elementul localizat prin $\mathbf{r}_n$. Din figura 2.1 se pot exprima coordonatele vectorului direcțional $\mathbf{u}$ și ale vectorului de poziție al elementului n .

$$\vec{u} = \begin{bmatrix} \cos \phi \cos \theta \\ \sin \phi \cos \theta \\ \sin \theta \end{bmatrix}, \quad \vec{r}_n = \begin{bmatrix} r_{xn} \\ r_{yn} \\ r_{zn} \end{bmatrix} \quad (2.3)$$

Prin urmare proiecția vectorului de poziție al elementului m , $\mathbf{r}_n$ pe direcția $\mathbf{u}$ este:

$$r_n = \vec{r}_n^T \cdot \vec{u} = r_{xn} \cos \phi \cos \theta + r_{yn} \sin \phi \cos \theta + r_{zn} \sin \theta \quad (2.4)$$

Rezultatul însumării ieșirilor celor N elemente ale ariei prezintă caracteristicile unui filtru spațial. Într-un caz general filtrarea spațială este echivalentă cu un filtru de medie alunecătoare sau filtru FIR (pentru o arie liniară uniformă), pentru care fasciculul de ieșire este dat de:

$$y(t) = a \cdot X(t) = x(t) \cdot a \cdot e^{j k R} \quad (2.5)$$

unde a definește o fereastră de ponderare, $X(t)$ conține ieșirile tuturor elementelor iar R conține defazajul corespunzător fiecărui senzor:

$$\begin{aligned} a &= [a_0 \dots a_{N-1}] \\ X(t) &= [x_0(t) \dots x_{N-1}(t)]^T = x(t) \cdot e^{j k R} \\ R &= [r_0 \dots r_{N-1}]^T \end{aligned} \quad (2.6)$$

Fereastra de pondere, specificată prin vectorul a al coeficienților filtrului se consideră arbitrară în acest punct dar va fi folosită una dintre ferestrele cunoscute (Hamming, Bartlett, Rectangulară etc.) pentru a controla lățimea lobului principal și amplitudinea lobilor laterali. Dirijarea sau filtrarea spațială pe o anumită direcție $\mathbf{u}_0$ este asigurată prin însumarea coerentă pentru această direcție a ieșirilor elementelor ariei. Aceasta necesită mai întâi determinarea proiecției vectorului de poziție a elementului curent $\mathbf{r}_n$ pe vectorul $(\mathbf{u} - \mathbf{u}_0)$:

$$\begin{aligned} \hat{r}_n = \vec{r}_n^T \cdot (\mathbf{u} - \mathbf{u}_0) &= \begin{bmatrix} r_{xn} & r_{yn} & r_{zn} \end{bmatrix} \begin{bmatrix} \cos \phi \cos \theta - \cos \phi_0 \cos \theta_0 \\ \sin \phi \cos \theta - \sin \phi_0 \cos \theta_0 \\ \sin \theta - \sin \theta_0 \end{bmatrix} \\ R - R_0 &= [\hat{r}_0 \dots \hat{r}_{N-1}]^T \end{aligned} \quad (2.7)$$

Rezultă de aici ecuația generală pentru sinteza de fascicul:

$$y(t, u_0) = x(t) \cdot a \cdot e^{j k (R - R_0)} = x(t) \cdot H(\omega, u_0) \quad (2.8)$$

unde directivitatea fasciculului este dată de:

$$H(\omega, u_0) = a \cdot e^{j k (R - R_0)} \quad (2.9)$$

și definește caracteristicile spațiale ale ariei. Procesul de decalare temporală cu R_0 a ieșirii fiecărui element conduce la însumarea coerentă când $\mathbf{u} = \mathbf{u}_0$ și este denumit formarea sau dirijarea unui fascicul pe direcția $\mathbf{u}_0$.

În cazul general geometria ariei este arbitrară, totuși pentru simplificarea analizei și a implementării se utilizează forme regulate. Cazul tratat va fi cel al ariei liniare.

2.1.2 GEOMETRIA ARIEI LINIARE

O geometrie frecvent utilizată este aria liniară, care poate produce o imagine bidimensională polară. Fiecare punct al imaginii este definit printr-o pereche de parametrii distanță (moment prelevare eșantion) – unghi (direcție de dirijare). Aria liniară conține N elemente dispuse uniform cu o distanță d între elemente.

Considerând aria coliniară cu axa x , vectorul care localizează elementul n al ariei este:

$$\mathbf{r}_n = [nd \ 0 \ 0]^T$$

Proiecția acestuia pe vectorul $\mathbf{u}$ va fi:

$$r_n = \hat{\mathbf{r}}_n \cdot \hat{\mathbf{u}} = nd \cos \phi \cos \theta$$

În cazul spațiului bidimensional elevația este nulă ($\theta = 0$) și cu substituția $\phi = \pi/2 - \psi$ se obține în final vectorul ieșirilor elementelor ariei:

$$\mathbf{X}(t) = \mathbf{x}(t) \cdot e^{jkR} = \mathbf{x}(t) \cdot e^{jknd \cos \phi \cos \theta} = \mathbf{x}(t) \cdot e^{jknd \sin \psi}, \text{ unde } n = [0 \dots N-1]^T \quad (2.10)$$

Substituția $\phi = \pi/2 - \psi$ introduce unghiul ψ format de direcția de propagare a unei plane (perpendiculară pe suprafața traductorului) cu direcția de observare în plan. Pentru cazul general al unei direcții de dirijare ψ_0 , proiecția $R - R_0$, este prin definiție:

$$\hat{\mathbf{r}}_n = \mathbf{r}_n^T \cdot (\mathbf{u} - \mathbf{u}_0) = [nd \ 0 \ 0] \begin{bmatrix} \sin \psi - \sin \psi_0 \\ 0 \\ 0 \end{bmatrix} = nd(\sin \psi - \sin \psi_0) \quad (2.11)$$

$$\mathbf{R} - \mathbf{R}_0 = [\hat{r}_0 \dots \hat{r}_{n-1}]^T = nd(\sin \psi - \sin \psi_0)$$

și substituind acest rezultat în ecuația generală pentru sinteza de fascicul obținem ecuația ce descrie formatorul (sintetizorul) de fascicul cu arie liniară:

$$y(t, \psi_0) = \mathbf{x}(t) \cdot \mathbf{a} \cdot e^{jknd(\sin \psi - \sin \psi_0)} = \mathbf{a} \cdot \mathbf{X}(t) \cdot e^{-jknd \sin \psi_0} \quad (2.12)$$

Forma finală a ecuației este una implementabilă prin însumarea ieșirilor elementelor ariei, defazate în prealabil după un algoritm care maximizează rezultatul însumării pentru o direcție dată ψ_0 .

2.1.3 CAZUL CÂMPULUI APROPIAT

În situația observării în câmp apropiat, frontul de undă nu mai este plan și curbura sa introduce o întârziere suplimentară care trebuie compensată pentru a asigura însumarea coerentă. În acest caz trebuie cunoscută atât direcția pe care se află sursa cât și distanța la sursă (fig. 2.2).

Aria este localizată prin vectorul de poziție $\mathbf{r}_n$, unde n este indexul elementului curent al ariei. Sursa este specificată prin vectorul $\mathbf{r}$ de versor $\mathbf{u}$, iar aria va fi dirijată pe direcția descrisă de vectorul unitate $\mathbf{u}_0$. Orientarea vectorilor direcționali ($\mathbf{u}$, $\mathbf{u}_0$) este definită prin azimut (ϕ , ϕ_0) și elevație (θ , θ_0).

Distanța pe calea de propagare directă de la sursă la senzorul n este: $p_n = \|\mathbf{r} - \mathbf{r}_n\|$

unde norma euclidiană $\|\bullet\|$, este definită ca: $\|x\| = (x^T x)^{1/2} = \left(\sum_{i=1}^n x^2(i) \right)^{1/2}$

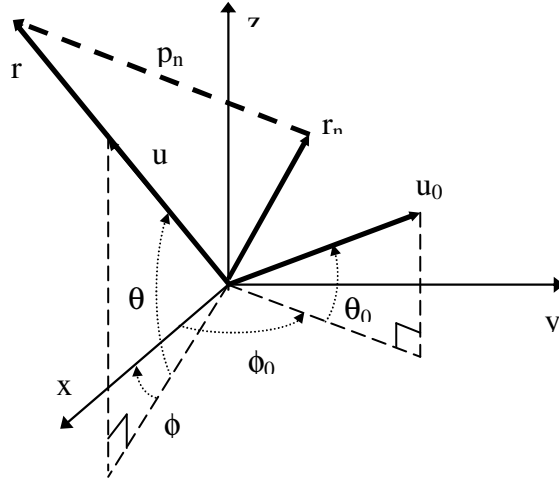


Fig. 2.2 Sistemul de coordonate tridimensional pentru formarea fasciculelor în câmp apropiat

Prin dezvoltare se obține expresia pentru p_n :

$$\begin{aligned} p_n^2 &= \|r - r_n\|^2 = (r^T - r_n^T)(r - r_n) = \|r\|^2 - 2r_n^T r + \|r_n\|^2 = \|r\|^2 \left(1 - \frac{2r_n^T r}{\|r\|^2} + \frac{\|r_n\|^2}{\|r\|^2} \right) \\ p_n &= \|r\| \left(1 - \frac{2r_n^T r}{\|r\|^2} + \frac{\|r_n\|^2}{2\|r\|^2} \right)^{1/2} \end{aligned} \quad (2.13)$$

Astfel semnalul recepționat de elementul n este:

$$x_n(t) = e^{j(\omega t + k p_n)} = x(t) \cdot e^{j k p_n} \text{ pentru } n = 0, \dots, N-1 \quad (2.14)$$

În spațiul bidimensional se consideră că aria este coliniară cu axa x , vectorul care localizează elementul m al ariei fiind: $r_n = [nd \ 0 \ 0]^T$, iar vectorul de poziție al sursei formează unghiul ψ cu direcția de propagare a unei plane și este de modul R . În această situație întârzierea spațială are expresia:

$$p_n = R \left(1 - \frac{2nd \sin \psi}{R} + \frac{(nd)^2}{R^2} \right)^{1/2} \quad (2.15)$$

Ecuția ce descrie formatorul de fascicul cu arie liniară pentru un punct de coordonate (R, ψ) din câmpul apropiat este:

$$y(t, R, \psi_0) = a \cdot X(t) \cdot e^{-j k p_n(R, \psi_0)} \quad (2.16)$$

Cazul câmpului îndepărtat caracterizat de front de undă plan se întâlnește în aplicațiile RADAR în care se formează fascicule de unde electromagnetice și în SONAR unde se sintetizează fascicule de unde ultrasonice. Pentru aplicațiile în câmp apropiat, unde dimensiunea regiunii de interes este comparabilă cu dimensiunea ariei, frontul de undă este sferic și acest lucru trebuie luat în considerare pentru determinarea exactă a întârzierilor cerute de însumarea coerentă. Acesta este cazul imagisticii medicale în care este necesară focalizarea fasciculelor pe obiecte într-un domeniu restrâns.

2.2 FORMATOARE DE FASCICUL ULTRASONIC TIP ÎNTÂRZIERE – ÎNSUMARE

Fasciculele de unde ultrasonice sunt sintetizate prin comanda electronică a ariilor de traductoare ultrasonice de diferite tipuri: liniare, circulare, rectangulare, curbate. Ecuațiile (2.12) și (2.16) descriu *formatoarele de fascicul de tip întârziere - însumare* care au la bază un algoritm dintre cele mai vechi și mai simple și care se dovedește și astăzi deosebit de util. Ideea de realizare a fasciculelor prin întârziere și însumare este simplă. Semnalul recepționat din mediu este adus la intrările formatorului de fascicul. Ieșirile formatorului, obținute prin întârzierea adecvată a intrărilor și însumarea lor, maximizează semnalul pentru o anumită direcție spre a nu fi afectat de zgomet sau de undele care se propagă pe alte direcții.

Considerăm aria liniară cu N elemente egal distanțate între ele cu un spațiu d pe direcția axei x . Originea sistemului de coordonate se află în centrul ariei. Poziția elementului n este : (figura 2.3)

$$x_n = \left(n - \frac{N-1}{2} \right) d \quad \text{pentru } 0 \leq n \leq N-1 \quad (2.17)$$

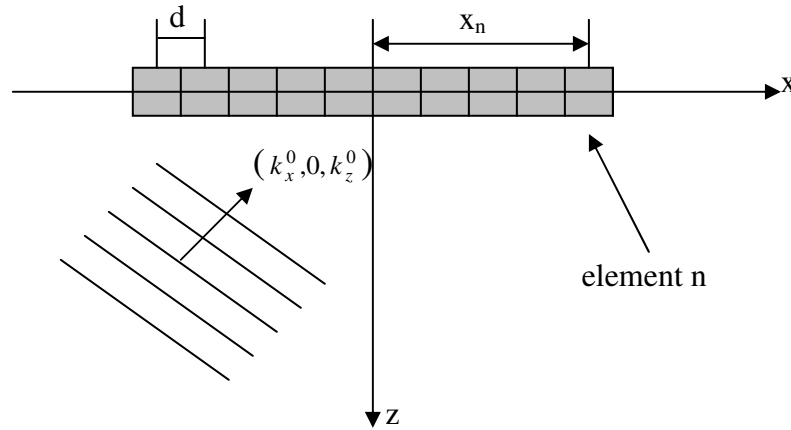


Figura 2.3. Aria liniară cu N elemente și spațiul inter-elemente d

Când observăm un câmp de unde $f(\vec{x}, t)$ cu aria liniară unidimensională ($\vec{x} = (nd, 0, 0)$), semnalul recepționat de elementul n al ariei este $y_n(t) = f(\vec{x}_n, t)$. Formatorul de fascicul de tip întârziere - însumare constă în aplicarea unei întârzieri Δ_n semnalului furnizat de fiecare element și însumarea semnalelor rezultate. Ieșirea formatorului de fascicul de tip întârziere - însumare este:

$$z(t) = \sum_{n=0}^{N-1} a_n y_n(t - \Delta_n)$$

Ponderea amplitudinii a_n este denumită uneori *apodizare* și poate lua orice valoare între 0 și 1.

Pentru un câmp de unde plane, cu frecvența ω și numărul de undă $\vec{k}$, observat cu aria unidimensională din fig 2.3 avem:

$$y_n(t) = e^{j(\omega t - kx_n \sin \theta)} \quad (2.18)$$

și

$$z(t) = \sum_{n=0}^{N-1} a_n e^{j(\omega(t - \Delta_n) - kx_n \sin \theta)} = e^{j\omega t} \sum_{n=0}^{N-1} a_n e^{-j(\omega \Delta_n + kx_n \sin \theta)}$$

Când $\Delta_n = -\frac{k}{\omega} x_n \sin \theta = -\alpha x_n \sin \theta = -\frac{1}{c} x_n \sin \theta$, adică întârzierea Δ_n este aleasă pentru a observa direcția de propagare a unei plane, ieșirea formatorului de fascicul atinge valoarea maximă egală cu $e^{j\omega t} \sum_{n=0}^{N-1} a_n$.

Putem dirija fasciculul ariei pe o anumite direcție de propagare prin folosirea unui set de întârzieri $\Delta_n = -\frac{1}{c} x_n \sin \theta^0$. În acest caz semnalul sintetizat pentru propagarea unei unde plane într-o direcție oarecare θ este dat de:

$$z(t) = e^{j\omega t} \sum_{n=0}^{N-1} a_n e^{jkx_n (\sin \theta^0 - \sin \theta)} \quad (2.19)$$

Răspunsul formatorului de fascicul de tip întârziere și însumare la o undă monocromatică este denumit în mod curent caracteristica ariei. În acest caz caracteristica ariei va fi:

$$H(\theta, \theta^0) = \sum_{n=0}^{N-1} a_n e^{jkx_n (\sin \theta^0 - \sin \theta)} \quad (2.20)$$

Caracteristica ariei determină diagrama de directivitate a ariei sau structura fasciculului. Structura fasciculului va lua aceeași formă ca a ecuației 2.20 cu un unghi de dirijare fix θ^0 și un unghi curent θ . Structura fasciculului propagat în câmp îndepărtat într-o singură direcție este:

$$W(\theta) = \sum_{n=0}^{N-1} w_n e^{jkx_n (\sin \theta^0 - \sin \theta)} \quad (2.21)$$

2.2.1. ÎNTÂRZIAREA PENTRU DIRIJARE ȘI FOCALIZARE

Un fascicul ultrasonic generat de o arie liniară fazată poate fi atât focalizat cât și dirijat (direcționat) prin întârzierea corespunzătoare a semnalelor emise și recepționate. În sistemele radar și sonar este necesară numai dirijarea undelor deoarece țintele se află în regiunea de câmp îndepărtat. (front de undă plan) În sistemele ultrasonice nu numai dirijarea dar și focalizarea este cerută, deoarece se lucrează în regiunea câmpului apropiat în care se propagă unde sferice.

Considerăm o arie comandată în fază de N elemente, cu distanța între elemente $d = \lambda/2$, care trebuie direcționate și focalizate într-un punct (r, θ) , unde θ este unghiul de dirijare a fasciculului iar r este distanța. Pentru simplificare alegem originea sistemului de coordonate în centrul ariei comandate în fază cum e arătat în figura 2.4.

Întârzierea într-un singur sens pentru elementul m este dată de:

$$\tau_n = (r - r'_n)/c \quad (2.22)$$

unde r'_n este distanța dintre elementul n și punctul (r, θ) , $0 \leq n \leq N-1$.

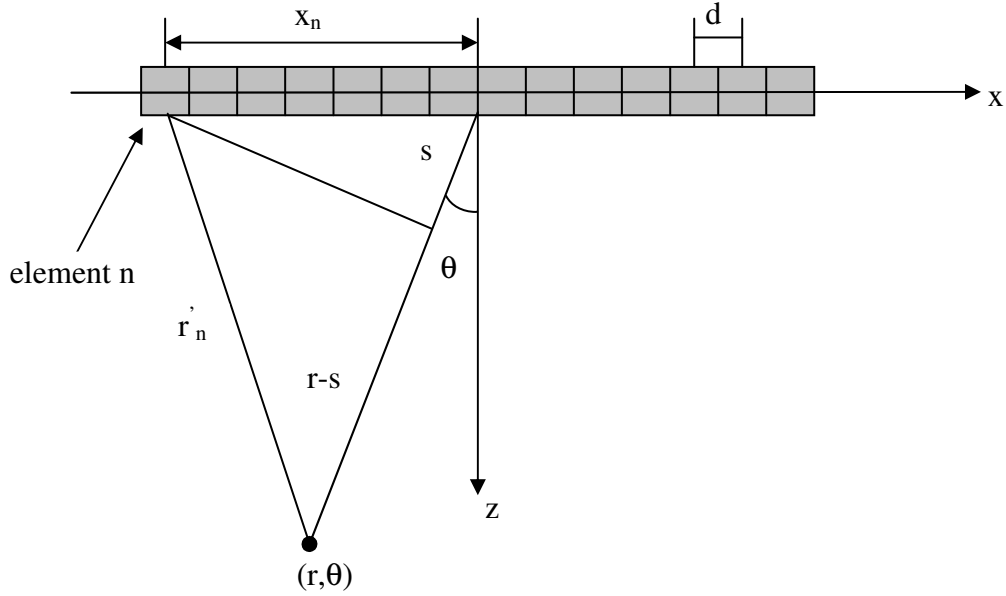


Figura 2.4: Geometria dirijării ariei fazate spre un punct de coordonate (r, θ) .

Conform legii cosinusului distanța dintre punct și elementul m , r_n' , este:

$$r_n' = \sqrt{x_n^2 + r^2 - 2x_n r \sin \theta} \quad (2.23)$$

unde r_n' este poziția elementului n . Întârzierea într-un singur sens pentru elementul n poate fi obținută prin substituirea ecuației 2.23 în ecuația 2.22 și astfel vom avea:

$$\tau_n = \frac{1}{c} \left(r - \sqrt{x_n^2 + r^2 - 2x_n r \sin \theta} \right) \quad (2.24)$$

Se dezvoltă funcția $\tau_n(x)$ în serie MacLauren: $\tau_n(x_n) = \tau_n(0) + \tau_n'(0) \cdot x_n + \frac{\tau_n''(0)}{2!} x_n^2 + \dots$

$$\tau_n'(x_n) = \frac{r}{c} \left[\frac{1}{2} \left(1 + \frac{x_n^2}{r^2} - \frac{2x_n \sin \theta}{r} \right)^{-1/2} \right] \left(\frac{2x_n}{r^2} - \frac{2 \sin \theta}{r} \right) \Rightarrow \tau_n'(0) = -\frac{\sin \theta}{c}$$

$$\tau_n''(x_n) = \frac{r}{c} \left[\frac{1}{r^2} \left(1 + \frac{x_n^2}{r^2} - \frac{2x_n \sin \theta}{r} \right)^{-1/2} - \frac{1}{2} \left(1 + \frac{x_n^2}{r^2} - \frac{2x_n \sin \theta}{r} \right)^{-3/2} \left(\frac{2x_n}{r^2} - \frac{2 \sin \theta}{r} \right) \left(\frac{x_n}{r^2} - \frac{\sin \theta}{r} \right) \right]^{-3/2}$$

$$\tau_n''(0) = \frac{r}{c} \left[\frac{1}{r^2} - \frac{\sin^2 \theta}{r^2} \right] = \frac{\cos^2 \theta}{rc}$$

Și se obține:

$$\tau_n(x_n) = -\frac{x_n \sin \theta}{c} + \frac{x_n^2 \cos^2 \theta}{2rc} = \tau_n^s + \tau_n^f \quad (2.25)$$

unde τ_n^s este întârzierea de direcționare iar τ_n^f este întârzierea de focalizare.

$$\tau_n^s = \frac{s}{c} = \frac{x_n \sin \theta}{c} \quad (2.26)$$

$$\tau_n^f = \frac{x_n^2 \cos^2 \theta}{2rc} \quad (2.27)$$

Funcția principală a focalizării este aceea de a îmbunătăți rezoluția laterală. Întârzierea datorată focalizării τ_n^f este dependentă de distanța r la punct. Prin urmare focalizarea poate fi fixă, dinamică sau compusă.

Focalizarea fixă sintetizează o focalizare doar pentru o anumită rază de acțiune R . În acest caz ecuația 2.27 poate fi rescrisă ca:

$$\tau_n^f = \frac{x_n^2 \cos^2 \theta}{2Rc}$$

Un punct de focalizare tipic este situat în mijlocul regiunii de vizualizare. Un fascicul focalizat are diametru minim la distanța R . Dincolo de această distanță, diametrul lateral al fascicului crește.

Pentru focalizarea dinamică, întârzierea de focalizare este ajustată în funcție de raza r . Focalizarea dinamică la recepție este de obicei folosită pentru ca adâncimea câmpului să fie extinsă fără reducerea frecvenței cadrelor. Recepția focalizată dinamic mărește electronic distanța focală a fascicului recepționat funcție de timp. Ecourile returnate de la toate adâncimile din interiorul țesutului sunt continuu în punctul focal. Focalizarea dinamică poate de asemenea să fie aplicată la transmisia fasciculelor. Se poate arăta că rezoluția laterală a imaginilor poate îmbunătățită și că lobii laterali pot fi suprimați dacă focalizarea dinamică este folosită și pentru transmisie și pentru recepție. Focalizarea dinamică în ambele sensuri poate fi folosită numai în vizualizarea prin apertură sintetică. În vizualizarea cu apertura reală implementarea focalizării dinamice la transmisie este impracticabilă deoarece ar necesita un timp de achiziție a datelor nelimitat. La majoritatea instrumentelor mai vechi întârzierea dinamică poate fi selectată la valori discrete și de aceea poate fi obținut doar un set discret de puncte de focalizare. Deoarece numărul de puncte de focalizare este limitat, unda rezultată are unele neregularități, dar acestea sunt suficient de mici încât practic nu au efect asupra calității imaginii.

Focalizarea compusă este un caz particular al focalizării dinamice, care folosește un număr mai mic de puncte de focalizare. Aceasta mărește adâncimea câmpului prin transmiterea câtorva secvențe de pulsuri, fiecare secvență fiind focalizată într-un punct diferit. Fasciculele cu lungimi focale diferite pot fi transmise numai succesiv. Cel de al doilea fascicul trebuie să fie transmis după ce toate ecourile produse de fasciculul anterior se întorc la traductor. Imaginile obținute prin transmisii cu lungimi focale diferite sunt decupate în jurul zonei focale și montate una după alta formând astfel un nou cadru de imagine. Este evident că acest proces crește timpul de formare a cadrului de imagine. Rata cadrelor descrește cu numărul de zone focale. De exemplu, rata cadrelor este N_f pentru o zonă focală. Când numărul de zone focale crește la patru, rata cadrelor descrește la $N_f/4$. De aceea, în practică, trebuie avut în vedere un compromis între calitatea imaginilor și rata cadrelor.

2.2.2 ÎNTÂRZIEREA ȘI ÎNSUMAREA LA EMISIE ȘI RECEPȚIE

Când un impuls scurt este transmis de elementul m și semnalul ecou este recepționat de elementul n , cum este arătat în figura 2.5, întârzierea dus-întors este:

$$\tau_{m,n} = \tau_m + \tau_n \quad (2.28)$$

unde (m,n) este o combinație între elementul de transmisie și cel de recepție $0 \leq m, n \leq N-1$ iar τ_m și τ_n pot fi obținute din ecuația (2.25).

Pentru o arie cu N elemente, semnalul de scanare tip A, $a_{PA}(t)$ este:

$$a_{PA}(t) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} y_{m,n}(t - \tau_{m,n}) \quad (2.29)$$

unde $y_{m,n}(t)$ este semnalul ecou și $r_{m,n}$ este întârzierea formatorului pentru combinația (m,n) dintre elementul de transmisie și cel de recepție dată în ecuația (2.28). Prima și a doua sumă corespund cu sinteza fascicului transmis și respectiv a celui recepționat.

Pentru fiecare punct în planul imaginii, semnalul de scanare tip A poate fi exprimat prin:

$$a(r, \theta) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} y_{m,n} \left(\frac{2r}{c} - r_{m,n} \right) \quad (2.30)$$

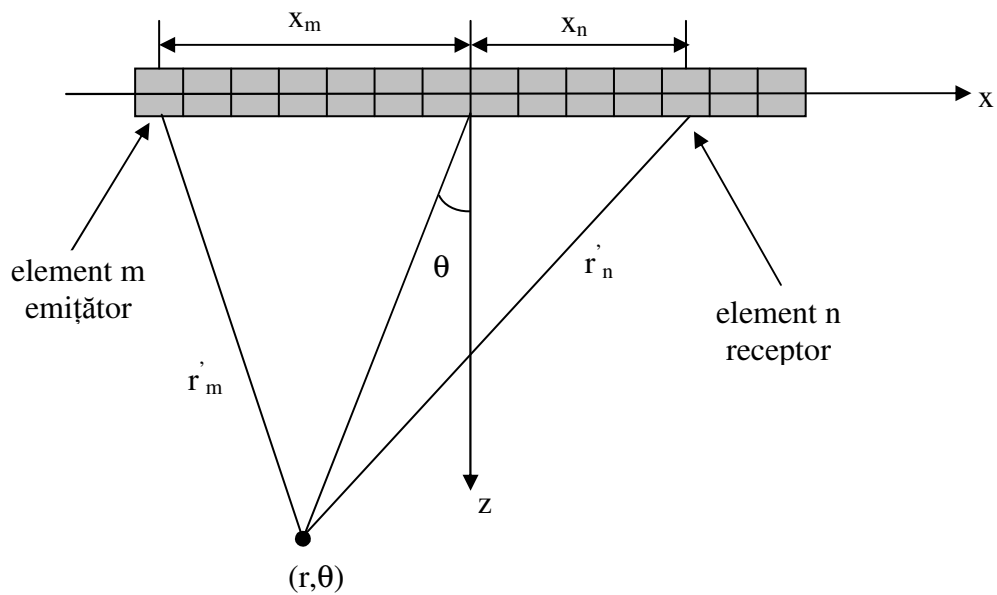


Figura 2.5: Legătura geometrică dintre elementele emițător - receptor și punctul de focalizare.

Când câmpul de undă este o undă plană monocromatică cu frecvența temporală ω , structura fascicului cu pondere uniformă pentru ambele direcții este:

$$W_{PA}(\theta) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} e^{-j\omega(r'_m + r'_n)} \quad (2.31)$$

Înlocuind ecuația 2.26 în ecuația 2.31, se obține:

$$w_{PA}(\theta) = \sum_{n=0}^{N-1} e^{j\frac{\omega}{c}x_n \sin \theta} \sum_{m=0}^{N-1} e^{j\frac{\omega}{c}x_m \sin \theta} = \left[\frac{\sin\left(k \frac{Nd}{2} \sin \theta\right)}{\sin\left(k \frac{d}{2} \sin \theta\right)} \right]^2 \quad (2.32)$$

Această ecuație este adevărată în câmpul depărtat, sau în regiunea focală a ariei comandate în fază. Cum era de așteptat, structura fascicului dus-întors datorată ariei cu ponderare uniformă este un sinus cardinal pătratic cu primul lob lateral la -25dB cum e arătat în figura 2.6.

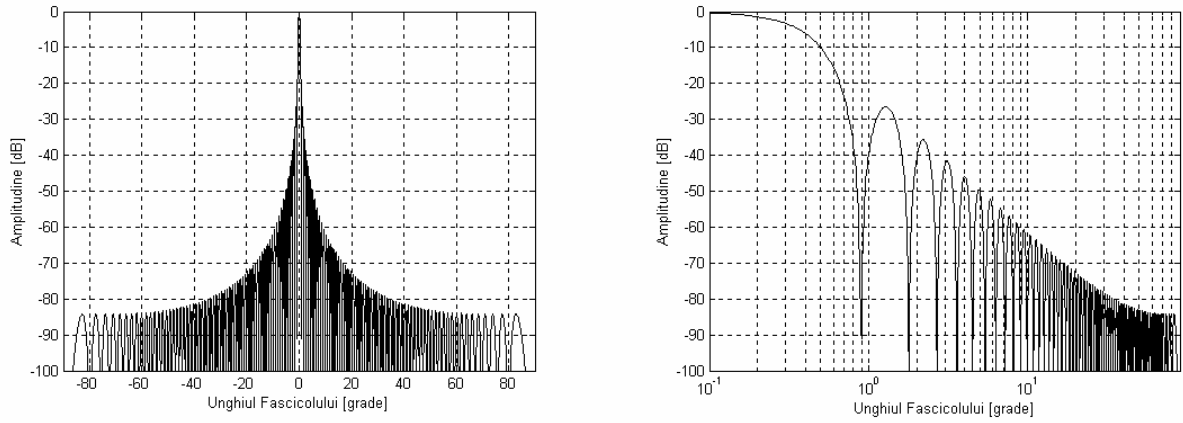


Figura 2.6: Structura fasciculului dus-întors pentru o arie fazată cu ponderare uniformă.

2.2.3 STRUCTURA FASCICULULUI

Utilizând structura fasciculului, se poate analiza ieșirea formatoarelor de fascicul pentru unde propagate în orice direcție. Structura fasciculului descrie calitatea acestuia, măsurată cel mai adesea prin lățimea fasciculului și nivelul lobilor laterali. În această secțiune, se va vedea cum poate fi controlată structura fasciculului prin proiectarea formatorului

După cum s-a discutat în secțiunea precedentă, când un sistem liniar dirijează pe direcția de propagare perpendiculară pe axa ariei, structura fasciculului este:

$$W(\theta) = \sum_{n=0}^{N-1} w_n e^{-jkx_n \sin \theta} \quad (2.33)$$

Structura fasciculului într-un singur sens în câmpul depărtat cu pondere uniformă este:

$$W(\theta) = \sum_{n=0}^{N-1} e^{-jkd \left(-\frac{N-1}{2} + n \right) \sin \theta} = e^{jkd \frac{N-1}{2} \sin \theta} \sum_{n=0}^{N-1} \left(e^{jkd \sin \theta} \right)^{-n} \quad (2.34)$$

Folosind egalitatea:

$$\sum_{n=0}^{N-1} a^{-n} = \frac{1 - a^{-N}}{1 - a} = \frac{a^{1-N/2}}{a^{-1/2}} \cdot \frac{a^{N/2} - a^{-N/2}}{a^{1/2} - a^{-1/2}}$$

Ecuția 2.34 se reduce la:

$$W(\theta) = \frac{\sin \left(k \frac{N}{2} d \sin \theta \right)}{\sin \left(k \frac{d}{2} \sin \theta \right)} \quad (2.35)$$

când: $k \frac{N}{2} d \sin \theta \neq m\pi$ pentru $m = \pm 1, \pm 2, \dots$

Se observă că egalitatea $W(\theta) = 0$ este îndeplinită când: $k \frac{N}{2} d \sin \theta = m\pi$ cu $m = \pm 1, \pm 2, \dots$

Pentru $m=1$, avem:

$$\sin \theta = \frac{\lambda}{Nd} \quad (2.36)$$

Ecuția 2.36 dă unghiul primului zero în caracteristica fasciculului format cu o arie comandată în fază cu N elemente. Pentru unghiuri mici avem $\sin \theta = \theta$, astfel:

$$\theta_{\text{zero}} \approx \frac{\lambda}{Nd} \quad (2.37)$$

θ_{zero} este lobul principal care conține cea mai mare parte din energia transmisă sau reflectată. Lățimea fasciculului la -3dB , corespunzătoare cu poziția unde amplitudinea este jumătate din

valoarea sa maximă este folosită frecvent pentru măsurarea lăţimii fasciculului. Aceasta este aproximativ $0,89\theta_{zero}$ şi defineşte rezoluţia laterală a sistemului.

Structura fasciculului în ambele sensuri se poate găsi prin convoluţia între aria transmiţătoare şi cea receptoare. În domeniul transformării Fourier aceasta reprezintă multiplicarea caracteristicii fasciculului transmis cu caracteristica fasciculului recepţionat. Întrucât acelaşi traductor transmite şi recepţionează ultrasunetul, se realizează de fapt convoluţia aperturii cu ea însăşi şi caracteristica fasciculului este astfel $W^2(\theta)$.

Dacă ieşirea fiecărui element este eşantionată cu frecvenţa $f_s = 1/t_s$ atunci întârzierea în timp discret este:

$$q_{nb} = \frac{x_n \sin \theta}{ct_s} = \frac{nd \sin \theta}{ct_s} \quad (2.38)$$

şi expresia ieşirii formatorului devine:

$$z(t) = \sum_{n=0}^{N-1} w_n y_n(t - \Delta_n) = \sum_{n=0}^{N-1} w_n y_n(t) e^{-jk^0 x_n \sin \theta} = \sum_{n=0}^{N-1} w_n y_n(t_s) e^{-j\omega^0 q_{nb} t_s} = \sum_{n=0}^{N-1} w_n y_n(t_s) e^{-j\Omega q_{nb}} \quad (2.39)$$

unde $\Omega = \omega t_s = 2\pi f/f_s$ este frecvenţa digitală unghiulară. Ecuaţia precedentă poate fi pusă în următoarea formă matriceală:

$$F(m) = w(n) \cdot Y(m,n) \exp^{-j\Omega q_{nb}} \quad (2.40)$$

unde w este vectorul ponderilor, $F(m)$ vectorul ieşire funcţie de indexul de timp discret m iar $Y(m,n)$ matricea intrărilor funcţie de indexul de timp discret m şi indexul elementului curent n . Vectorul întârzierilor $\exp^{-j\Omega q_{nb}}$ este indexat cu elementul curent n .

Ecuaţia (2.40) corespunde cu structura de implementare în timp discret din figura 2.7 şi sugerează modalitatea de implementare sau simulare în MATLAB a formatorului simplu întârziere-însumare.

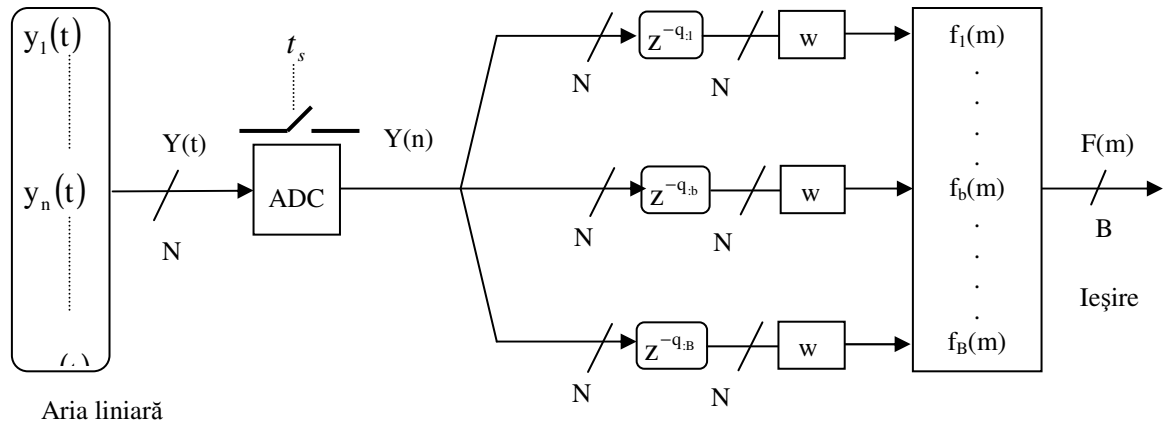


Figura 2.7: Structura formatorului întârziere – însumare în domeniul timp

Ieşirile elementelor ariei sunt convertite în semnale digitale folosind câte un convertor analog numeric (ADC) pentru fiecare element. Fiecărui fascicul îi corespunde un set de întârzieri discrete q_{nb} , unde $n = 0 \dots N-1$ iar b este fasciculul curent. Procedura de întârziere a intrărilor se realizează prin multiplicarea element cu element a fiecărei coloane din matricea intrărilor cu vectorul întârzierilor. Funcţia de ponderare w însumează semnalele decalate în timp pentru a forma fiecare fascicul. Structura de întârziere-sumare este repetată de B ori, B fiind numărul de fascicule dorit. Dacă pentru fiecare direcţie de dirijare se folosesc ponderi identice este mult mai eficientă ponderarea ieşirilor elementelor imediat după conversie. Se elimină astfel $(B-1) \cdot N$ multiplicări pentru fiecare set de eşantioane.

Întârzierile descrise prin ecuația (2.38) sunt multipli întregi ai perioadei de eșantionare t_s și prin urmare sunt realizabile fascicule pe direcțiile pentru care:

$$q_{nb} = \frac{nd \sin \theta}{ct_s} = 0, \pm 1, \pm 2, \dots$$

ceea ce impune următoarea restricție asupra unghiului de dirijare θ :

$$\theta = \arcsin\left(\frac{q_{nb}}{n} \cdot \frac{c}{d \cdot f_s}\right), q_{nb} = 0, \pm 1, \pm 2, \dots \quad (2.41)$$

Ecuația 2.41 arată că la formatoarele digitale în domeniul timp direcțiile de dirijare sunt în număr finit. Creșterea numărului acestora se poate face prin mărirea frecvenței de eșantionare, prin interpolare sau folosind tehnici de formare în domeniul frecvență. Aceste aspecte vor fi discutate în continuare.

2.2.4 EȘANTIONAREA ÎN DOMENIUL TIMP ȘI ÎN DOMENIUL SPAȚIU

Undele ultrasonice generate și emise de traductoarele ultrasonice sunt semnale continue în timp, care suferă un proces de împrăștiere înapoi la interacțiunea cu mediul investigat. Semnalele împrăștiate înapoi sunt recepționate și eșantionate de către traductor. În anumite condiții, semnalele continue în timp pot fi reprezentate adecvat prin eșantioane.

Teorema eșantionării Nyquist stabilește că frecvența de eșantionare trebuie să fie aleasă ca fiind cel puțin de două ori mai mare decât frecvența maximă a semnalelor, dacă semnalele sunt de bandă limitată. Dacă componenta de frecvență cea mai mare a semnalului de bandă limitată este ω , frecvența de eșantionare ω_s trebuie să satisfacă condiția: $\omega_s \geq 2\omega$. Dacă frecvența de eșantionare este aleasă $\omega_s < 2\omega$, semnalele sunt subeșantionate și apare fenomenul de aliere. Semnalele originale nu pot fi reconstruite din eșantioane.

Dacă traductorul prezintă o apertură discretă, eșantionarea semnalelor în domeniul spațiu se supune aceluiași constrângerii. Dacă semnalele sunt de bandă limitată cu un număr de undă mai mic decât $|\vec{k}| = 2\pi/\lambda$, atunci pot fi eșantionate fără pierdere de informație atâta timp cât perioada de eșantionare în domeniul spațiu este: $d \leq \pi/|\vec{k}| = \lambda/2$

unde λ este lungimea de undă. Prin urmare elementele ariei trebuie separate cu $\lambda/2$ sau mai puțin pentru a evita sub-eșantionarea spațială. O manifestare a eșantionării inadecvate în domeniul spațiu este apariția unui lob principal fals, sau lob imagine în caracteristica fasciculului, cum este ilustrat în figura 2.8.

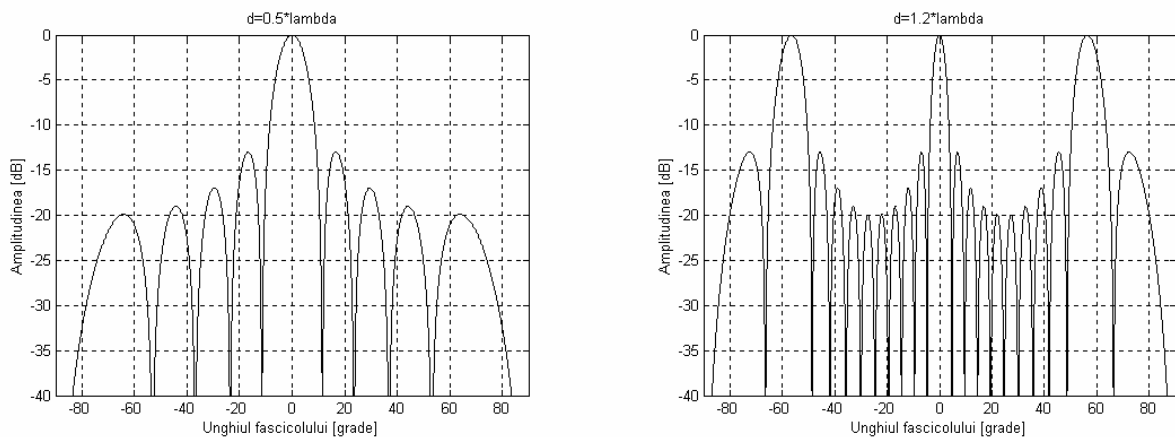


Figura 2.8: Structura fasciculului unidirecțional al unei arii liniare de 10 elemente cu $d=0.5\lambda$ (stânga) și $d=1.2\lambda$ (dreapta).

2.2.5. METODE DE CONTROL ALE FASCICULULUI DE UNDE

În general, structura dorită a fasciculului este aceea cu un lob principal îngust și lobi laterali de nivel redus în comparație cu lobul principal. Controlul fasciculului constă în a modela structura fasciculului folosind anumite metode. În imagistica ultrasonică medicală, metodele de control ale fasciculului: focalizarea dinamică, apertura dinamică și apodizarea (ponderarea), dau posibilitatea unui control flexibil asupra fasciculului ultrasonic.

În focalizarea dinamică, întârzierea pentru focalizare variază în funcție de distanța focală. Focalizarea dinamică furnizează un fascicul limitat de difracție. La focalizarea dinamică, diametrul focal este mult mai mic pentru adâncimile apropiate decât pentru cele depărtate. Pentru a avea o lățime uniformă zonei focale în întreg câmpul imaginii, numărul de elemente folosite la recepție este adesea redus la adâncimile apropiate în comparație cu cele depărtate.

Variația aperturii cu adâncimea este numită apertură dinamică. Cu apertura dinamică lobi laterali pentru puncte focale din apropiere pot fi ținuti la un nivel mic cu costul unui lob principal mai lat.

Apodizarea constă în reducerea amplitudinii vibrației spre muchia feței traductorului. Este folosită pentru a îmbunătăți forma fasciculului și a reduce nivelurile lobilor laterali. Reducerea amplitudinii se realizează prin aplicarea unui anumit tip de fereastră de ponderare: Hamming, Hanning, Bartlett, Kaiser, cosinus, etc. Figura 2.9 arată structura fasciculului obținut cu o arie fazată de 128 de elemente, în cazul propagării într-un singur sens, cu și fără fereastră de ponderare. La aplicarea unei ferestre, nivelurile lobilor laterali din structura fasciculului se reduc, dar lățimea lobului principal crește. Este de aceea necesar un compromis între lățimea lobului principal și nivelurile lobilor laterali.

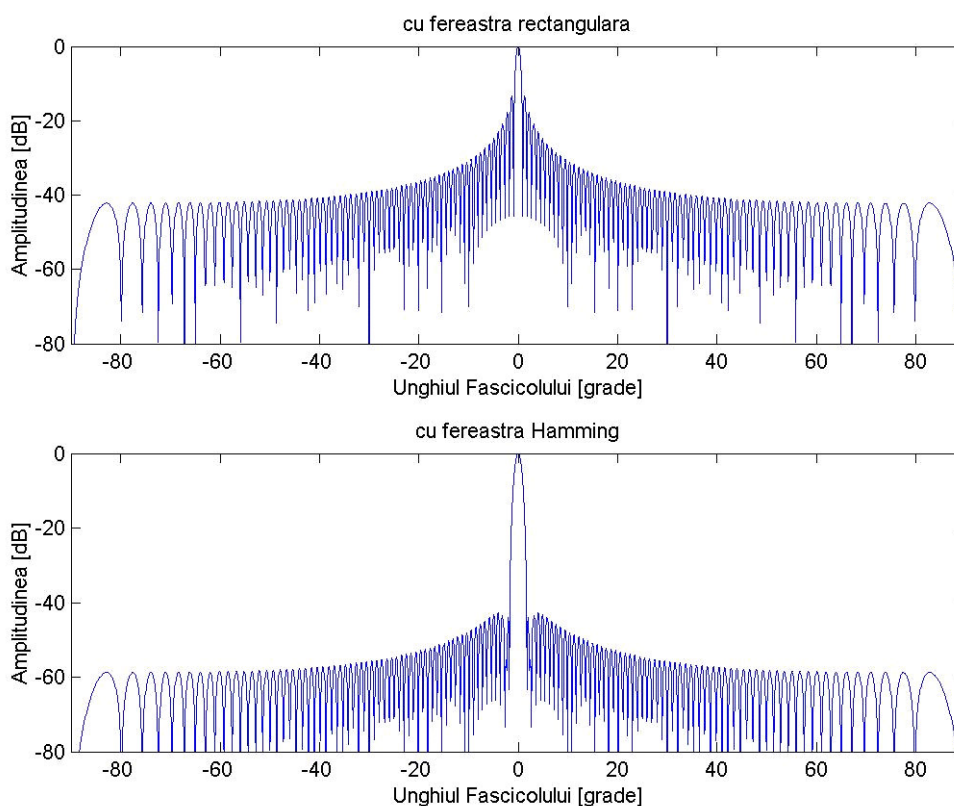


Figura 2.9 Structura fasciculului unei aii periodice cu 128 elemente pentru cazul ponderării cu fereastră rectangulară și respectiv cu fereastră Hamming.

2.3 FORMATOARE DIGITALE DE FASCICUL ULTRASONIC

Formatorul de fascicul reprezintă unul dintre blocurile esențiale ale unui sistem imagistic ultrasonic. Tehnicile de formare a fasciculelor ultrasonice joacă unul dintre cele mai importante roluri în imagistica ultrasonică [Lu94].

2.3.1 ISTORICUL EVOLUȚIEI FORMATOARELOR DE FASCICUL

Majoritatea sistemelor imagistice ultrasonice lansează cu un traductor o undă acustică de presiune în mediul de investigat și ascultă cu același traductor ecourile generate de interfețele acustice din mediu, înțelegând prin interfață acustică suprafața de separație dintre două regiuni cu impedanțe acustice diferite. Acest mod de operare puls-ecou înregistrează amplitudinea ecoului funcție de distanță (timp) și produce o linie în modul A. Primele dispozitive de investigare ultrasonică activau un traductor monoelement cu focalizare fixă pentru a trasa pe un osciloscop o linie mod A. Ulterior prin mișcarea laterală a traductorului și modularea strălucirii spotului cu amplitudinea semnalului recepționat s-a realizat scanarea în modul B care reprezintă grafic o secțiune transversală a mediului investigat.

În 1975 Albert Macovski a brevetat o metodă de control a elementelor unei arii liniare sau circulare pentru generarea unui fascicul focalizat dinamic care poate baleia o regiune din spațiu prin simpla modificare a întârzierilor aplicate elementelor ariei [Macovski75]. Această metoda a eliminat necesitatea baleierii mecanice pentru a scana o imagine în modul B și a deschis era ariilor ultrasonice comandate prin fază. Pentru o arie liniară (fig. 2.3) întârzierea totală aplicată fiecărui canal de recepție este dată de (2.25):

$$\tau(x) = \frac{x^2 \cos^2 \theta}{2Rc} - \frac{x \sin \theta}{c}$$

unde θ este unghiul de dirijare, R distanța focală, x distanța de la elementul curent la centrul ariei iar c viteza de propagarea ultrasunetelor în mediul investigat (1540 m/s pentru mediile biologice).

Primul termen dependent de R și θ corespunde focalizării la o distanță R , iar al doilea, dependent doar de unghiul θ descrie dirijarea după această direcție.

Această structură de arie a devenit un standard în sistemele comerciale de scanare cu arie fazată datorită unei construcții simple, flexibilității și unei bune sensibilități acustice.

Pentru implementarea analogică a întârzierilor fixe sau dinamice pentru o arie fazată în [Macovski75] s-a sugerat folosirea unui dispozitiv cu cuplare prin sarcină (CCD) pentru deplasarea unei cantități de sarcină de-a lungul unei linii de întârziere, cu o frecvență de tact dependentă de adâncimea R . Limitările unei astfel de soluții s-au datorat eficienței reduse a transferului de sarcină pentru dispozitivele disponibile la momentul respectiv și perturbațiilor rezultate din folosirea unui semnal de tact pentru fiecare linie de întârziere.

O altă implementare a folosit dispozitive cu undă de suprafață (SAW) cu prize sau tronsoane de cabluri coaxiale pentru a întârzia corespunzător semnalele pe fiecare canal. Comutarea pentru obținerea întârzierilor aferente focalizării dorite s-a realizat cu multiplexoare analogice sau rețele de comutatoare controlate de un microcontroler. Dificultățile majore ale acestei soluții au provenit de la zgomotul injectat în semnal la modificarea întârzierilor. Ca atare pentru obținerea unor imagini de calitate au fost necesare multiplexoare și linii de întârziere de înaltă performanță cu zgomot redus și prin urmare scumpe. Pe de altă parte focalizarea dinamică reclamă pentru fiecare canal rezoluții de întârziere de ordinul unei fracțiuni din perioada purtătoarei ultrasonice, fapt ce a presupus folosirea de linii de întârziere cu sute de prize, dificil de realizat. Costul acestor componente analogice de bandă largă, de înaltă precizie au făcut prohibitivă realizarea sistemelor cu un număr mare de canale.

În 1979 Maslak a brevetat o metodă pentru întârzierea dinamică fără comutarea prizelor de întârziere [Maslak79]. Prin heterodinarea semnalului de pe fiecare canal la o frecvență intermediară, un circuit de manipulare a fazei poate realiza ajustarea fină a întârzierii, în timp ce valoarea brută a acesteia se realizează cu o linie de întârziere. Pe această cale se pot crea multiple zone focale la recepție fără folosirea costisitoarelor linii de întârziere cu zgomot redus. Un canal al sistemului de întârziere propus este prezentat în fig. 2.10 A.

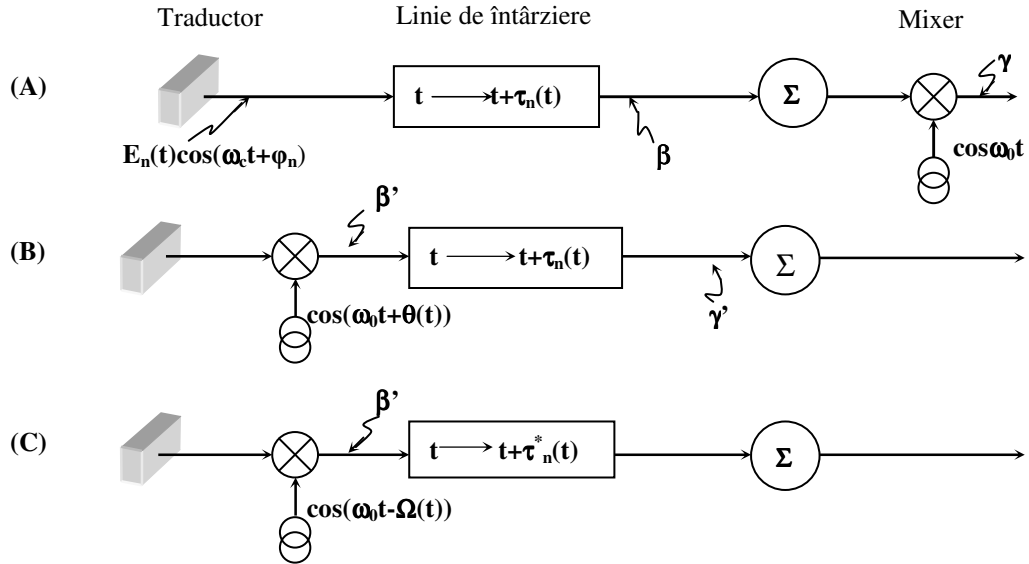


Figura 2.10 Scheme pentru focalizare dinamică (Maslak79)

Semnalul recepționat la intrare de forma $E_n(t)\cos(\omega_c t + \varphi_n)$ este întârziat cu o valoare dependentă de distanță (timp) $\tau_n(t)$, însumat cu celelalte canale și apoi heterodinat cu o frecvență intermediară cu $\cos\omega_0 t$. Anvelopa pulsului ultrasonic este dată de $E_n(t)$, ω_c este purtătoarea ultrasonică iar φ_n este defazajul pentru un parcurs dus – întors de la centrul ariei la receptor. Semnalul în punctul β are expresia $E_n[t + \tau_n(t)]\cos\{\omega_c[t + \tau_n(t)] + \varphi_n\}$ și reprezintă o purtătoare și o anvelopă, întârziate, care interferează constructiv, la generarea fasciculului prin însumarea cu celelalte canale, deoarece $\tau_n(t)$ compensează defazajul φ_n . Ieșirea mixerului are expresia:

$$\frac{1}{2}E_n(t + \tau_n(t))\cos\{\omega_0 t + [\omega_c(t + \tau_n(t)) + \varphi_n]\} + \frac{1}{2}E_n(t + \tau_n(t))\cos\{\omega_0 t - [\omega_c(t + \tau_n(t)) + \varphi_n]\} \quad (2.42)$$

unde primul termen indică banda laterală superioară (suma frecvențelor) iar al doilea banda laterală inferioară (diferența frecvențelor). Prin filtrare termenul sumă va fi eliminat și se va păstra doar componenta de frecvență intermediară. Motivul heterodinării fasciculului în acest mod este reducerea ratei de procesare în blocurile ce succed mixerului și prin urmare posibilitatea folosirii de componente mai lente dar mai ieftine.

Al doilea cadru din fig.2.10 descrie situația în care mixerul precede liniile de întârziere dar în acest caz în semnalul de mixare a fost inclusă o variabilă pentru ajustarea fazei $\theta(t)$. Variabila $\theta(t)$ se definește astfel ca la ieșirea blocului de întârziere $\theta(t + \tau_n(t)) = \omega_0 \tau_n(t)$. Pentru această fază a semnalului de mixare semnalul în punctul β' este:

$$\frac{1}{2}E_n(t)\cos\{\omega_0 t - \theta(t) + [\omega_c t + \varphi_n]\} + \frac{1}{2}E_n(t)\cos\{\omega_0 t - \theta(t) - [\omega_c t + \varphi_n]\} \quad (2.43)$$

Este ușor de observat de aici că dacă se aplică o întârziere ideală $t \rightarrow t + \tau_n(t)$; termenul $\omega_0 t$ generează o componentă $\omega_0 \tau_n(t)$ care este anulată de $\theta(t + \tau_n(t))$. Ca urmare semnalul după

întârziere este identic cu cel indicat de relația (2.42). Efectul imediat al heterodinării semnalului ultrasonic la diferența frecvențelor este că prizele liniei de întârziere nu mai sunt atât de apropiate deoarece purtătoarea este de frecvență mai mică. Aceasta simplifică structura liniei de întârziere deoarece sunt necesare mai puține prize.

Simplificarea finală dezvoltată de Maslak este ilustrată în fig.2.10C, unde linia nu furnizează întârzierea ideală ci o altă întârziere $\tau_n^*(t)$ variabilă sau nu funcție de timp. Setând faza mixerului astfel ca $\Omega(t+\tau_n^*(t)) = \omega_0\tau_n^*(t) + \omega_c(\tau_n(t)-\tau_n^*(t))$ obținem următoarea expresie pentru semnalul de la ieșirea liniei de întârziere:

$$\begin{aligned} & \frac{1}{2} E_n(t+\tau_n^*(t)) \cos\{\omega_0 t + [\omega_c(t+\tau_n(t)) + \phi_n] - 2\omega_c[\tau_n(t) - \tau_n^*(t)]\} \\ & + \frac{1}{2} E_n(t+\tau_n^*(t)) \cos\{\omega_0 t - [\omega_c(t+\tau_n(t)) + \phi_n]\} \end{aligned} \quad (2.44)$$

Acest semnal conține un termen sumă de frecvență mai complicat decât precedentul, dar acesta va fi oricum eliminat mai târziu prin filtrarea trece jos sau trece bandă. Pe de altă parte termenul diferență are aceiași purtătoare ca în (2.42) dar anvelopa este decalată cu $\tau_n^*(t)$. Deoarece anvelopa are o variație lentă comparativ cu purtătoarea, un fascicul sumă generat cu acest semnal poate fi considerat echivalent cu cel generat printr-un sistem ideal de întârzieri. Particularitatea acestei soluții este că întârzierea aplicată semnalului $\tau_n^*(t)$ poate rămâne constantă un timp mult mai lung comparativ cu modificările fazei mixerului. Cu alte cuvinte pentru un fascicul, o singură priză pe linia de întârziere poate asigura formarea adecvată a fasciculului, dacă faza semnalului de mixare pentru fiecare canal este actualizată în concordanță cu întârzierea de focalizare dorită. Aceasta permite realizarea formatorului de fascicul cu linii analogice și multiplexoare ieftine și cu performanțe de zgomot medii deoarece frecvența comutării prizelor este substanțial redusă și chiar eliminată în anumite situații. Modificarea întârzierilor se realizează pe fiecare canal cu un mixer relativ simplu și precis.

Extensia acestui principiu la procesarea bandă de bază cu componentele complexe în fază (I) și în cuadratură (Q) este imediată. Un dezavantaj al acestei tehnici este că necesită două linii de întârziere pentru a procesa ambele componente. De asemenea valoarea limitată a raportului semnal zgomot (SNR) pentru componentele analogice (tipic 60-80 dB) influențează calitatea fasciculului generat.

O dificultate asociată cu mixarea analogică este că aceasta este o abordare se tip bandă îngustă. Se consideră un semnal de intrare alcătuit din două componente cosinusoidale care reprezintă purtătoarea ω_c și banda laterală ω_b . Sistemul prezentat în fig.2.10C folosește termenul $\Omega_n(t)$ pentru a elimina termenii lui ω_c induși prin folosirea unei întârzieri $\tau_n^*(t)$ diferită de întârzierea necesară $\tau_n(t)$. Pentru acest semnal de bandă largă setarea $\Omega(t+\tau_n^*(t)) = \omega_0\tau_n^*(t) + \omega_c(\tau_n(t)-\tau_n^*(t))$ nu elimină termenii induși în componenta ω_b . Prin urmare la formarea fasciculelor ultrasonice din semnale de bandă largă apar erori de fază care sunt cu atât mai semnificative cu cât liniile de întârziere sunt mai grosiere. Cu toate acestea această tehnică a fost larg folosită în primele sisteme ultrasonice pentru generarea de fascicule de bună calitate cu costuri rezonabile.

2.3.2 FORMAREA DIGITALĂ A FASCICULELOR RECEPTOARE

De la sfârșitul anilor 80 progresele înregistrate în conversia analog numerică au condus la realizarea formatoarelor de fascicul în tehnică digitală. S-au dezvoltat patru tehnici digitale pentru modificarea dinamică a întârzierilor.

O primă tehnică este asemănătoare cu sistemul cu frecvență intermediară propus de Maslak cu excepția unui convertor analog numeric (CAN) inserat înaintea liniilor de întârziere care sunt implementate de această dată prin blocuri de memorie [O'Donnell90]. Figura 2.11.A arată un sistem care are la bază un mixer cu fază ajustabilă pentru translatarea semnalului ultrasonic la o frecvență joasă și modificarea fină a întârzierilor. Avantajul unei astfel de implementări față de sistemele analogice rezidă în aceea că în tehnică digitală modificarea întârzierii nu introduce zgomot și $\tau_n(t)$ poate fi modificată pentru a urmări îndeaproape valoarea necesară $\tau_n(t)$ ceea ce lărgeste banda de lucru a unui astfel de sistem. De asemenea, deoarece fasciculul este generat ca rezultat al însumării digitale, semnalul de ieșire poate avea un domeniu extins de variație dinamică. Ca dezavantaj se reține necesitatea includerii unui filtru analogic anti-aliere înaintea CAN pentru a elimina componentele de frecvență înaltă din semnalul de la ieșirea mixerului.

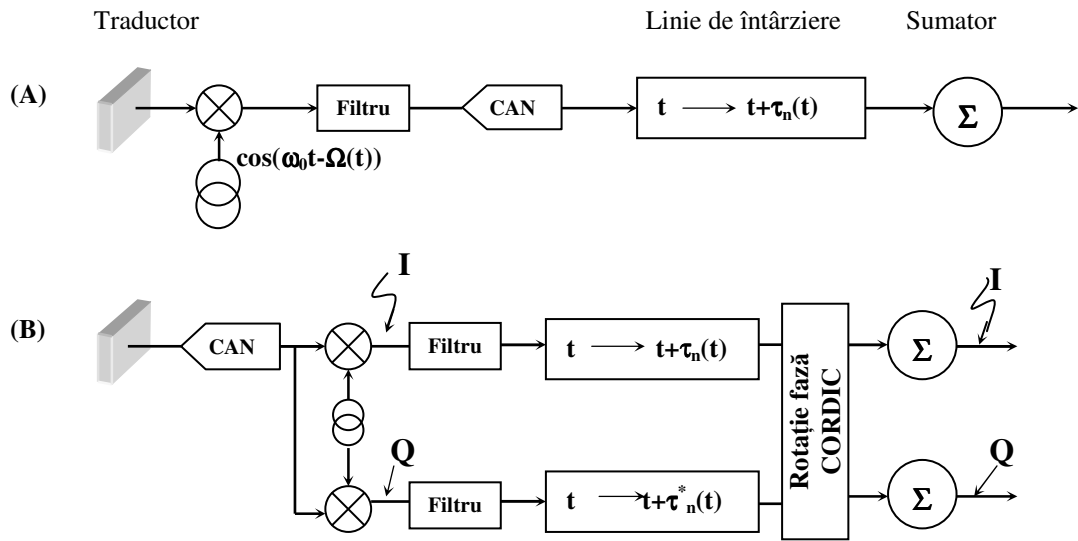


Figura 2.11 Scheme pentru focalizare dinamică digitală

Într-un sistem similar indicat în fig. 2.11.B, CAN – ul este conectat direct la semnalul recepționat de la traductor iar demodularea de frecvență în banda de bază este realizată digital [O'Donnell90, Engeler92, O'Donnell91, Maslak96]. În acest caz CAN-ul operează la frecvență mai mare decât în cazul precedent întrucât semnalul este pe frecvența purtătoare originală. Componentele în fază I și în cuadratură Q pot fi filtrate și decimate pentru a reduce rata datelor și capacitatea de stocare a memoriilor ce succed demodulatorului. Pentru realizarea întârzierii fine se ajustează faza semnalului bandă de bază prin rotirea eșantionului complex întârziat brut folosind un procesor CORDIC [Freeman95].

Structura detaliată a unui canal de recepție pentru un astfel de sistem este arătată în figura 2.12. Blocul de control al canalului primește la intrare tactul principal, distanța R și unghiul θ și generează un tact întârziat DCLK pentru comanda conversiei analog numerice, un semnal de tact pentru demodulare, filtrare și decimare (CLK') și un altul pentru procesarea în banda de bază BBCLK. De asemenea același bloc calculează și trimite spre procesorul CORDIC semnalul pentru rotirea fazei. Practic din R și θ se determină pentru canal întârzierea exprimată printr-un număr întreg de perioade de tact (întârzierea brută) plus o fracțiune de perioadă (întârzierea fină). Întârzierea brută se implementează prin activarea semnalului DCLK după un număr întreg de perioade iar ajustarea se face prin rotire cu valoarea indicată de întârzierea fină.

Avantajele acestei soluții constau în aceea că nu mai sunt necesare componente analogice complexe și precise pentru a genera și selecta diferite valori ale fazei semnalului pentru mixare. O valoare digitală aplicată la intrarea procesorului CORDIC rotește cu precizie fiecare eșantion cu unghiul dorit.

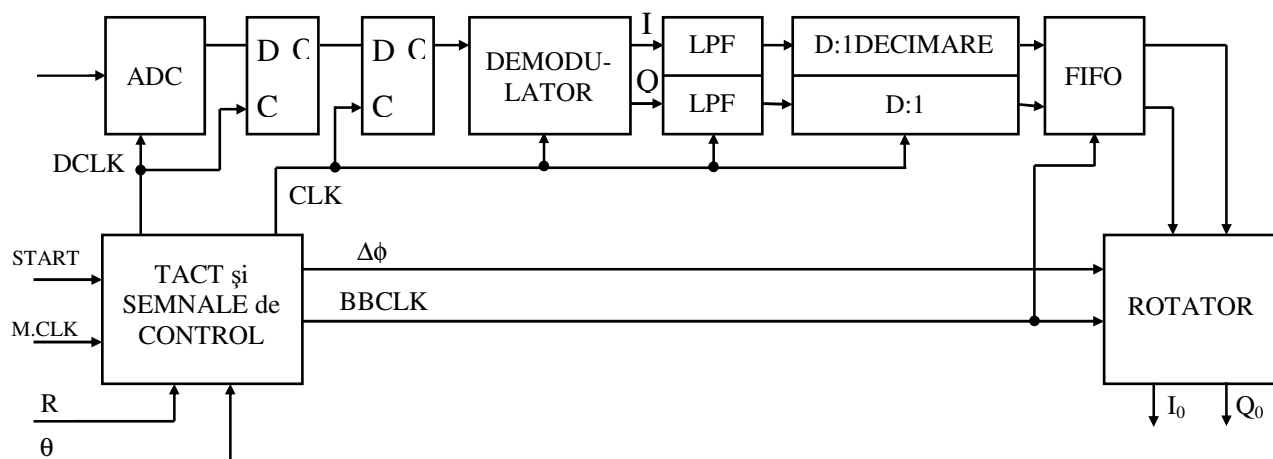


Figura 2.12 Structura canalului pentru focalizarea dinamică digitală prin rotirea fazei

O a doua tehnică de generare și control digital a întârzierii dinamice folosește interpolarea pentru a supraeșantiona semnalul digital livrat de CAN și a asigura eșantioane cu decalaj temporal suficient de fin între ele. Această tehnică reduce pretențiile în ceea ce privește frecvența de eșantionare a CAN prețul plătit fiind creșterea complexității secțiunii de procesare digitală. Pridham și Mucci în 1979 [Pridham79] au propus extinderea cu zerouri a secvenței eșantioanelor generate de CAN pe fiecare canal urmată de sinteza fasciculului prin întârziere și însumare la o rată înaltă și în final filtrarea și decimarea secvenței obținute.

Deoarece operația de formare a fasciculului prin întârziere și însumare și cea de filtrare sunt operații liniare (pentru un punct focal fixat) acestea pot fi interschimbate. În concordanță cu această observație Pridham și Mucci au propus două alternative de abordare. În prima dintre ele circuitul pentru extinderea cu zerouri și filtrul de interpolare pentru fiecare canal sunt plasate în urma CAN dar înaintea formatorului de fascicul. În a doua alternativă corespunzătoare interpolării post – formare, filtrul de interpolare este plasat după formatorul de fascicul. Această soluție este optimă în ceea ce privește cerințele de procesare deoarece folosește un singur filtru de interpolare la ieșirea formatorului și nu un filtru pe fiecare canal receptor.

Pentru a reduce în continuare cerințele privind frecvența de eșantionare Kim a extins tehnica de interpolare / decimare [Kim95] realizând sinteza parțială de fascicul la frecvența înaltă obținută prin extensia cu zerouri și apoi filtrând și decimând secvența înainte de formarea fasciculului final. Avantajul acestei abordări provine din faptul că prin procesarea parțială se reduce rata de procesare în formatorul digital.

Funcția implementată prin interpolare / decimare poate fi îndeplinită și aplicând diferite filtre trece jos sau trece bandă unei secvențe de eșantioane ne extinse cu zerouri. Structura unui astfel de interpolator / decimator este dată în figura 2.13. În acest caz fiecare întârziere necesită propriul său filtru, spre deosebire de metoda extinderii cu zerouri unde se selectează diferite secvențe de filtrare prin deplasarea eșantioanelor ne nule pe o versiune supraeșantionată a aceluiași filtru. Lipschutz a propus în 1994 o structură ca cea din fig.2.13 și a introdus condiția suplimentară ca toți coeficienții filtrelor să fie de forma $1/2^m$, unde m este întreg [Lipschutz94]. Prin aceasta multiplicările se reduc la simple operații de deplasare binară.

Spre deosebire de metoda de defazare discutată anterior, metodele de interpolare – decimare sunt de bandă largă deoarece întârziere cu aceeași acuratețe toate frecvențele din banda de trecere a filtrului. Există desigur și în acest caz erori datorate trunchierii filtrului și acurateții coeficienților. Metoda Lipschutz este simplă datorită folosirii coeficienților de

forma $1/2^m$ dar prezintă o flexibilitate limitată în asigurarea de întârzieri precise pentru semnale cu diferite purtătoare sau lăţimi de bandă.

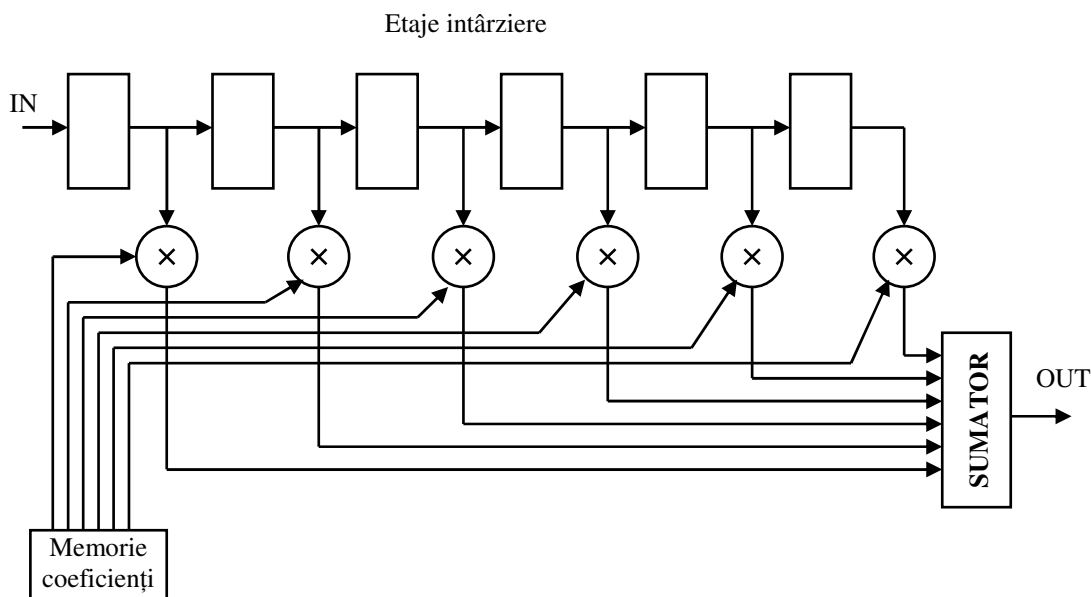


Figura 2.13 Structura interpolatorului Lipschutz

A treia tehnică de generare a fasciculelor folosește eșantionarea neuniformă a semnalului RF pentru a asigura acuratețea dorită a întârzierilor, păstrând frecvența de eșantionare la o valoare relativ redusă. [Song94, O'Donnell89a, O'Donnell89b].

În această tehnică sunt preluate doar eșantioanele necesare pentru focalizarea fasciculului ultrasonic în fiecare punct al imaginii. Pentru aceasta se generează semnale de tact diferite pentru fiecare element traductor, semnale care respectă criteriul Nyquist dar sunt întârziate unul față de altul astfel ca să se asigure focalizarea în punctul din imagine dorit. Cu alte cuvinte eșantionarea și întârzierea sunt realizate simultan și nu separat ca în tehnicile anterioare. În consecință frecvența de eșantionare este determinată de distanța dintre punctele imaginii sau mai general de rata minimă de eșantionare pentru detecția anvelopei semnalului focalizat. Pentru un traductor ultrasonic cu o lăţime de bandă egală cu frecvența sa centrală, frecvența minimă de eșantionare pentru $f_0=5\text{MHz}$, va fi 15MHz (frecvența de eșantionare Nyquist) sau 10MHz dacă se folosesc tehnici de eșantionare trece bandă ca de exemplu eșantionarea de ordinul doi sau eșantionarea în cuadratură [Chang93], [Cho96], [Na98]. Aceasta duce la o reducere dramatică în cerințele de hardware comparativ cu metodele care folosesc eșantionarea uniformă convențională. În plus utilizarea memoriilor FIFO elimină operația de adresare pentru semnalele eșantionate rezultând în continuare o reducere a complexității structurii hardware a formatorului de fascicul. Elementul cheie al sistemului indicat în figura 2.14 este blocul de generare a semnalelor de activare neuniformă a eșantionării. Prezența blocurilor de memorie FIFO asigură formarea fasciculului ultrasonic prin însumare coerentă. Astfel înscrierea memoriilor de face cu tactul neuniform generat pentru fiecare canal în parte iar citirea se face simultan. Latchurile L înseriate cu memoria asigură secvențialitatea corectă în fazele de *conversie – scriere memorie* și respectiv *citire memorie – însumare*. Dimensionarea memoriei FIFO este funcție de valoarea maximă a întârzierii care apare când se baleiază regiunea dorită cu o arie dată.

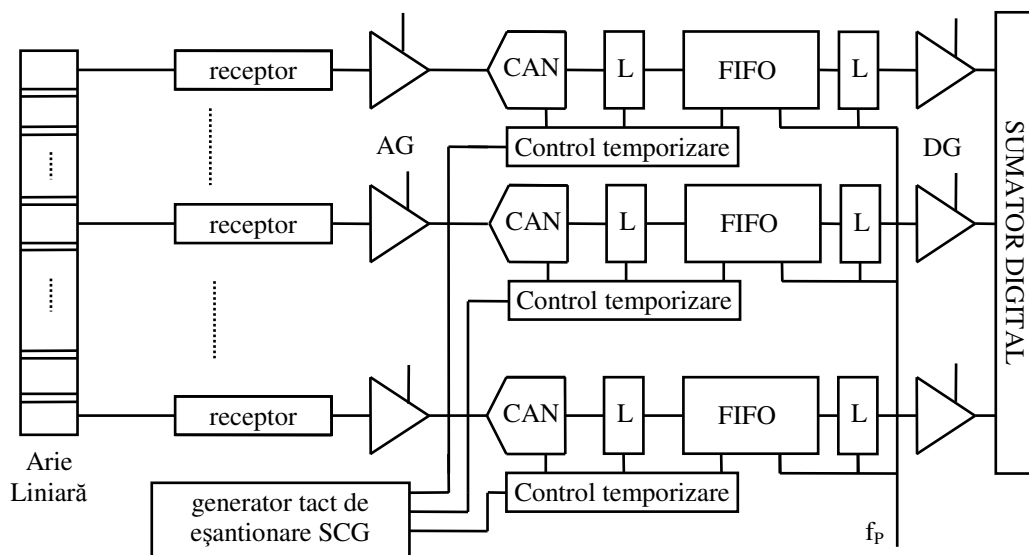


Figura 2.14 Structura formatorului de fascicul cu eșantionare neuniformă

O soluție interesantă care folosește tehnica de eșantionare neuniformă și reduce complexitatea structurii hardware a fost propusă de Song în 1994 [Song94]. Sistemul propus, indicat în figura 2.15, folosește de asemenea tehnica de eșantionare neuniformă pentru compensarea întârzierilor, dar prelevează eșantioane analogice din semnalele ecou și formează fasciculul la nivelul analogic. Prin urmare CAN-ul memoria FIFO și circuitul de control a temporizării din fig. 2.14 vor fi înlocuite cu un singur dispozitiv analogic AFIFO (FIFO Analogic). Ieșirea fiecărui element este conectată la un canal receptor care realizează la nivel analogic amplificarea, compensarea temporală a câștigului (TGC) și compresia semnalului. Ieșirile receptoarelor sunt eșantionate neuniform exact ca în cazul formatorului digital dar eșantioanele din semnal nu sunt cuantificate ca valori numerice ci reprezintă valori analogice instantanee. Dispozitivul analogic AFIFO este folosit pentru a realiza eșantionarea neuniformă și stocarea eșantioanelor analogice. Întârzierea necesară pentru focalizare este asigurată prin decalarea în timp a momentului eșantionării și nu prin selecția eșantionului potrivit dintr-o colecție de eșantioane prelevate uniform.

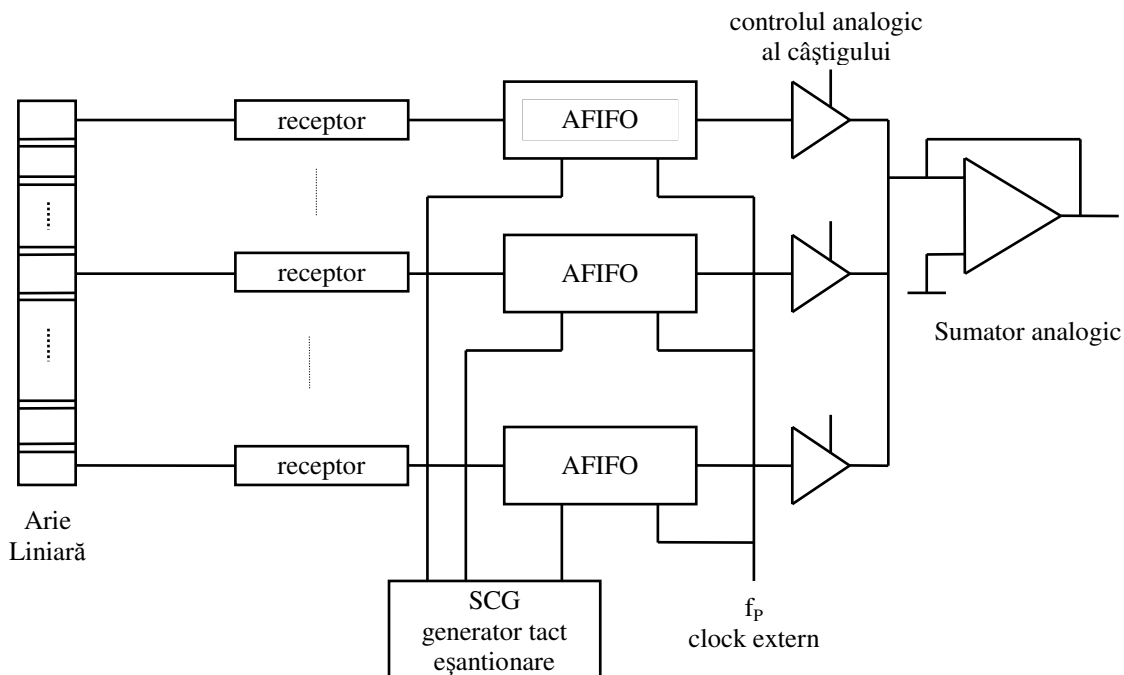


Figura 2.15 Formator de fascicul cu eșantionare neuniformă – varianta analogică

Pentru fiecare punct al imaginii eşantioanele compensate ca întârziere sunt stocate în memoriile AFIFO la diverse momente de timp. Eşantioanele pentru fiecare punct trebuie prelevate simultan la ieşirea AFIFO şi ca atare tuturor memoriilor le este furnizat un tact de citire uniform cu frecvenţa f_p activat după un timp mai mare sau cel puţin egal cu întârzierea maximă. Conform principiului FIFO, fără vreo operaţie de selecţie sau adresare a eşantioanelor stocate în AFIFO, eşantioanele semnalelor ecou cu întârzierile corect compensate pentru un punct curent al imaginii sunt prelevate simultan la un sumator analogic.

Figura 2.16. indică o posibilă configuraţie pentru AFIFO în care un număr de celule de eşantionare / memorare (CEM) sunt integrate cu o logică digitală simplă pentru controlul intrării şi ieşirii. Semnalul de intrare este conectat printr-un buffer la intrările tuturor CEM: Tactul de eşantionare neuniform este furnizat secvenţial fiecărei celule CEM, începând cu prima celulă, prin intermediul unui generator de secvenţă constituit dintr-un numărător în inel şi porţi logice. Tactul uniform de ieşire comun tuturor dispozitivelor AFIFO este de asemenea distribuit printr-un generator de secvenţă de ieşire implementat printr-un numărător în inel. Acest generator activează simultan multiplexoarele analogice de pe ieşirea AFIFO la ieşirea cărora se obţin eşantioanele punctului imagine dorit.

Practic pentru fiecare canal suportul hardware pentru compensarea întârzierilor poate fi integrat în două circuite: un generator programabil pentru tactul de eşantionare (SCG) şi memoria AFIFO. În structura generală a sistemului de imagistic la acestea se adaugă pentru fiecare canal un amplificator cu câştig controlabil şi un sumator analogic global care formează fasciculul prin însumarea eşantioanelor cu întârzierea compensată. SCG fiind un circuit digital poate fi implementat fie ca o memorie pe 1 bit ce se încarcă cu secvenţa de eşantionare precalculată, fie ca un circuit ce generează on line secvenţa de eşantionare. Prima variantă este mai simplă de implementat dar necesită un circuit digital extern pentru a stoca secvenţa de tact precalculată. Practic circuitul integrat pentru SCG va conţine două memorii pe 1 bit dintre care una furnizează secvenţa de tact pentru linia curentă de investigare, în timp ce a doua este încărcată cu secvenţa de tact aferentă liniei următoare. Pentru al doilea tip de circuit SCG trebuie să fie furnizate din exterior doar coordonatele punctului imagine şi poziţia elementului.

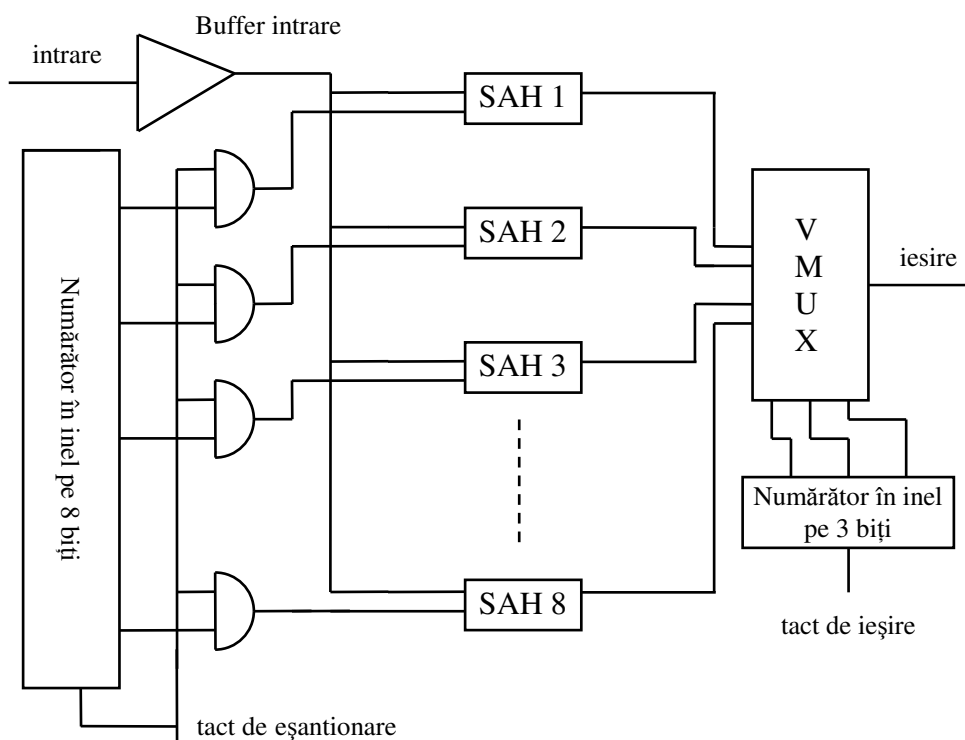


Figura 2.16 Structura AFIFO

A patra tehnică de generare a fasciculelor de ultrasunete a apărut ca și precedenta din necesitatea de a reduce în continuare structura hardware a sistemului la nivelul fiecărui canal în condițiile folosirii de arii cu un număr tot mai mare de elemente (128; 256). Această tehnică presupune folosirea convertoarelor cu supraeșantionare în componența blocului digital de intrare, pentru a reduce complexitatea hardware. General Electric a brevetat în 1993 [Noujaim93] un formator care folosește un CAN delta-sigma ($\Delta\Sigma$) cu supraeșantionare și scheme simple de procesare pentru întârzierea și însumarea coerentă.

Tehnicile digitale multi-bit de formare a fasciculelor indicată până aici produc imagini ultrasonice de înaltă calitate. Folosind tehnici de supraeșantionare pot fi obținute imagini cu o calitate similară în condițiile reducerii semnificative a complexității și gabaritului sistemului. Formatoarele ce au la bază convertoare $\Delta\Sigma$ prezintă două avantaje majore față de metodele tradiționale multi bit. Primul dintre ele costă în simplificarea structurii CAN și posibilitatea integrării pe același chip împreună cu blocurile de procesare digitală care creează fasciculul. Circuitele necesare pentru a implementa CAN $\Delta\Sigma$ sunt mult mai simple decât cele folosite în CAN flash multi-bit (câteva amplificatoare operaționale și un comparator față de 255 comparatoare necesare pentru un CAN flash pe 8 biți), fapt ce se regăsește în reducerea resurselor de interconectare, a gabaritului și puterii consumate.

În al doilea rând se pot asigura simplu întârzieri precise prin manipularea eșantioanelor prelevate cu o rată de eșantionare ridicată. Circuitele complicate pentru procesare în banda de bază sau pentru interpolare / decimare pot fi eliminate deoarece o memorie FIFO convențională poate asigura întârzierea cu rezoluția dorită înainte de însumare. În plus, însumarea pentru întreaga arie se simplifică întrucât primul nivel de sumatoare este alcătuit din sumatoare pe un bit.

Datorită domeniului dinamic larg necesar pentru aplicațiile ultrasonore, CAN $\Delta\Sigma$ trebuie să fie un modulator de ordinul doi sau trei pentru a menține un raport semnal zgomot adecvat. În [Norman96] a fost propus un modulator de ordinul patru pentru imagistica ultrasonică. Pentru a asigura atât un raport semnal zgomot ridicat cât și o acuratețe adecvată a întârzierilor s-a ales o frecvență de eșantionare de cel puțin 32 ori mai mare decât frecvența purtătoare. Cuantificatorul de la ieșirea modulatorului se alege în mod obișnuit cu două niveluri de cuantizare datorită simplității sale. Adăugarea de niveluri de cuantificare îmbunătățește raportul semnal zgomot al sistemului dar complică circuitul de reacție, în special CAN – ul din componența modulatorului.

După cum se arată în fig. 2.17 ieșirea digitală a modulatorului se aplică liniilor digitale de întârziere cu lungime variabilă, implementate ca memorii FIFO sau registre de deplasare, care realizează întârzierea dinamică a semnalelor. Acest registru de deplasare trebuie să opereze la frecvența de eșantionare și conține câteva sute de celule pentru a găzdui întârzierile dinamice pe durata fiecărui fascicul. Lungimea liniei de întârziere depinde de câțiva factori: frecvența de eșantionare, geometria ariei, unghiul de dirijare, adâncimea maximă de investigare și frecvența purtătoare (frecvența centrală a traductorului). Când lungimea liniei de întârziere se modifică pentru a produce o nouă zonă focală, eșantioanele trebuie repetate, inserate sau eliminate la intrarea, ieșirea sau în punctul median a registrului.

Eșantioanele întârziate de la toate elementele ariei sunt apoi însumate digital pentru a forma fasciculul. Această însumare trebuie realizată la frecvența de eșantionare și necesită un set de sumatoare digitale pipeline pentru întreaga arie. Odată fasciculul generat prin sumare, semnalul se aplică unui filtru trece jos cu pantă abruptă pentru a elimina zgomotul de cuantizare $\Delta\Sigma$. Acest filtru de ordin superior poate fi foarte lung și complicat și cu un anumit consum dar este necesar un singur astfel de filtru pentru generarea unui fascicul. Decimarea șirului filtrat de eșantioane reduce considerabil rata de eșantionare, permițând astfel folosirea în continuare a elementelor de procesare tradiționale. Detecția anvelopei, conversia formatului de scanare și afișarea pot fi realizate prin metode folosite și de celelalte tehnici.

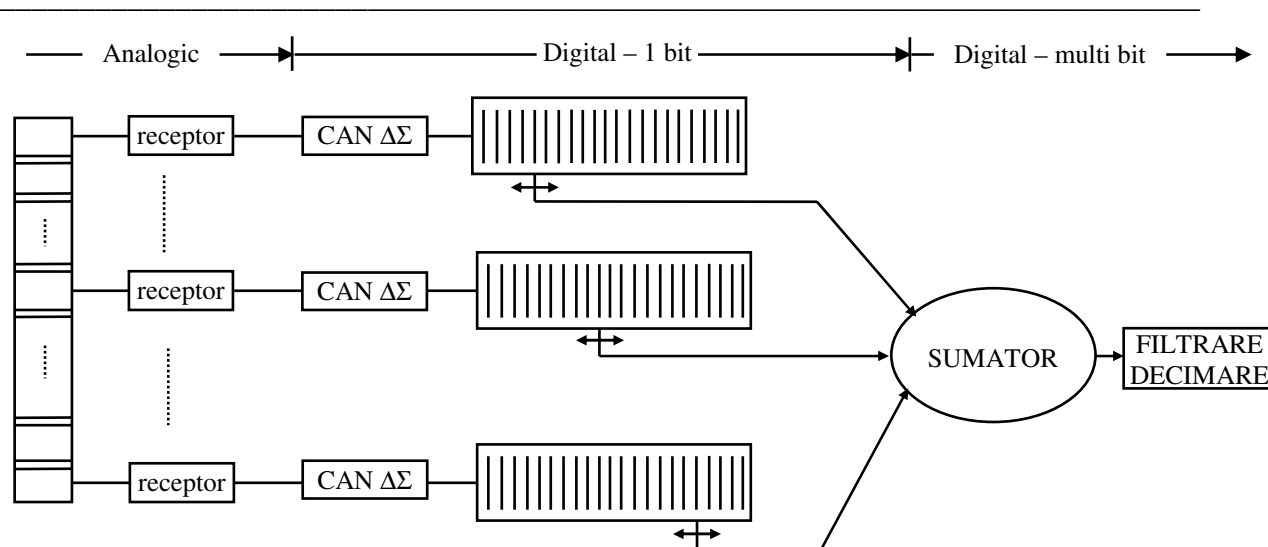


Figura 2.17 Formator de fascicul cu CAN $\Delta\Sigma$

2.4.CONCLUZII

Sinteza fasciculelor este o metodă de evaluare selectivă a semnalelor recepționate de o arie de traductoare de pe o direcție dorită simultan cu atenuarea răspunsului ariei pentru semnalele provenite de pe alte direcții. Procedura de sinteză permite o vizualizare multidimensională a mediului prin folosirea unei arii de traductoare adecvate care procesează după caz energie acustică sau electromagnetică. Ideea fundamentală pe care se bazează formarea de fascicul este realizarea condiției de însumare coerentă a semnalelor recepționate de elementele unei arii de la un punct din spațiu.

Se deosebesc două modalități de abordare a problematicii formării fasciculelor funcție de poziționarea punctului de interes. Astfel pentru câmpul apropiat unde frontul de undă este sferic formarea de fascicul presupune dirijarea după o direcție și focalizarea într-un punct dat pe direcția respectivă. În câmp îndepărtat unde frontul de undă poate fi considerat plan formarea fasciculelor se reduce la dirijarea după o direcție dată. Se poate concluziona de aici că structura formatoarelor se alege în funcție raportul între dimensiunea ariei și distanța de la aceasta la punctul de interes care reprezintă condiția de câmp apropiat / câmp îndepărtat pentru un domeniu de aplicabilitate dat.

Tipul undelor folosite și extinderea regiunii de interes raportată la dimensiunea ariei definesc domeniile de aplicabilitate și funcțiile formatorului:

Domeniul de aplicabilitate	Caracteristici de câmp	Tip de undă	Funcțiile formatorului
RADAR	Front plan / câmp îndepărtat	Electromagnetică	Dirijare
SONAR	Front plan / câmp îndepărtat	Acustică	Dirijare
IMAGISTICĂ MEDICALĂ	Front sferic / câmp apropiat	Acustică	Dirijare Focalizare

Pentru aplicațiile în câmp apropiat, dimensiunea regiunii de interes este comparabilă cu dimensiunea ariei, frontul de undă este sferic și acest lucru trebuie luat în considerare pentru determinarea exactă a întârzierilor cerute de însumarea coerentă. Acesta este cazul imagisticii medicale în care este necesară focalizarea fasciculelor pe obiecte într-un domeniu restrâns.

Implementarea condiției de însumare coerentă se face prin algoritmul cunoscut sub numele de întârziere și însumare. Practic compensarea diferențelor de drum între diferite

elemente ale aceleiași arii se poate realiza atât prin tehnici în domeniul timp cât și prin tehnici în domeniul frecvență. Pentru aplicațiile de imagistică medicală, care folosesc arii cu număr mare de elemente și necesită pentru scanarea în timp real un număr identic de canale de recepție, sunt preferate tehnicile în domeniul timp caracterizate printr-o complexitate mai scăzută a structurii canalului de procesare. **Tehnicile de formare a fasciculelor în domeniul timp sunt practic variante de implementare ale algoritmului de formare prin întârziere și însumare.** Trăsătura comună a tuturor acestora este realizarea cu acuratețe cât mai bună a condiției de însumare coerentă.

Evoluția tehnicilor de formare a fasciculelor ultrasonice este un proces care rulează în paralel cu dezvoltarea și perfecționarea resurselor de procesare. Scopul final al procesului de formare a fost întotdeauna același, însumarea de semnale întârziate cât mai precis, dar modalitatea de implementare a fost subordonată stadiului atins în concepția circuitelor electronice și a algoritmilor de prelucrare.

Pe aceste considerente în secțiunea a treia a capitolului se analizează evoluția formatoarelor de fascicul cu accent pe dezvoltările asociate cu tehnica digitală. Analiza evidențiază patru categorii de tehnici de formare digitală a fasciculelor care s-au dezvoltat succesiv pe principiul “*maximum de calitate cu mijloacele disponibile*”. Nu este mai puțin adevărat că alocarea de resurse pentru un domeniu de importanță majoră, cum este asistența medicală, a impulsionat direct perfecționarea resurselor de prelucrare prin concepția de circuite integrate pe scară foarte largă, dedicate investigării medicale. De asemenea diseminarea metodelor de prelucrare din domeniile strategice conexe investigării medicale, Radar și Sonar, în anii 90, au impulsionat dezvoltarea tehnicilor de formare a fasciculelor ultrasonice prin preluarea metodelor de dirijare și adaptarea lor la cerințele specifice de dirijare și focalizare.

Tehnicile de *formare digitală prin demodularea benzii de bază*, soluția folosită la sfârșitul anilor 80 reprezintă implementarea digitală a metodelor analogice de formare propuse în deceniul șapte. Practic migrarea spre domeniul digital, cu avantajele cunoscute (extinderea domeniului dinamic, viteză de procesare, putere de calcul, consum redus), s-a realizat în prima variantă prin conversia numerică a semnalului demodulat analogic. Banda de frecvență a acestuia corespunde cu ratele de eșantionare care se puteau realiza cu CAN-urile disponibile la momentul respectiv. Avantajul unei astfel de implementări față de sistemele analogice rezidă în aceea că în tehnică digitală modificarea întârzierii nu introduce zgomot și întârzierea poate fi modificată pentru a urmări îndeaproape valoarea necesară ceea ce lărgiște banda de lucru a unui astfel de sistem. Constituie un dezavantaj necesitatea includerii unui filtru analogic anti-aliere înaintea CAN pentru a elimina componentele de frecvență înaltă din semnalul de la ieșirea mixerului.

Creșterea frecvenței de operare a convertoarelor a permis trecerea la soluția cu demodulare numerică în cuadratură și îmbunătățirea preciziei întârzierilor prin folosirea unui algoritm de rotire a fazei (CORDIC) cunoscut din anii 50. În această tehnică de procesare a benzii de bază se exprimă întârzierea ca sumă de doi termeni: întârzierea brută reprezentată printr-un număr întreg de perioade de eșantionare și întârzierea fină care exprimă diferența în fracțiuni de perioadă de eșantionare între valoarea calculată și întârzierea brută. Activarea procesorului de rotire a fazei la multiplii ai frecvenței de eșantionare permite să se realizeze întârzierea fină cu o precizie bună.

Avantajele acestei soluții constau în aceea că nu mai sunt necesare componente analogice complexe și precise pentru a genera și selecta diferite valori ale fazei semnalului pentru mixare.

În paralel cu tehnicile de procesare prin demodularea benzii s-a dezvoltat *tehnica de generare și control digital a întârzierii dinamice prin interpolare*. În această tehnică se folosește interpolarea pentru a supraeșantiona semnalul digital livrat de CAN și a asigura eșantioane cu decalaj temporal suficient de fin între ele. Se reduc astfel pretențiile în ceea ce

privește frecvența de eșantionare a CAN prețul plătit fiind creșterea complexității secțiunii de procesare digitală. Într-o primă variantă s-a propus extinderea cu zerouri a secvenței eșantioanelor generate de CAN, pe fiecare canal, urmată de sinteza fasciculului prin întârziere și însumare la o rată înaltă și în final filtrarea și decimarea secvenței obținute.

O versiune interesantă a fost derivată din tehnica prin interpolare pornind de la observația că din punct de vedere al sintezei de fascicul funcția implementată de mecanismul interpolare / decimare poate fi îndeplinită și filtrând trece jos sau trece bandă secvența de eșantioane ne extinse cu zerouri. În acest caz fiecare întârziere necesită propriul său filtru, spre deosebire de metoda extinderii cu zerouri unde se selectează diferite secvențe de filtrare prin deplasarea eșantioanelor ne nule pe o versiune supraeșantionată a aceluiași filtru. Dacă se introduce condiția suplimentară ca toți coeficienții filtrelor să fie de forma $1/2^m$, unde m este întreg, complexitatea scade deoarece multiplicările se reduc la simple operații de deplasare binară (Metoda Lipschutz).

Spre deosebire de tehnicile care folosesc demodularea, metodele de interpolare – decimare sunt de bandă largă deoarece întârzie cu aceeași acuratețe toate frecvențele din banda de trecere a filtrului. Există desigur și în acest caz erori datorate trunchierii filtrului și acurateții coeficienților. Metoda Lipschutz este simplă datorită folosirii coeficienților de forma $1/2^m$ dar prezintă o flexibilitate limitată în asigurarea de întârzieri precise pentru semnale cu diferite purtătoare sau lățimi de bandă.

Tehnica de formare prin eșantionare neuniformă marchează focalizarea interesului în proiectarea echipamentelor imagistice ultrasonice asupra complexității sistemului. În această tehnică de generare a fasciculelor se folosește *eșantionarea neuniformă* a semnalului RF pentru a asigura acuratețea dorită a întârzierilor, păstrând frecvența de eșantionare la o valoare relativ redusă. Sunt prelevate doar eșantioanele necesare pentru focalizarea fasciculului ultrasonic în fiecare punct al imaginii. Pentru aceasta se generează semnale de tact diferite pentru fiecare element traductor, semnale care respectă criteriul Nyquist dar sunt întârziate unul față de altul astfel ca să se asigure focalizarea în punctul din imagine dorit. Cu alte cuvinte *eșantionarea și întârzierea sunt realizate simultan* și nu separat ca în tehnicile anterioare. În consecință frecvența de eșantionare este determinată de distanța dintre punctele imaginii sau mai general de rata minimă de eșantionare pentru detecția anvelopei semnalului focalizat. Aceasta duce la o reducere dramatică în cerințele de hardware comparativ cu metodele care folosesc eșantionarea uniformă convențională. În plus utilizarea memoriilor FIFO elimină operația de adresare pentru semnalele eșantionate rezultând în continuare o reducere a complexității structurii hardware a formatorului de fascicul. Elementul cheie al sistemului este blocul de generare a semnalelor de activare neuniformă a eșantionării.

Într-o variantă a *tehnicii de formare prin eșantionare neuniformă*, care reduce complexitatea structurii hardware, se prelevează eșantioane analogice din semnalele ecou și se formează fasciculul la nivel analogic. Ieșirile canalelor de recepție sunt eșantionate neuniform exact ca în cazul formatorului digital dar eșantioanele din semnal nu sunt cuantizate ca valori numerice ci reprezintă valori analogice instantanee. Prin urmare convertorul, memoria și circuitul de control a temporizării sunt înlocuite cu un FIFO Analogic (AFIFO) care realizează eșantionarea neuniformă și stocarea eșantioanelor analogice. Întârzierea necesară pentru focalizare este asigurată prin decalarea în timp a momentului eșantionării și nu prin selecția eșantionului potrivit dintr-o colecție de eșantioane prelevate uniform.

Formarea prin eșantionare analogică neuniformă deschide practic drumul spre realizarea unui formator de fascicul implementat într-un singur circuit integrat. Acest obiectiv poate deveni realitate odată cu definitivarea soluțiilor de realizare a unor circuite complexe programabile, în variantă mixtă, care integrează în același chip un număr suficient de mare de celule fundamentale analogice și digitale. Ca etapă intermediară, formarea prin eșantionare analogică neuniformă sugerează o structură de formator receptor implementabilă în trei circuite integrate:

- **circuitul analogic, ce conține blocurile de memorie FIFO analogică (unul pentru fiecare canal) și sumatorul analogic, realizabil sub forma unui ASIC**
- **convertorul analog numeric (într-un singur exemplar)**
- **blocul de comandă și control digital implementabil într-o structură logică programabilă (FPGA)**

Provocarea ultimului deceniu este îmbunătățirea rezoluției investigării, prin folosirea de arii cu număr mare de elemente (128 - 256), simultan cu păstrarea complexității sistemului și implicit a costurilor în limite rezonabile și cu asigurarea frecvenței de scanare cerute de aplicația concretă. Subordonată acestui obiectiv, în ultimii ani s-a dezvoltat o nouă tehnică de generare a fasciculelor de ultrasunete care presupune folosirea convertoarelor cu supraeșantionare în componența blocului digital de intrare, pentru a reduce complexitatea hardware.

Prin folosirea tehnicilor de supraeșantionare pot fi obținute imagini cu o calitate similară cu cea obținută prin formarea digitală multi-bit, în condițiile reducerii semnificative a complexității și gabaritului sistemului. Formatoarele care au la bază convertoare $\Delta\Sigma$ prezintă două avantaje majore față de metodele tradiționale multi bit:

- Simplificarea structurii CAN și posibilitatea integrării pe același chip împreună cu blocurile de procesare digitală pentru sinteza fasciculului. Circuitele necesare pentru a implementa CAN $\Delta\Sigma$ sunt mult mai simple decât cele folosite în CAN flash multi-bit fapt ce conduce la reducerea resurselor de interconectare, a gabaritului și puterii consumate.

- Realizarea de întârzieri precise prin manipularea eșantioanelor prelevate cu o rată de eșantionare ridicată. Circuitele complicate pentru procesare în banda de bază sau pentru interpolare / decimare sunt eliminate și o memorie FIFO convențională asigură întârzierea cu rezoluția dorită, înainte de însumare. În plus, însumarea pentru întreaga arie se simplifică întrucât primul nivel de însumare este alcătuit din sumatoare pe un bit.

Formarea prin supraeșantionare constituie într-un pas înainte spre obiectivul de realizare a unui formator de fascicul implementat într-un singur circuit integrat. Simplificarea considerabilă a structurii convertoarelor analog numerice permite redefinirea structurală a formatorului receptor pentru implementarea în două circuite integrate:

- **secțiunea analogică, ce conține convertoarele $\Delta\Sigma$ (unul pentru fiecare canal), realizabilă sub forma unui ASIC**
- **secțiunea digitală, alcătuită din: liniile digitale de întârziere, sumator, decimator și blocul de comandă și control , implementabilă într-o structură logică programabilă (FPGA)**

Comparativ cu formarea prin eșantionare analogică neuniformă, folosirea conversiei $\Delta\Sigma$ reduce complexitatea secțiunii analogice în favoarea celei digitale, care beneficiază la acest moment de resurse mult mai puternice pentru implementare.

Abordarea domeniului formării digitale de fascicul în prezenta lucrare are în vedere tocmai evaluarea potențialului tehnicilor de formare prin supraeșantionare și decelarea posibilităților de perfecționare prin:

- identificarea unei structuri de **convertor** $\Delta\Sigma$ convenabilă din punct de vedere al raportului complexitate / performanță
- implementarea unui mecanism de generare “on line” a întârzierilor pornind de la un set minim de date de intrare și reducerea pe această cale a capacității de memorare necesare.
- optimizarea procedurii de sumare – filtrare – decimare din punct de vedere al vitezei de operare și al necesarului de resurse.

CAPITOLUL 3

CUANTIFICAREA

Imagistica ultrasonică, la fel ca alte aplicații din electronică, s-a dezvoltat treptat de la procesarea analogică la cea digitală, datorită gamei dinamice largi posibilă prin folosirea calculului în virgulă fixă sau mobilă, îmbunătățirii vitezei și optimizării consumului de putere la circuitele numerice moderne. Conversia, între mediul real, fundamental analogic și domeniul numeric, unde se concentrează în cea mai mare parte procesarea, necesită convertoare analog numerice și numeric analogice (CAN și CNA) de viteză ridicată și precizie. Aplicațiile ultrasunetelor reprezintă una din tehnologiile care au condus, în anii '80, la dezvoltarea CAN-urilor de 8 biți pentru frecvențe de ordinul MHz. Înaintea apariției acestor convertoare, producătorii de sisteme imagistice ultrasonice nu au avut altă posibilitate decât să prelucreze semnalele în circuite analogice și să folosească limitările gamei dinamice impuse de aceste circuite.

Astăzi CAN-urile multibit sunt larg răspândite, astfel că producătorii din diverse domenii au acum la dispoziție o gamă largă de rate de eșantionare și de mărimi ale cuvântului de ieșire.

Frecvența purtătoare a semnalelor ultrasonice, de 5÷10MHz, cere o rată de eșantionare de peste 30MHz și un domeniu dinamic minim de intrare de ordinul 50÷60dB. Alegerea unui CAN ieftin este limitată de necesitatea folosirii de CAN-uri de 8÷12 biți, la o frecvență de eșantionare de 40MHz. Există CAN-uri speciale, care oferă frecvențe de eșantionare de ordinul GHz cu 10 biți sau mai mulți, dar acestea sunt scumpe, sunt consumatoare de putere și greu de integrat într-un sistem ultrasonic. De fapt, în loc să crească rata de eșantionare, majoritatea firmelor s-au concentrat pe realizarea unui consum de putere scăzut. Probabil că piața sau tehnologia de fabricație sunt de așa natură încât CAN-urile de 10 biți, la 100MHz nu sunt practice din punct de vedere al prețului. Ca rezultat formatoarele digitale de fascicul ultrasonic ce se bazează pe CAN-uri multibit sunt limitate de o rată de eșantionare redusă, ceea ce presupune folosirea de tehnici de interpolare / decimare, de deplasare a benzii de bază sau de eșantionare neuniformă pentru a asigura întârzierea cu precizie adecvată.

Câștigă încredere pe piață un alt tip de CAN care utilizează metode de supraeșantionare și un compromis între numărul de biți și frecvența de eșantionare. Un modulator delta-sigma, $\Delta\Sigma$ este un tip de CAN cu doar câțiva biți de ieșire, deseori numai unul singur, care operează la o rată ridicată. Aceste CAN-uri sunt mai simple în multe privințe față de CAN-urile flash și pot asigura un nivel dinamic bun pentru semnale de frecvențe joase. Majoritatea CAN-urilor $\Delta\Sigma$ disponibile pe piață filtrează și decimează ieșirea modulatorului $\Delta\Sigma$ pentru a asigura o ieșire multibit "lentă", similară cu CAN-urile flash standard. Interesul pentru aceste modulatatoare în aplicația dată este bazat pe rata de eșantionare ridicată necesară pentru performanțe bune din punct de vedere al rezoluției, și de asemenea pe structura simplă a cuantificatorului cu un singur bit. Ambele caracteristici sunt folositoare în formatoarele de fascicul ultrasonic pentru că:

- 1). nu necesită circuite speciale pentru a asigura cu precizie întârzierea adecvată;
- 2). simplifică circuitele formatorului prin reducerea numărului de biți ai magistralei de date.

3.1. CONVERTOARE RAPIDE MULTI - BIT

Conversia analog digitală presupune eșantionarea temporală a semnalului de intrare, de obicei cu un circuit de eșantionare memorare, și cuantificarea amplitudinii eșantionului prin asignarea unui cod numeric unic pentru un domeniu de tensiuni. Convertoarele multi –

bit folosesc în mod uzual eşantionarea uniformă în timp şi amplitudine; totuşi aşa cum s-a arătat în capitolul precedent, acest tip de eşantionare nu este întotdeauna ideal pentru imagistica ultrasonică, fapt reflectat de metodele de eşantionare neuniformă propuse din motive de performanţă în aceste aplicaţii. CAN – urile obişnuite alocă N biţi pentru a cuantiza $Q = 2^N$ sau $2^{N-1} + 1$ niveluri funcţie de cum se doreşte ieşirea: cu front central (mid-riser), aşa ca în fig. 3.1.A, sau simetrică (mid-tread), cum este arătat fig. 3.1.B, pentru $N=2$.

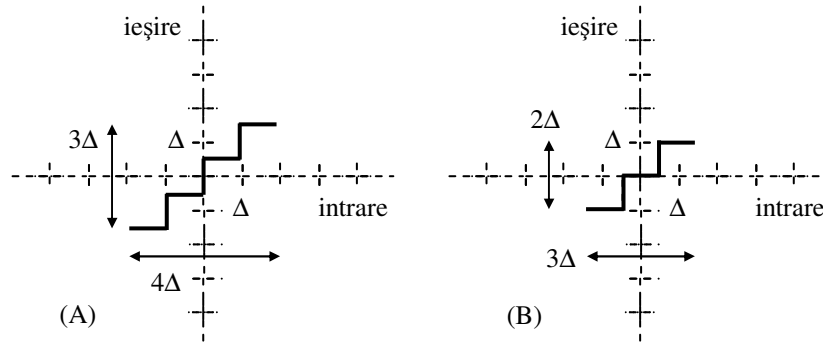


Figura 3.1. Caracteristici de transfer pentru CAN multi-bit

Cuanta Δ este definită ca $\Delta = V_{vv}/(Q-1)$, unde V_{vv} este amplitudinea de ieşire vârf la vârf. Uzual este preferată ieşirea cu front central deoarece pentru semnale apropiate de zero aceasta comută pe când cuantificatorul cu ieşire simetrică forţează nivelul constant zero pentru aceste niveluri. De asemenea tehnica de cuantificare cu front central exploatează avantajul a 2^N niveluri de cuantificare în timp ce cuantificatorul “mid-tread” este asimetric sau pierde niveluri de ieşire. Acest aspect este foarte important pentru imagistica structurilor în mişcare (eco-cardiografie) unde un ofset de curent continuu egal cu $\Delta/2$ poate transforma un cuantificator “mid-riser” în unul “mid-tread” care poate reduce la zero semnalele mici (ex: ecourile furnizate de sânge).

Procesul de cuantificare este puternic neliniar, prin urmare în analiza statistică se modelează de obicei cuantificatorul printr-o sursă de zgomot alb aditiv, independentă de intrare. Această aproximare se bazează pe următoarele proprietăţi ale secvenţei eroare de cuantificare $e[n]$:

- $e[n]$ este din punct de vedere statistic independentă de semnalul de intrare $x[n]$ (sunt necorelate).
- $e[n]$ este uniform distribuită în domeniul $[-\Delta/2, \Delta/2]$.
- $e[n]$ este identic independent distribuită (are un spectru de putere plat).

Aceste proprietăţi sunt valabile pentru intrări nesaturate, cuantificator cu un număr mare de biţi şi secvenţă de eşantioane de intrare necorelate. Evident, există situaţii când unele din aceste condiţii nu sunt îndeplinite, în special în cazul cuantificatoarelor pe un bit din modulatoarele $\Delta\Sigma$ care vor fi discutate mai târziu. Presupunând aceste condiţii valabile pentru cuantificatoarele multi – bit, eroarea pătratică medie sau dispersia zgomotului de cuantificare este:

$$\sigma_e^2 = \frac{1}{\Delta} \int_{-\Delta/2}^{+\Delta/2} e^2 de = \frac{\Delta^2}{12} = \frac{\left(\frac{V_{vv}}{2^N - 1}\right)^2}{12} \cong \frac{\left(\frac{V_{vv}}{2^N}\right)^2}{12} \quad (3.1)$$

Se presupune că distribuţia spectrală de putere pentru eroarea de cuantificare este uniformă în frecvenţă de la $-f_s/2$ la $+f_s/2$. Densitatea constantă a puterii spectrale pentru acest zgomot este:

$$P_e(f) = \sigma_e^2 / f_s \quad (3.2)$$

Rata de supraeșantionare (OSR) se definește prin raportul dintre frecvența de eșantionare și dublul frecvenței limită a benzii semnalului, $2f_B$ (frecvența Nyquist). Aceasta este:

$$OSR = f_s/(2f_B) = 1/(2f_B T) \quad (3.3)$$

unde T este perioada de eșantionare. Dacă rata de supraeșantionare crește, zgomotul de cuantificare poate fi distribuit într-un domeniu larg de frecvențe, și ca urmare puterea de zgomot în bandă se modifică. Folosind (3.3) se poate scrie puterea zgomotului, în banda limitată a semnalului $(-f_B + f_B)$, astfel:

$$\sigma_{ey}^2 = \int_{-f_B}^{f_B} P_e(f) df = 2 \int_0^{f_B} P_e(f) df = \sigma_e^2 (2f_B T) = \frac{\sigma_e^2}{OSR} \quad (3.4)$$

Aceasta arată faptul binecunoscut că prin creșterea frecvenței de eșantionare se reduce zgomotul de cuantificare în bandă. Dacă se exprimă termenii în decibeli (dB), se observă că la fiecare dublare a ratei de supraeșantionare (prin reducerea benzii semnalului sau prin creșterea ratei de eșantionare) zgomotul în bandă scade cu 3 dB.

Raportul semnal zgomot (SNR – signal to noise ratio), sau mai precis raportul semnal – zgomot de cuantificare (SQNR - signal to quantization noise ratio) pentru un CAN multi – bit se obține ca logaritmul raportului dintre puterea semnalului σ_x^2 și puterea zgomotului de cuantificare dată de (3.1). Acesta se poate scrie astfel:

$$SNR = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) = 10 \log_{10} \left(\frac{\sigma_x^2}{V_{vv}^2} \right) + 10,8 + 6,02N \quad (3.5)$$

unde s-a folosit valoarea aproximativă din (3.1) pentru a obține dependența de numărul de biți din cuantizator N . Se observă că pentru fiecare bit adăugat la ieșire se obține o îmbunătățire a SNR cu 6 dB. Datorită dependenței directe între N și SNR, performanțele de zgomot ale CAN – urilor sunt comparate adesea prin evaluarea numărului efectiv de biți la ieșire. De exemplu, îmbunătățirea SNR cu 3 dB ca rezultat al dublării frecvenței de eșantionare, poate fi exprimată și ca adăugarea unei jumătăți de bit la precizia CAN.

O ultimă caracteristică de interes a CAN este *domeniul dinamic*, DR (Dynamic Range), care definește semnalul maxim de intrare raportat la semnalul minim detectabil peste nivelul zgomotului de cuantificare. Un SNR de zero dB definește intrarea sinusoidală minimă având aceiași putere în bandă cu zgomotul de cuantizare (σ_e^2), și ca atare fiind dificil de detectat. Semnalul sinusoidal maxim de intrare are puterea $\sigma_x^2 = V_{vv}^2/8$, iar domeniul dinamic se poate obține introducând această valoare în (3.5).

$$DR = 6,02N + 1.76 \text{ (dB)} \quad (3.6)$$

Această relație indică din nou o dependență explicită de numărul biților cuantificatorului. Pentru CAN – urile multi – bit care operează la frecvența Nyquist valoarea de vârf a SNR corespunde cu domeniul dinamic, fapt în general neadevărat pentru CAN sigma – delta. De asemenea pentru convertoarele multi – bit cu cuantificare neuniformă valoarea de vârf a SNR nu corespunde cu domeniul dinamic întrucât pentru niveluri reduse de semnal se reduce valoarea cuantei fapt ce înseamnă detectarea de semnale mai mici și implicit domeniu dinamic de valoare mai mare.

3.2. CAN $\Delta\Sigma$

3.2.1 TEHNICI DE ÎMBUNĂTĂȚIRE A REZOLUȚIEI

Convertoarele analog numerice pot atinge rezoluții înalte prin folosirea tehnicilor de supraeșantionare și de "formare a zgomotului". (noise shaping) Supraeșantionarea, adică eșantionarea semnalului la frecvențe mult peste limita Nyquist, împrăștie zgomotul de cuantificare într-un domeniu larg de frecvențe și reduce astfel cantitatea de zgomot în banda semnalului. Majoritatea zgomotului este împins în afara benzii și poate fi atenuat cu un filtru digital.

3.2.2 SUPRAEȘANTIONAREA

Pentru eșantionarea la frecvența Nyquist când banda semnalului $f_B = f_s/2$, puterea de zgomot este conținută în întregime în banda semnalului. În cazul supraeșantionării, aceeași putere de zgomot este distribuită într-un domeniu de frecvență dat de frecvența de eșantionare f_s , care este mult mai mare decât banda semnalului f_B . Numai o fracțiune redusă din puterea de zgomot cade în banda semnalului, iar zgomotul din afara benzii poate fi eliminat cu un filtru digital trece jos cu o caracteristică abruptă.

Parametrul principal folosit pentru evaluarea performanțelor convertoarelor este și în acest caz raportul semnal zgomot (SNR) definit ca raport între puterea semnalului la ieșire σ_{xy}^2 și puterea zgomotului la ieșire σ_{ey}^2 . Pentru supraeșantionare, valoarea maximă pentru SNR se scrie:

$$\text{SNR}_{\text{peak}} = 10 \log \left(\frac{\sigma_{xy}^2}{\sigma_{ey}^2} \right) = 6,02N + 1,76 + 10 \log(\text{OSR}) \quad (3.7)$$

Din relația (3.7) se observă că o creștere a rezoluției cu un bit reclamă multiplicarea cu 4 a OSR. Extinderea rezoluției convertorului pe aceasta cale cu 5 biți conduce la valori impracticabile ale OSR ($10^{0,5*6,02} = 1024$). Supraeșantionarea poate reduce zgomotul din banda semnalului dar nu îmbunătățește liniaritatea convertorului. Deoarece necesită factori de supraeșantionare mari și cuantificatori riguroși liniari, supraeșantionarea de sine stătătoare nu este o cale potrivită pentru a obține convertoare de înaltă rezoluție

3.2.3 FORMAREA ZGOMOTULUI

Formarea zgomotului în convertoarele analog numerice presupune utilizarea de funcții de transfer diferite pentru semnal și respectiv pentru zgomotul de cuantificare. Modulatoarele $\Delta\Sigma$ sunt o clasă de modulatori cu reacție cu ajutorul cărora pot fi implementate funcțiile de transfer necesare formării zgomotului. Structura principală a unui modulator $\Delta\Sigma$ și modelul său liniar sunt date în fig. 3.1. $H(z)$ este un filtru în timp discret, cu câștig foarte mare la frecvențe joase în banda de interes a cărei ieșire este convertită în valoare numerică de către un cuantificator. Semnalul de reacție este asigurat de un CNA plasat în bucla de reacție. Semnalele de eroare introduse de cuantificator și CNA sunt e_Q și e_{CNA} . Folosind principiul superpoziției ieșirea sistemului din fig. 3.1 poate fi exprimată astfel:

$$Y(z) = \frac{H(z)}{1 + H(z)} X(z) + \frac{1}{1 + H(z)} E_Q(z) - \frac{H(z)}{1 + H(z)} E_{\text{CNA}}(z) \quad (3.8)$$

Se definesc funcțiile de transfer pentru semnal $\text{STF}(z)$, zgomot $\text{NTF}(z)$ și convertorul numeric analogic $\text{NTF}_{\text{CNA}}(z)$ prin relațiile:

$$STF(z) = \frac{H(z)}{1 + H(z)} \quad (3.9)$$

$$NTF(z) = \frac{1}{1 + H(z)} \quad (3.10)$$

$$NTF_{CNA}(z) = -\frac{H(z)}{1 + H(z)} \quad (3.11)$$

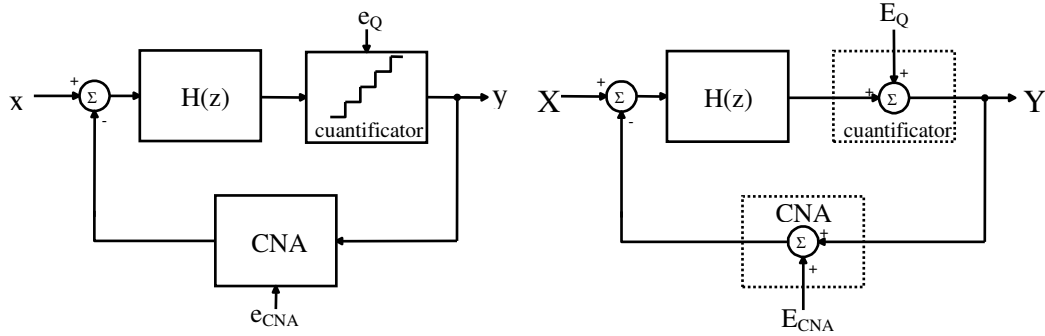


Fig. 3.1 Modulator $\Delta\Sigma$, Modelul linear.

Zerourile pentru $NTF(z)$ sunt polii lui $H(z)$, altfel spus când $H(z)$ tinde la infinit din relația (3.10) se observă că $NTF(z)$ tinde la zero. Prin combinarea reacției și a câștigului mare al lui $H(z)$ în banda de interes putem modela zgomotul de cuantificare în forma dorită. Pentru o astfel de alegere, adică un câștig $H(z)$ mare în banda semnalului, $STF(z)$ va fi unitar în bandă iar $NTF(z)$ va tinde la zero. Prin urmare zgomotul de cuantificare este atenuat în banda de trecere în timp ce semnalul este practic neafectat. În aceleași condiții relația (3.11) arată că $NTF_{CNA}(z)$ tinde la -1 . Prin urmare erorile CNA constituie o problemă importantă deoarece se însumează direct la ieșirea modulatorului și nu pot fi distinse. Rezolvarea problemei constă în asigurarea liniarității CNA-ului ceea ce constituie o cerință severă pentru aplicațiile de înaltă rezoluție cu cuantificatoare multibit.

3.2.4.CNA PE UN BIT

Avantajul unui CNA pe un bit este că acesta este inerent liniar. Deoarece un CNA de un bit are doar două niveluri de ieșire, iar două puncte determină o linie dreaptă, orice deviație de la valorile ideale introduce pur și simplu un offset sau o eroare de câștig în locul neliniarității. Acesta este motivul major de a utiliza tehnici de supraeșantionare cu cuantificare pe un bit.

3.2.5 CAN CU SUPRAEȘANTIONARE

Arhitectura de bază a unui convertor $\Delta\Sigma$ cu supraeșantionare este dată în fig.3.2. Primul etaj este un filtru antialiasing folosit pentru a limita banda semnalului de intrare la frecvențe sub jumătatea frecvenței de supraeșantionare, f_s . Datorită frecvențelor de supraeșantionare mari un filtru simplu RC este suficient pentru funcția de antialiasing. Semnalul continuu $x_c(t)$ de la ieșirea filtrului este eșantionat într-un circuit de eșantionare memorare și apoi procesat de modulatorul $\Delta\Sigma$, care convertește semnalul analogic într-un semnal digital cu rezoluție redusă și zgomot distribuit. Filtrul digital trece jos conectat la ieșirea modulatorului atenuază puternic zgomotul din afara benzii iar semnalul obținut este reeșantionat la rata Nyquist fără ca această operație să afecteze SNR. Filtrul trece jos plus blocul de reeșantionare alcătuiesc o structură numită decimator care convertește semnalul supraeșantionat de rezoluție

redușă într-un semnal digital cu rezoluție ridicată și eșantionat la frecvență Nyquist. Trebuie menționat că dacă modulatorii $\Delta\Sigma$ sunt implementate cu circuite cu capacități comutate, nu este necesar un circuit de eșantionare memorare distinct deoarece semnalul continuu este inherent eșantionat prin comutatoarele și capacitățile din structura modulatorului.

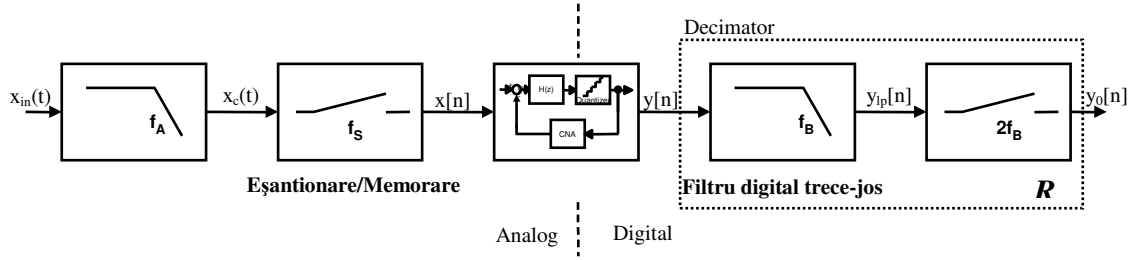


Fig. 3.2 CAN $\Delta\Sigma$ cu supraeșantionare

3.3 MODULATOARE $\Delta\Sigma$ DE ORDINUL 1

Structura unui modulator $\Delta\Sigma$ de ordinul 1 este prezentată în fig. 3.3. Modulatorul este alcătuit dintr-un integrator, un cuantificator și un CNA care asigură reacția. Semnalul cuantificat este versiunea filtrată a diferenței între intrarea $x[n]$ și reprezentarea analogică $y_a[n]$ a ieșirii modulatorului $y[n]$. Filtrul este un integrator în timp discret a cărui caracteristică de transfer este $z^{-1}/(1-z^{-1})$. Dacă CNA-ul este ideal ieșirea modulatorului $y[n]$ este dată de relația:

$$Y(z) = X(z)z^{-1} + E_q(z)(1-z^{-1}) \quad (3.12)$$

adică $STF(z) = z^{-1}$ și $NTF(z) = (1-z^{-1})$. Ieșirea este versiunea întârziată a intrării plus zgomotul de cuantificare format printr-un filtru trece jos. Deoarece $NTF(z)$ are un zero pentru $z=1$, adică la frecvența de DC, atenuarea $NTF(z)$ este infinită în curent continuu. Panta caracteristicii de transfer a zgomotului este de 20dB/decadă.

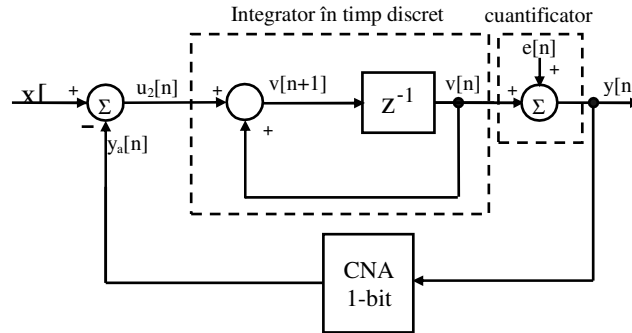


Fig. 3.3 Modulator $\Delta\Sigma$ de ordinul 1

Pentru a calcula valoarea de vârf a SNR, care indică de fapt domeniul dinamic, (DR), al convertorului, se determină mai întâi amplitudinea $NTF(z)$. Pentru $z = e^{sT} = e^{j2\pi f/f_s}$ obținem:

$$|NTF(z)| = |1 - z^{-1}| = |1 - e^{-j2\pi f/f_s}| = \left| \frac{e^{j\pi f/f_s} - e^{-j\pi f/f_s}}{2j} \cdot 2j \cdot e^{-j\pi f/f_s} \right| = 2 \sin(\pi f/f_s) \quad (3.13)$$

Puterea zgomotului la ieșirea modulatorului este:

$$\sigma_{ey}^2 = \int_{-f_B}^{f_B} P_{ey}(f) |NTF(z)|^2 df = \frac{1}{f_s} \frac{\Delta^2}{12} \int_{-f_B}^{f_B} |2 \sin(\pi f/f_s)|^2 df$$

dacă $OSR \gg 1$, $f_s \gg f$ se poate aproxima $\sin(\pi f / f_s) = \pi f / f_s$ și rezultă:

$$\sigma_{ey}^2 \approx \frac{1}{f_s} \frac{\Delta^2}{12} \int_{-f_B}^{f_B} [2(\pi f / f_s)]^2 df = \frac{\Delta^2}{12} \frac{\pi^2}{3} \frac{1}{(OSR)^3} \quad (3.14)$$

Pentru un cuantificator cu N biți și o valoare a cuantei Δ puterea semnalului la ieșire σ_{xy}^2 are valoarea:

$$\sigma_{xy}^2 = \left(\frac{\Delta \cdot 2^N}{2\sqrt{2}} \right)^2 = \frac{\Delta^2 \cdot 2^{2N}}{8} \quad (3.14a)$$

Valoarea de vârf a SNR este dată de:

$$SNR_{peak} = 10 \cdot \log \left(\frac{\sigma_{xy}^2}{\sigma_{ey}^2} \right) = 10 \cdot \log \left(\frac{3}{2} 2^{2N} \right) + 10 \cdot \log \left(\frac{3}{\pi^2} \cdot (OSR)^3 \right) \quad (3.15)$$

Relația arată că la dublarea OSR domeniul dinamic crește cu 9dB. Deși valoarea maximă a SNR se îmbunătățește semnificativ față de relația (3.7) totuși este necesar un OSR de valoare relativ ridicată pentru a obține un SNR de 16 biți (≈ 100 dB) cu un modulator de ordinul 1.

Modelarea în domeniul timp

Modulatorul $\Delta\Sigma$ este în principiu un sistem cu reacție, care încearcă să forțeze semnalul de ieșire $y[n]$ să devină egal cu semnalul de intrare $x[n]$. În domeniul timp, pentru structura din fig.3.3 se poate scrie:

$$v[n] = u[n-1] + v[n-1]$$

$$y[n] = \begin{cases} 1, & v[n] \geq 0 \\ -1, & v[n] < 0 \end{cases} \quad (3.16)$$

$$y_a[n] = y[n]$$

$$u[n] = x[n] - y_a[n]$$

Deoarece ieșirea $y[n]$ poate lua doar două valori 1 și -1, nu va egala intrarea decât în cazul când intrarea are una din aceste două valori. Ca urmare o eroare, $u[n]$, va exista întotdeauna. Dacă ieșirea este la valoarea $y[n]=1$, $y[n]$ este mai mare decât intrarea, eroarea este negativă și prin urmare integratorul acumulează valori negative care se regăsesc în $v[n]$. După un număr de perioade de tact valoarea acumulată de integrator produce comutarea cuantificatorului la valoarea $y[n]=-1$, eroarea devine pozitivă și se reia procesul de acumulare care după un număr de perioade de tact va readuce ieșirea la valoarea $y[n]=1$, s.a.m.d. Ieșirea modulatorului este un șir de biți a căror valoare medie este aproximația digitală a intrării, $x[n]$.

Fig.3.4 indică ieșirea modulatorului pentru un semnal de intrare sinusoidal cu frecvența de 1kHz și amplitudine 0,5. Informația de amplitudine pentru sinusoidă este codificată în numărul de valori 1 relativ la numărul de valori -1.

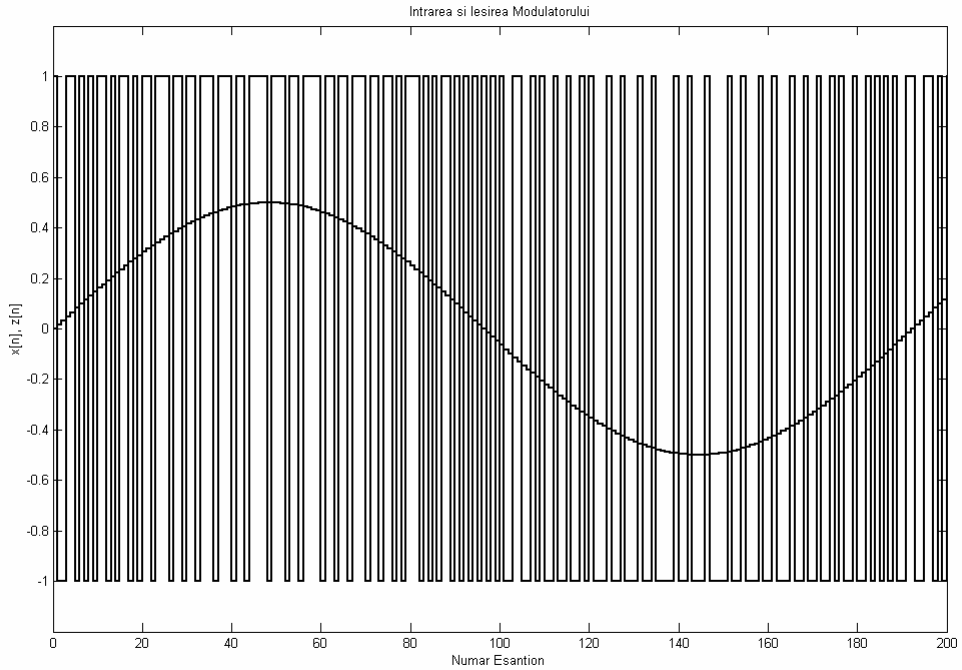


Fig. 3.4 Ieșirea modulatorului $\Delta\Sigma$ pentru o intrare sinusoidală

3.4 MODULATOARE $\Delta\Sigma$ DE ORDINUL 2

Un modulator $\Delta\Sigma$ de ordinul 2 este prezentat în fig. 3.5. Structura conține două integratoare cu întârziere pe calea directă, primul precedat de o atenuare de 0,5 iar al doilea de o amplificare egală cu 2. Ieșirea modulatorului este dată de:

$$Y(z) = X(z)z^{-2} + E_Q(z)(1 - z^{-1})^2 \quad (3.17)$$

adică $STF(z) = z^{-2}$ și $NTF(z) = (1 - z^{-1})^2$.

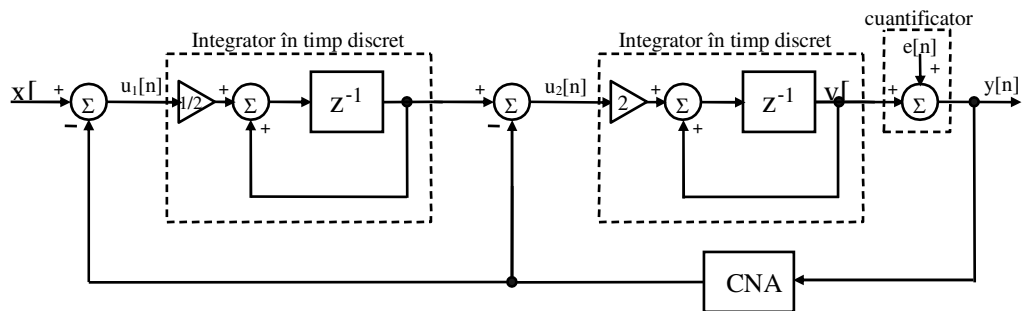


Fig. 3.5 Modulator $\Delta\Sigma$ de ordinul II

Comparativ cu NTF de ordinul întâi caracteristica de ordinul doi suprimă într-o măsură mai mare zgomotul de cuantificare la frecvențe joase și amplifică zgomotul exterior benzii semnalului, adică o fracțiune mai mare din puterea de zgomot este împinsă la frecvențe înalte. Valoarea maximă a SNR se deduce într-o manieră similară și se obține:

$$\text{SNR}_{\text{peak}} = 10 \cdot \log \left(\frac{\sigma_{xy}^2}{\sigma_{ey}^2} \right) = 10 \cdot \log \left(\frac{3}{2} 2^{2N} \right) + 10 \cdot \log \left(\frac{5}{\pi^4} \cdot (\text{OSR})^5 \right) \quad (3.18)$$

Prin urmare în această situație dublarea OSR conduce la creșterea domeniului dinamic cu 15dB (2,5 biți) și obținerea unui domeniu dinamic de 100dB (16 biți) necesită pentru OSR o valoare mult mai mică decât în cazul modulatorului de ordinul întâi (165 față de 1700).

Modulatoare $\Delta\Sigma$ de ordin superior

Modulatoarele $\Delta\Sigma$ cu modelarea zgomotului de ordin superior pot asigura rezoluții înalte prin mutarea unei cantități importante de zgomot în afara benzii semnalului. Prin generalizarea deducției din secțiunea precedentă se poate arăta că la un modulator de ordinul L:

$$\text{SNR}_{\text{peak}} = 10 \cdot \log \left(\frac{\sigma_{xy}^2}{\sigma_{ey}^2} \right) = 10 \cdot \log \left(\frac{3}{2} 2^{2N} \right) + 10 \cdot \log \left(\frac{2L+1}{\pi^{2L+1}} \cdot (\text{OSR})^{2L+1} \right) \quad (3.19)$$

adică SNR crește cu 6L+3 dB la dublarea OSR

3.5 ARHITECTURI CU O SINGURĂ BUCLĂ

Modulatorul $\Delta\Sigma$ de ordinul doi prezentat în figura 3.5 este larg utilizat deoarece este simplu de implementat și insensibil la asimetriile componentelor. Problema modulatorului de ordinul 2 este factorul de supraeșantionare ridicat necesar pentru a asigura cerința de domeniu dinamic. În conformitate cu relația (3.18) un domeniu dinamic de 12 biți (72 dB) necesită un factor de supraeșantionare OSR=56, un domeniu dinamic de 14 biți (84 dB) necesită un OSR=97 iar un domeniu de 16 biți (96 dB) necesită un OSR=169. Prin urmare acest tip de modulator nu este potrivit pentru aplicațiile de mare viteză.

Deplasarea dincolo de ordinul 2 se face cu precauții deosebite în ceea ce privește stabilitatea. O arhitectură monobucă de ordinul 4, stabilă este indicată în figura 3.6. Au fost scalate câștigurile integratoarelor și s-au plasat limitatoare pe ieșirile lor pentru a se asigura stabilitatea la supraîncărcare sau la conectare. Această arhitectură asigură 14 biți rezoluție la un factor de supraeșantionare egal cu 64.

O extensie imediată a arhitecturii din fig.3.6 se obține prin adăugarea unui CNA multibit pentru a îmbunătăți domeniul dinamic al modulatorului. Acest modulator de ordinul 4 asigură 14 biți rezoluție la o rată de supraeșantionare de 16 dacă folosim un CNA cu 4 biți. În această arhitectură integratoarele au o caracteristică de transfer fără întârziere pentru stabilizarea buclei, iar câștigurile sunt selectate pentru a asigura compromisul între stabilitate și domeniul dinamic. Neajunsul evident al acestei abordări constă în necesitatea calibrării convertorului pe 4 biți pentru linearitate pe 14 biți.

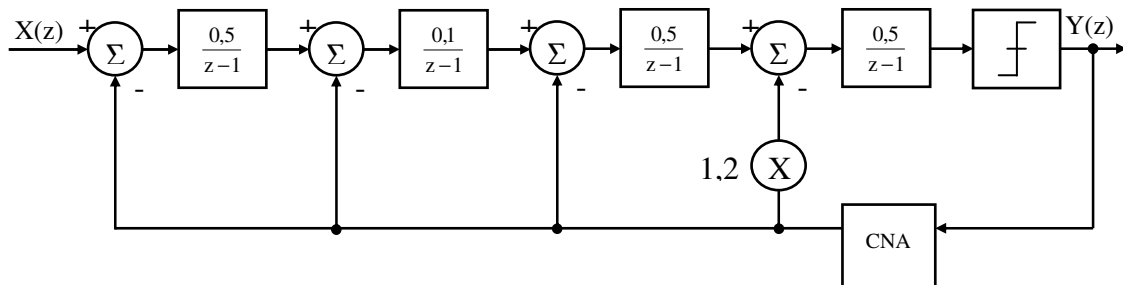


Fig. 3.6 Modulator $\Sigma\Delta$ de ordinul IV

3.6 ARHITECTURI INTERPOLATIVE

O abordare alternativă la modulatorile de ordin superior distribuie zerourile caracteristicii de zgomot mai degrabă în banda semnalului decât în domeniul de DC. Dacă este realizată corect această abordare poate deplasa mai eficient zgomotul de cuantizare în afara benzii de interes. Cunoscutele la modul general ca arhitecturi interpolative, aceste modulatori folosesc tehnicile standard de filtrare cu capacități comutate pentru a forma zgomotul în mod dorit. De exemplu arhitectura interpolativă din figura 3.7 asigură 16 biți rezoluție la o rată de supraeșantionare de 64.

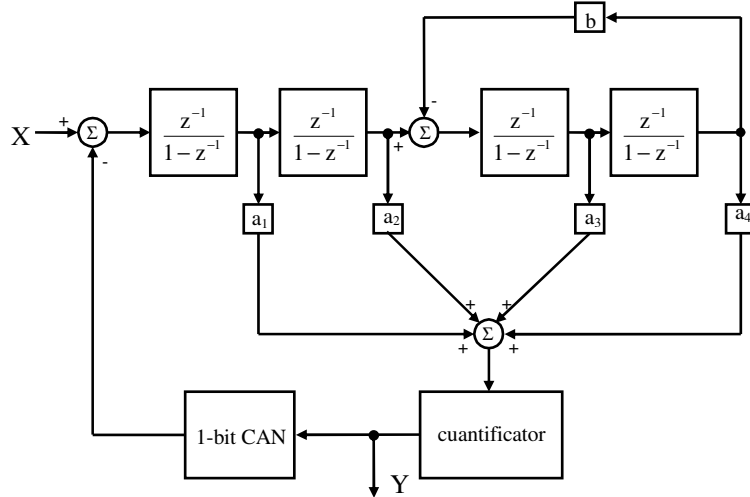


Fig. 3.7 Modulator $\Delta\Sigma$ de ordinul IV – arhitectură interpolativă

Arhitectura interpolativă din fig.3.7 poate fi extinsă pentru ordine mai mari. De exemplu, un modulator de ordinul șapte cu CNA pe trei niveluri asigură 19 biți rezoluție la o rată de supraeșantionare de 64. Această performanță este doar cu trei biți mai bună decât cea a unui modulator de ordinul patru operând cu aceeași rată de supraeșantionare ceea ce sugerează o îmbunătățire mai redusă a performanței la creșterea ordinului modulatorului pentru acest tip de arhitectură.

3.7 MODULATOARE $\Delta\Sigma$ ÎN CASCADĂ

Modulatorile $\Delta\Sigma$ de ordin superior pot fi implementate prin conectarea în cascadă a unor modulatori de ordin redus. Cascadarea nu afectează stabilitatea modulatorului în ansamblul său dacă etajele considerate separat sunt stabile. Acest tip de structură este numit și MASH (Multi-stage noise Shaping).

La modulatorile $\Delta\Sigma$ în cascadă, intrarea fiecărui etaj este comandată de o valoare ponderată a zgomotului de cuantificare al etajului precedent. Ieșirea fiecărui etaj poate fi utilizată pentru a elimina zgomotul de cuantificare al etajului precedent. Astfel rămâne numai zgomotul de cuantificare al ultimului etaj, care va fi atenuat de caracteristica NTF de ordin egal cu ordinul structurii cascada în ansamblu. Un exemplu de arhitectură de ordinul trei este indicat în fig.3.8. Este o arhitectură de tip 2-1 constând dintr-un etaj de ordinul doi urmat de un etaj de ordinul întâi.

Dacă $b=2$ ieșirea primului etaj al modulatorului este:

$$Y_1(z) = X(z)z^{-2} + E_{Q1}(z)(1 - z^{-1})^2 \quad (3.20)$$

iar ieșirea celui de al doilea etaj $Y_2(z)$ este:

$$Y_2(z) = X_2(z)z^{-2} + E_{Q2}(z)(1 - z^{-1})^2 \quad (3.21)$$

unde

$$X_2(z) = \beta[(1 - \lambda)Y_1(z) - E_{Q1}(z)] \quad (3.22)$$

Ieșirea globală a arhitecturii 2-1, $Y(z)$ este dată de:

$$Y(z) = H_1(z)Y_1(z) + H_2(z)Y_2(z) \quad (3.23)$$

Filtrele digitale $H_1(z)$ și $H_2(z)$ se aleg astfel încât să se elimine eroarea de cuantificare a primului etaj $E_{Q1}(z)$. Funcțiile care îndeplinesc această cerință sunt:

$$H_1(z) = z^{-1} - (1 - \hat{\lambda})(1 - z^{-1})^2 z^{-1} \quad (3.24)$$

$$H_2(z) = \frac{1}{\hat{\beta}}(1 - z^{-1})^2 \quad (3.25)$$

unde $\hat{\lambda}$ și $\hat{\beta}$ sunt aproximările numerice pentru câștigurile analogice λ și β .

Deoarece λ și β sunt câștiguri analogice iar $H_1(z)$ și $H_2(z)$ sunt implementate în circuite digitale, λ și β nu vor fi egale cu $\hat{\lambda}$ și $\hat{\beta}$. Se definesc termenii eroare astfel:

$$\lambda = \hat{\lambda}(1 + \delta_\lambda) \quad (3.26)$$

$$\beta = \hat{\beta}(1 + \delta_\beta)$$

Dacă se neglijează termenii de ordin superior din relația (3.23) rezultă pentru ieșirea globală expresia:

$$Y(z) = X(z)z^{-3} - \delta_\beta(1 - z^{-1})^2 z^{-1} E_{Q1}(z) + \frac{1}{\hat{\beta}}(1 - z^{-1})^3 E_{Q2}(z) \quad (3.27)$$

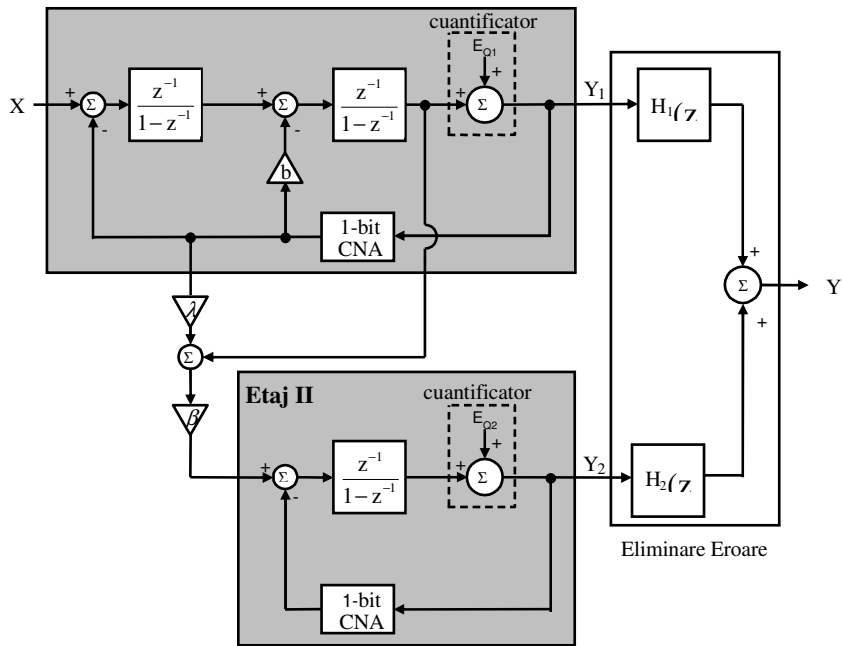


Fig. 3.8 Modulator $\Delta\Sigma$ de ordinul III – arhitectură cascadă 2-1

Din rel. 3.27 rezultă că în absența erorilor de adaptare, rămâne doar zgomotul de cuantificare al etajului doi $E_{Q2}(z)$ care suferă o formare de ordinul trei. În prezenta erorilor de adaptare o fracțiune din zgomotul de cuantificare al primului etaj $E_{Q1}(z)$ apare la ieșirea modulatorului și este formată printr-o diferențiere de ordinul doi. Prin urmare neadaptările dintre circuitele analogice și cele digitale vor degrada performanțele modulatorilor. Pentru a diminua efectul neadaptării, ordinul primului etaj din arhitectura cascadă va fi mai mare decât al etajului succesor. Acesta este motivul pentru care un modulator de ordinul trei în cascadă se implementează printr-o arhitectură 2-1 în locul unei arhitecturi 1-2 sau 1-1-1.

3.8 ERORILE DE CUANTIFICARE A FAZEI

Efectul erorilor de fază se poate găsi considerând formatorul tip întârziere – însumare descris de ecuația: $z(t, \theta, r) = \sum_{n=0}^{N-1} a_n y_n(t - \tau_n)$, unde $z(t, \theta, r)$ este ieșirea formatorului, $y_n(t)$ este intrarea furnizată de elementul n , dintr-o arie cu N elemente, iar a_n este ponderea sau apodizarea. Întârzierea pentru fiecare element, τ_n , este determinată de direcția θ pe care aria este dirijată iar în cazul sistemelor focalizate și de adâncimea r . Pentru o arie de N elemente cu distanța interelement d , considerând referința în centrul ariei, relațiile (2.26) și (2.27) care descriu întârzierile pentru dirijare și focalizare au expresiile:

$$\tau_n^s = \frac{(n - N/2)d \sin \theta}{c} \quad (3.28)$$

$$\tau_n^f = -\frac{(n - N/2)^2 d^2 \cos^2 \theta}{2rc} \quad (3.29)$$

Într-un formator numeric întârzierile descrise de (3.28) și (3.29) sunt cuantificate la un număr întreg de perioade de eșantionare $\hat{\tau}_n = k/f_s$ unde k este un întreg iar f_s frecvența de eșantionare. În urma cuantificării întârzierii rezultă o eroare de fază pentru fiecare element, e_n , în domeniul $[-\pi/OSR, \pi/OSR]$, unde $OSR = f_s / f_0$ este rata de supraeșantionare iar f_0 este frecvența semnalului. Pentru dirijarea pe o direcție θ_s și focalizarea la adâncimea r_f , ieșirea formatorului poate fi scrisă astfel:

$$z(t, \theta, r | \theta^s, r^f) = \exp(j\omega_0 t) \sum_{n=0}^{N-1} a_n \cdot \exp(je_n) \cdot \exp(j\omega_0(\tau_n^{sf} - \tau_n)) \quad (3.30)$$

Dacă se consideră că erorile de fază sunt de valori reduse termenul care conține eroarea se poate dezvolta în serie și reținând primii doi termeni se obține aproximarea:

$$\begin{aligned} z(t, \theta, r | \theta^s, r^f) &= \sum_{n=0}^{N-1} a_n \cdot \cos(e_n) \cdot \exp(j\omega_0(\tau_n^{sf} - \tau_n)) + j \cdot \sum_{n=0}^{N-1} a_n \cdot \sin(e_n) \cdot \exp(j\omega_0(\tau_n^{sf} - \tau_n)) \approx \\ &\approx \sum_{n=0}^{N-1} a_n \cdot \exp(j\omega_0(\tau_n^{sf} - \tau_n)) + j \cdot \sum_{n=0}^{N-1} a_n \cdot e_n \cdot \exp(j\omega_0(\tau_n^{sf} - \tau_n)) \end{aligned} \quad (3.31)$$

Prin urmare caracteristica fasciculului conține răspunsul dorit și un termen suplimentar care este dat de transformata Fourier a erorii de fază pe întreaga arie. Proprietățile erorii de fază sunt diferite pentru sistemele cu dirijare și focalizare față de cele care realizează doar dirijarea.

Pentru analiza ulterioară sunt importante proprietățile funcției pondere și ca atare se definesc unele caracteristici ale acesteia. Câștigul de putere coerentă normalizat (CPG) este:

$$\text{CPG} = \left[\frac{1}{N} \sum_{n=0}^{N-1} a_n \right]^2 \quad (3.32)$$

Câștigul de putere necoerentă normalizat (IPG) este:

$$\text{IPG} = \frac{1}{N} \sum_{n=0}^{N-1} a_n^2 \quad (3.33)$$

Raportul celor două este numit în analiza spectrală "lățimea de bandă normalizată echivalentă zgomotului" (ENBW). În contextul procesării spațiale un termen mai adecvat pentru acest raport ar fi "lățimea fasciculului normalizat echivalent cu zgomotul":

$$\text{ENBW} = \text{IPG} / \text{CPG} \quad (3.34)$$

Normalizarea implică valori maxime unitate pentru toate cele trei mărimi.

Dacă eroarea de cuantificare a fazei are o distribuție uniformă pe domeniul de cuantizare $[-\pi/\text{OSR}, \pi/\text{OSR}]$, dispersia erorii de fază este dată conform (3.1) de expresia: $[2(\pi/\text{OSR})]^2 / 12 = \pi^2 / (3 \cdot \text{OSR})^2$. Raportul între răspunsul dorit și cel nedorit se determină din primul și ultimul termen al relației (3.31). Semnalul dorit se obține ca rezultat al însumării coerente în formator. Componenta de eroare se presupune a fi necorelată și ca urmare procesul de însumare pentru aceasta va fi incoerent. Deși eroarea de cuantificare a fazei este într-un sens o mărime deterministă, deoarece poate fi prezisă din procesul de cuantificare, observația că erorile pentru fiecare canal sunt necorelate permite să se considere eroarea ca realizarea a unui proces aleator de tip zgomot alb. Acest tip de erori de cuantificare a fazei vor fi denumite aleatoare. Raportul între tensiunea medie de zgomot și semnalul util sau nivelul lobului secundar este:

$$\text{SL}_{\text{avg}} = \frac{\left[\sum_{n=1}^{N-1} a^2(n) \right]^{\frac{1}{2}}}{\sum_{n=1}^{N-1} a(n)} \cdot \frac{\pi}{\text{OSR} \sqrt{3}} = \frac{\pi}{\text{OSR}} \left(\frac{\text{ENBW}}{3N} \right)^{\frac{1}{2}} \quad (3.35)$$

Această expresie este cunoscută de mai multă vreme în literatura radar. Efecte similare sunt cauzate de variațiile aleatoare ale câștigului pentru fiecare canal și de cuantificarea funcției de pondere. Și în aceste cazuri este adecvat să se considere că erorile de amplitudine pentru diverse canale nu sunt corelate. În majoritatea cazurilor costurile de implementare pentru a elimina aceste erori sunt reduse față de cele necesare pentru a diminua efectele cuantificării fazei. Eroarea pătratică va fi distribuită după o distribuție de tip χ^2 cu două grade de libertate (distribuție exponențială). Ca urmare valorile maxime ale erorii pătratice sunt de 4,6 ori peste medie [Holm92]. Prin urmare nivelul maxim al lobului lateral este:

$$\text{SL}_{\text{peak}} \approx \frac{\pi}{\text{OSR}} \left(\frac{4.6 \cdot \text{ENBW}}{3N} \right)^{\frac{1}{2}} \quad (3.36)$$

3.8.1 LOBII DE CUNTIFICARE ÎNTR-UN SISTEM NEFOCALIZAT

Al doilea efect al cuantificării întârzierii îl constituie lobii de cuantificare discretă care se manifestă ca lobi de tip "gratii" sau lobi imagine. Aceștia apar ori de câte ori eroarea de cuantificare devine periodică datorită structurii ariei. Sunt trei condiții care potențază acest efect:

- o geometrie regulată a ariei (arie liniară uniformă sau arie fazată)
- emisia în undă continuă
- operarea în câmp îndepărtat

Ultima condiție nu este niciodată satisfăcută în imagistica ultrasonică și aceasta se manifestă prin reducerea lobilor laterali de cuantificare în sistemele ultrasonice medicale comparativ cu majoritatea sistemelor radar sau sonar.

Pentru o arie uniformă nefocalizată comandată în undă continuă, și o acuratețe de cuantificare a întârzierii $\tau = 1/f_s = 1/(OSR \cdot f_0)$, condiția de periodicitate a erorii de cuantificare a întârzierii este [Gray85]:

$$\left(p\tau = \frac{qd \sin \theta_{p,q}}{c} \right) \Rightarrow \sin \theta_{p,q} = \frac{p}{q} \cdot \frac{\lambda}{d \cdot OSR} \quad (3.37)$$

Prin comparație cu (3.28) se observă că relația corespunde cazului când dirijarea unei sub-arii de q elemente conduce la un timp de întârziere egal cu un număr întreg p de pași de cuantificare. Sub-aria se repetă periodic peste întreaga arie pentru dirijarea pe direcția θ_{pq} . Eroarea de cuantificare periodică duce la apariția de lobi laterali discreți, a căror direcție este dată de direcția lobilor imagine (de gratii) pentru o arie cu distanța interelement qd :

$$\sin \theta_k = k \cdot \frac{\lambda}{q \cdot d} + \sin \theta_{p,q}, \quad k \in \{ \dots -2, -1, 1, 2, \dots \} \quad (3.38)$$

Există un număr mare de combinații (p, q) care conduc la unghiuri valide, și prin urmare este practic imposibil să se elimine lobii de cuantificare evitând dirijarea ariei pe anumite direcții. Ceea ce interesează este nivelul lobilor de cuantificare pentru a evidenția posibilități de minimizare. Situația cea mai defavorabilă pentru lobii de cuantificare se obține dacă $q=2$, ceea ce conduce la $N/2$ perioade pentru eroarea de cuantificare așa cum se observă și în fig.3.9. De asemenea în această situație fiecare element are o eroare de cuantificare a fazei egală cu $\pm\pi/(2OSR)$ adică $\pm$ jumătatea pasului de cuantificare. Unghiurile corespunzătoare acestui caz sunt $\theta_{1,2}$, $\theta_{3,2}$, $\theta_{5,2}$, etc.

Raportul între amplitudinea lobului lateral și cea a lobului principal este egal cu suma erorii de cuantificare pentru N/q arii raportată la însumarea coerentă în lobul principal. Pentru $q=2$ se obține:

$$SL_{q=2} = \frac{(N/2) \left[\sin\left(\frac{\pi}{2OSR}\right) + \sin\left(\frac{\pi}{2OSR}\right) \right]}{N} \approx \frac{\pi}{2OSR} \quad (3.39)$$

Din compararea relațiilor (3.36) și (3.39) rezultă că lobii discreți de cuantificare predomină față de lobii datorati erorilor aleatoare pentru ariile de dimensiune:

$$N > \frac{4}{3} \cdot 4,6 \cdot ENBV \quad (3.40)$$

Pentru funcțiile de ponderare folosite în practică, performanța ariilor liniare cu mai mult de 8 elemente este limitată, în câmp îndepărtat, mai mult de lobii discreți de cuantificare decât de lobii laterali aleatori.

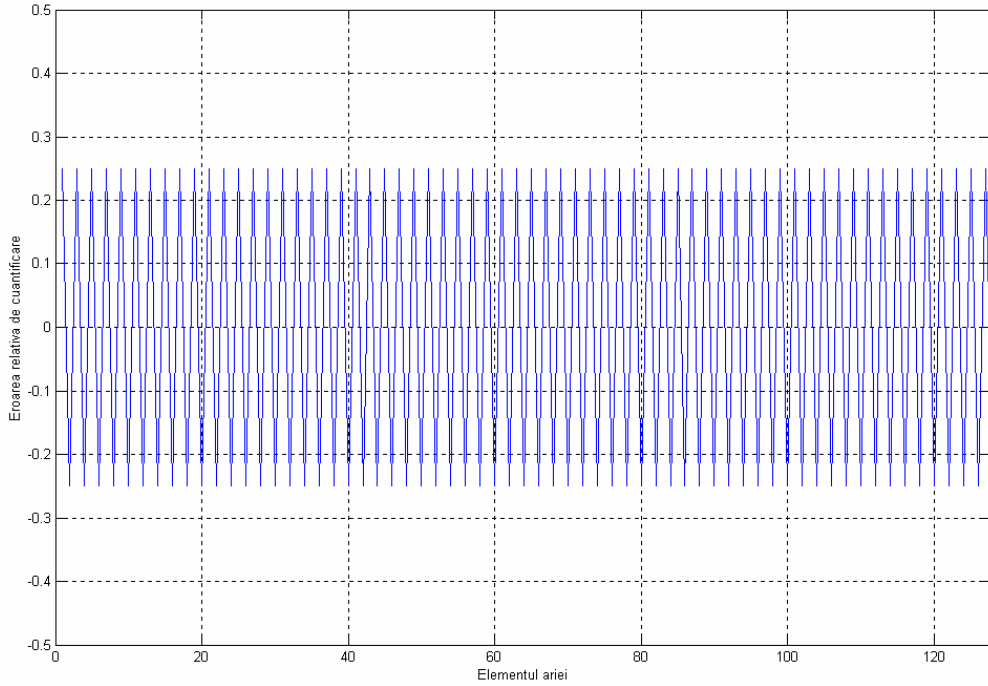


Fig. 3.9 Eroarea periodică de cuantificare pentru o arie uniformă nefocalizată

3.8.2 LOBII LATERALI ÎNTR-UN SISTEM FOCALIZAT

Când formatorul nu este focalizat, întârzierea crește liniar cu indicele elementului curent și în acest caz eroarea de fază poate deveni periodică. Aceasta duce la apariția unor componente discrete așa cum rezultă din transformata Fourier a erorilor de fază, relația (3.31). În cazul focalizării, la întârzierea de dirijare se adună termenul pătratic al întârzierii pentru focalizare. Când focalizarea se aplică pe distanțe reduse, în câmp apropiat, curbura funcției de focalizare este suficient de largă încât se poate considera că eroarea periodică de fază devine complet aleatoare. Eroarea de fază poate fi considerată ca fiind necorelată și se poate determina nivelul lobilor laterali și probabilitatea de distribuție a fasciculului.

Un sistem focalizat, fără dirijare, este caracterizat de o eroare de fază reprezentată în fig.(3.10): Acest tip de eroare produce lobi laterali în vecinătatea lobului principal dorit, care au ca efect lărgirea lobului principal.

Cazul cel mai nefavorabil este combinația între focalizare și dirijarea pe una din direcțiile care produc o eroare de fază periodică în câmp îndepărtat. Mai mult cazul cel mai defavorabil pentru eroarea periodică de fază este când eroarea de fază alternează pe cel mai scurt interval posibil deoarece atunci este cel mai puțin afectată de focalizare. Aceasta se observă ușor din direcțiile $\theta_{p,2}$ (3.38), unde eroarea de cuantificare variază între $e_0 \pm \pi/(2OSR)$, iar e_0 este o eroare de cuantificare constantă, dată de defazarea dintre faza de cuantificat și cuantificare. În acest caz, amplitudinea erorii de fază în câmp îndepărtat determină de asemenea forma erorii de fază datorată termenului de focalizare. Ori de câte ori suma dintre termenul negativ de focalizare, $e_{n,f}$, și eroarea de fază în câmp îndepărtat, $e_{n,s}$, este sub $-\pi/OSR$ va exista o tranziție în eroare de fază totală. În acest caz eroarea de fază, fig.3.11, se poate modela astfel:

$$e_n = (e_{n,s} + e_{n,f}) \cdot \text{sign}_f \quad (3.41)$$

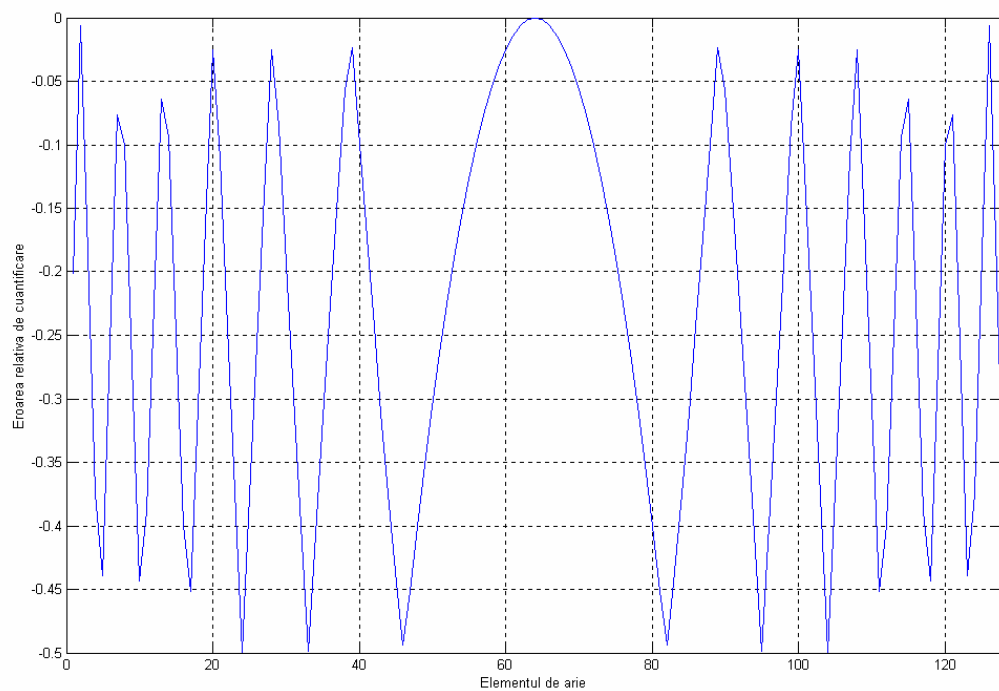


Fig. 3.10 Eroarea periodică de cuantificare datorată focalizării

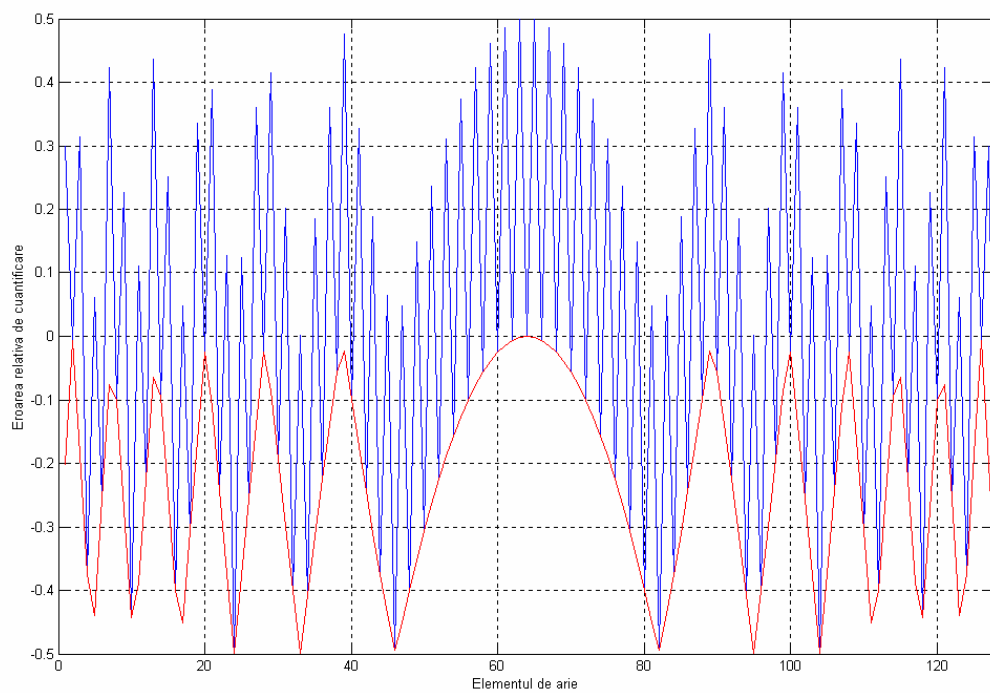


Fig. 3.11 Eroarea de cuantificare pentru un sistem cu dirijare și focalizare

Termenul de focalizare $e_{n,f}$ este indicat în fig.3.10. Contribuția sa la eroarea totală de cuantificare apare în caracteristica fasciculului ca o perturbare a lobului principal dar acest aspect nu este relevant în ceea ce privește nivelul lobilor secundari. Funcția $sign_f$ derivă din $e_{n,f}$ atribuind valoarea +1 secțiunii centrale de dimensiune N_c , valoarea -1 secțiunii următoare s.a.m.d. Trecherile prin zero ale funcției sunt simplu de determinat pe baza proprietăților parabolei și se exprimă funcție de N_c astfel: $\sqrt{2} \cdot N_c / 2$ distanța de la centru la a doua tranziție, $\sqrt{3} \cdot N_c / 2$ distanța de la centru la a treia tranziție etc. Multiplicarea din ecuația precedentă se translatează într-o convoluție în domeniul spațiu. Trebuie găsit termenul principal din transformarea Fourier a lui $sign_f$. Perioada acestui termen este dublul dimensiunii secțiunii centrale, $2N_c$. Modificarea de direcție a lobului imagine datorită acestui termen este echivalentă cu lobul imagine al unei arii de dimensiune $2N_c$ adică termenul $\sin(\theta_k)$ din (3.38) va modificat cu $\pm \lambda / (2N_c d)$. Aceasta este contribuția majoră la extinderea lobului principal dar nu este singura. Cazul cel mai defavorabil privind reducerea de nivel la focalizare este dat de transformata Fourier a lui $sign_f$ la frecvența $1/2N_c$. Din considerente de simetrie transformata Fourier va fi reală și diminuarea de nivel se poate determina din corelația între funcția $sign_f$ și cosinus. Se poate deduce o valoare aproximativă luând în considerare doar semiperioada funcției cosinus care corespunde secțiunii centrale și observând că pentru celelalte secțiuni variația rapidă a lui $sign_f$ conduce prin mediere la un termen nul. Aproximarea este cu atât mai bună cu cât funcția de pondere folosită atenuază mai mult spre frontiera aperturii. Corelația aproximativă este:

$$\frac{\sum_{N_c} a_n \cdot sign_f(n) \cdot \cos\left(2\pi \cdot \frac{n}{2N_c}\right)}{\sum_N (a_n \cdot sign_f(n))^2} = \frac{\sum_{N_c} a_n \cdot \cos\left(2\pi \cdot \frac{n}{2N_c}\right)}{\sum_N a_n^2} \quad (3.42)$$

Reducerea aproximativă de nivel relativ la estimarea (3.39) când N_c este mic în comparație cu N este:

$$\frac{\sum_{N_c} \cos\left(2\pi \cdot \frac{n}{2N_c}\right)}{N \cdot IPG} \approx \frac{N_c}{N} \cdot \frac{\frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \cos x dx}{IPG} = \frac{2}{\pi} \cdot \frac{N_c}{N} \cdot \frac{1}{IGP} \quad (3.43)$$

Ținând seama de (3.39) rezultă nivelul lobului lateral discret ca fiind:

$$SL_f \approx \frac{\pi}{2 \cdot OSR} \cdot \frac{2}{\pi} \cdot \frac{N_c}{N} \cdot \frac{1}{IGP} = \frac{N_c}{N \cdot OSR} \cdot \frac{1}{IGP} \quad (3.44)$$

Pentru a evalua nivelul lobilor trebuie determinat N_c . Se poate arăta că valoarea maximă se obține pentru $e_0 = \pi / OSR$ și corespunde situației când (3.29) este egală cu o jumătate de cuantă.

$$\left(\frac{(n - N/2)^2 d^2 \cos^2 \theta}{2rc} = \frac{\pi}{OSR} \cdot \frac{1}{2\pi f_0} \right) \Rightarrow n - N/2 = \pm \frac{N}{L \cos \theta} \left(\frac{r\lambda}{OSR} \right)^{1/2} \quad (3.45)$$

unde $L = Nd$ este apertura ariei.

Dimensiunea secțiunii centrale a aperturii este dublul acestei valori

$$N_c = \frac{2N}{L \cos \theta} \left(\frac{r\lambda}{OSR} \right)^{1/2} \quad (3.46)$$

Înlocuind această valoare în (3.44) se obține

$$SL_f \approx \frac{2}{L \cdot \cos \theta \cdot OSR} \cdot \frac{1}{IPG} \left(\frac{r\lambda}{OSR} \right)^{1/2} \quad (3.47)$$

Această ecuație indică nivelul lobului lateral, pentru un sistem focalizat, în cel mai defavorabil caz, când aria este dirijată pe o direcție dată de (3.38)

Examinarea relației (3.47) arată o reducere a nivelului lobiilor laterali datorată focalizării. Din egalarea relațiilor (3.36) și (3.47) se obține adâncimea de la care lobiile aleatorii devin preponderenți față de lobiile discrete

$$r_r = \frac{4,6 \cdot \pi^2 \cdot ENBW \cdot IPG^2}{12} \cdot \frac{(L \cos \theta)^2}{\lambda} \cdot \frac{OSR}{N} = k_p \cdot \frac{(L \cos \theta)^2}{\lambda} \cdot \frac{OSR}{N} \quad (3.48)$$

Constanta k_p depinde de funcția de pondere fiind 0,82 pentru pondere Hamming și 3,8 pentru ponderare uniformă. Pentru distanțe dincolo limita indicată de (3.48) lobiile discrete sunt preponderenți. Dacă r_r se situează dincolo de limita de tranziție spre câmpul îndepărtat r_t dată de

$$r_t = \frac{(L \cos \theta)^2}{4\lambda} \quad (3.49)$$

lobii laterali nu prezintă importanță pentru sistemul focalizat. Prin urmare lobiile laterali prezintă interes în sistemele în care $r_r < r_t$ adică:

$$k_p \cdot \frac{(L \cos \theta)^2}{\lambda} \cdot \frac{OSR}{N} < \frac{(L \cos \theta)^2}{4\lambda} \Rightarrow N > 4 \cdot k_p \cdot OSR \quad (3.50)$$

Prin urmare lobiile laterali prezintă interes pentru sistemele cu un număr mare de elemente care lucrează la frecvențe ridicate (OSR redus). Folosirea unor frecvențe de eșantionare ridicate conduce la diminuarea influenței lobiilor laterali. Relația (3.50) indică o valoare mai mare de 16 pentru OSR în cazul unei arii de 128 elemente cu ponderare uniformă.

3.9 CERINȚE DE PROIECTARE

Unul dintre avantajele cheie ale modulatorilor delta-sigma în comparație cu CAN multi-bit constă în simplitatea implementării CAN $\Delta\Sigma$. Un CAN multi-bit rapid folosește 256÷1024 comparatoare conectate la un codificator pentru a obține codul numeric adecvat unei anumite tensiuni de intrare. Pe de altă parte, un CAN $\Delta\Sigma$ folosește câteva amplificatoare operaționale (pentru nodurile diferențiator și sumatoare) și un singur comparator la ieșire pentru a genera secvența modulată de date. Dacă ieșirea pe un singur bit este asigurată, CNA-ul de pe calea de reacție a modulatorului $\Delta\Sigma$ este doar un amplificator operațional care se saturează la un nivel sau altul al tensiunii de alimentare. Datorită acestei simplificări CAN $\Delta\Sigma$ va ocupa mai puțin spațiu într-un formator de fascicul ultrasonic integrat și chiar dacă frecvențele de operare sunt mari consumul de putere al CAN $\Delta\Sigma$ va fi mai mic în comparație cu al unui CAN rapid (flash).

Deși implementarea CAN $\Delta\Sigma$ este relativ simplă la nivel de componente, aceste componente sunt totuși destul de complicate. Circuitele active de diferențiere și sumare folosite în modulatorii $\Delta\Sigma$ necesită o proiectare și o cablare atentă pentru asigurarea stabilității și a

performanțelor optime. Ratele de eșantionare mari ale CAN $\Delta\Sigma$ complică și ele proiectarea modulatorului datorită efectelor de diafonie care nu sunt neglijabile și datorită semnalelor parazitate. Prin comparație, CAN flash se bazează pe un divizor de tensiune repetitiv și circuite comparatoare, care trebuie proiectate precis și apoi reproduse de 1024 ori în circuit pentru a codifica cu acuratețe tensiunea de intrare.

O altă cerință importantă de implementare care diferențiază CAN $\Delta\Sigma$ de CAN flash este necesitatea preciziei componentelor analogice. S-au discutat mai înainte efectele integratoarelor neideale și câștigului în funcționarea $\Delta\Sigma$; totuși, precizia acestor componente active nu ridică probleme dacă se încadrează în limita de câteva procente în jurul valorii specificate. CAN flash se bazează pe componente analogice foarte precise pentru a asigura precizia de $1/2^{10} = 1/1024$ reprezentată de intervalul de cuantificare. Aceasta înseamnă că rezistoarele, capacitățile și amplificatoarele operaționale din CAN flash trebuie să se încadreze într-o limită de toleranță de 0,1%. În cazul în care componentele CAN flash nu au această precizie, vor rezulta coduri eronate sau ieșiri ne-monotone (ne-liniare). Desigur această diferență de ordin de mărime, pentru precizie simplifică proiectarea și constrângerile de fabricare a CAN $\Delta\Sigma$ în comparație cu tehnica flash.

O ultimă zonă de interes unde CAN $\Delta\Sigma$ sunt superioare CAN flash este proiectarea filtrului anti-aliere care precede digitizorul. CAN-urile multi-bit obișnuite funcționând la o rată puțin peste rata Nyquist necesită un filtru a cărui pantă variază puternic cu frecvența (steep roll-off) conectat înaintea CAN pentru a preveni suprapunerea peste semnal a zgomotului de intrare de frecvență înaltă. Ratele mari de supraeșantionare simplifică proiectarea filtrului anti-aliere deoarece este necesar un filtru de ordin mai mic. Ca rezultat CAN-urile $\Delta\Sigma$ cu supraeșantionare pot folosi un filtru anti-aliere foarte simplu (chiar un AO de bandă limitată, de exemplu preamplificatorul) la intrare reducându-se astfel complexitatea circuitului, mărimea și consumul de putere.

3.10 ALEGEREA ARHITECTURII

Alegerea arhitecturii pentru implementarea modulatorilor delta-sigma de mare viteză necesare în structura formatoarelor de fascicule ultrasonice are la bază următoarele cerințe:

- un factor de supraeșantionare ridicat pentru a asigura formarea cu acuratețe a fasciculului și diminuarea influenței lobilor laterali dați de eroarea de cuantificare
- asigurarea domeniului dinamic de 80-100dB pentru procesarea unei game largi de semnale inclusiv a reflexiilor provenite din mediul sangvin aflat în mișcare [Routh96]
- păstrarea complexității în limite rezonabile pentru a conserva șansa de integrare într-un singur circuit a convertoarelor pentru o arie de 64, 128 sau 256 elemente.

Figura 3.12 redă dependența DR de OSR, descrisă de relația (3.19), pentru modulatorii de ordinul 2, 3, 4 și 5. Se observă că la un OSR de 32 se obțin domenii dinamice de 65 dB pentru ordinul 3, 85 dB pentru ordinul 4 și 110 dB pentru ordinul 5, iar la un OSR de 64 valorile DR sunt 80dB, 105dB și 135dB. Rezultă de aici că pentru aplicațiile ultrasonice sunt potrivite modulatorii de ordinul 3 care lucrează la un OSR mai mare de 64 sau de ordin superior care lucrează la un OSR mai mare de 32. Pentru aplicațiile medicale limita benzii traductoarelor ajunge la 10 MHz ceea ce înseamnă frecvențe de lucru de 640 MHz pentru modulatorii de ordinul 4 sau mai mare și de 1280 MHz pentru modulatorii de ordinul 3. Pentru o aplicație concretă cu un traductor de 3,5 MHz cu o bandă de 50% se poate alege un modulator de ordinul 3 care lucrează la o frecvență de 320 MHz cu un OSR de 32 și asigura domeniul dinamic de 65dB. Restul de 20 dB ceruți de domeniul dinamic se asigură prin mecanismul de compensare automată a câștigului la nivelul amplificatorului de intrare.

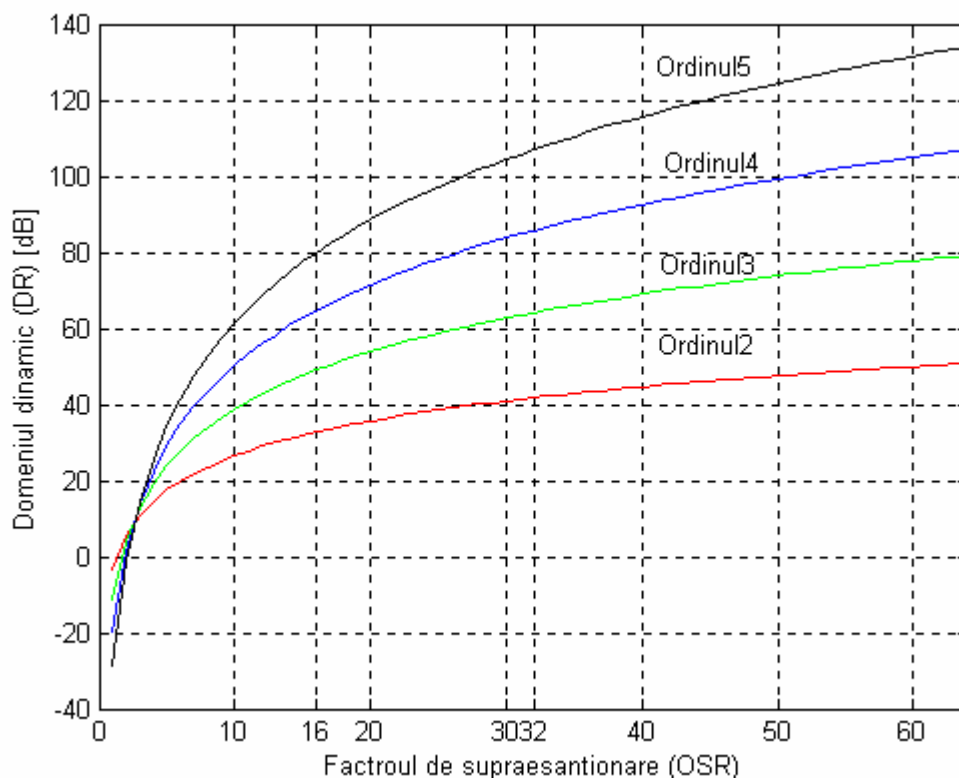


Fig. 3.12 Dependența domeniului dinamic de factorul de supraesantionare

Cu aceste observații se poate adopta un modulator de ordinul 3 care funcționează la o frecvență de 320 MHz și poate fi implementat în tehnologie BiCMOS. Arhitectura modulatorului se alege între varianta cu o singură buclă și cascada 2-1. Prima variantă este relativ mai simplă pe când a doua are avantajul stabilității rezultat din stabilitatea modulatorilor de ordinul I și II care îl compun.

3.11 CONCLUZII

Pornind de la primul dintre obiectivele enunțate anterior, identificarea unei structuri de convertor $\Delta\Sigma$ convenabilă din punct de vedere al raportului complexitate / performanță, în acest capitol se prezintă o analiză unitară a problematicii legate de cuantificarea semnalelor recepționate de o arie de transductoare ultrasonice. Analiza a fost orientată pe două direcții principale:

- **identificarea și evaluarea efectului erorilor de cuantificare a fazei asupra performanțelor formatorului digital de fascicul**
- **determinarea dependenței complexitate performanță pentru convertoarele $\Delta\Sigma$**

Prima direcție de analiză evidențiază următoarele efecte date de eroarea de cuantificare a fazei:

Deși eroarea de cuantificare a fazei este într-un sens o mărime deterministă, deoarece poate fi prezisă din procesul de cuantificare, observația că erorile pentru fiecare canal sunt necorelate permite să se considere eroarea ca realizarea a unui proces aleator de tip zgomot alb. Acest tip de erori de cuantificare a fazei sunt denumite aleatoare. Componenta de eroare se presupune a fi necorelată și ca urmare procesul de însumare pentru aceasta va fi incoerent. Structura lobului lateral rezultat din procesul de însumare este de asemenea aleatoare funcție de unghi și poate fi caracterizată printr-un nivel mediu al lobului lateral.

- Apariția unei erori periodice în cazul ariilor uniforme nefocalizate. Aceasta este situația formatoarelor de fascicul care operează în câmp îndepărtat și corespunde tehnicilor RADAR și SONAR.

- Erorile de cuantificare a termenului pătratic care descrie întârzierea pentru focalizare conduc la apariția de lobi laterali discreți în vecinătatea lobului principal

- Dirijarea în combinație cu focalizarea conduc la apariția de lobi discreți de cuantificare a căror amplitudine este de obicei mai mare decât a celor cauzăți numai de focalizare. Acesta este cazul cel mai defavorabil pentru lobi laterali la sistemele de investigare ultrasonică medicală care folosesc atât dirijarea cât și focalizarea

Tabelul de mai jos însumează efectele menționate și indică ecuațiile ce descriu nivelurile lobilor laterali.

Tabel 3.1

Lobii de cuantificare a fazei aleatoare	$SL_{avg} = \frac{\pi}{OSR} \left(\frac{ENBW}{3N} \right)^{\frac{1}{2}}$
Lobii discreți de cuantificare – numai dirijare	$SL_{q=2} \approx \frac{\pi}{2OSR}$
Lobii de cuantificare la focalizare – fără dirijare	$SL_f \approx \frac{2}{L \cdot OSR} \cdot \frac{1}{IPG} \left(\frac{r\lambda}{OSR} \right)^{1/2}$
Lobii discreți de cuantificare – sistem focalizat	$SL_f \approx \frac{2}{L \cdot \cos\theta \cdot OSR} \cdot \frac{1}{IPG} \left(\frac{r\lambda}{OSR} \right)^{1/2}$

Concluzia principală rezultată este că la investigarea ultrasonică medicală care folosește atât dirijarea cât și focalizarea, lobi laterali prezintă interes pentru sistemele cu un număr mare de elemente care funcționează la frecvențe ridicate (OSR redus). Folosirea unor frecvențe de eșantionare ridicate conduce la diminuarea influenței lobilor laterali. Relația (3.50) indică o valoare mai mare de 16 pentru OSR în cazul unei arii de 128 elemente cu ponderare uniformă. Acest rezultat constituie un criteriu de alegere a factorului de supraeșantionare și a structurii convertorului.

A doua direcție de analiză evidențiază performanțele pentru diverse structuri de convertoare parcurse în sensul creșterii complexității. Analiza pornește de la prezentarea locului și rolului convertorului în evoluția sistemelor de investigare și după evaluarea convertoarelor flash multibit prezintă modalitățile de obținere a unor rezoluții înalte prin folosirea tehnicilor de supraeșantionare și de formare a zgomotului.

S-a evidențiat dependența parametrilor convertorului, raportul semnal - zgomot (SNR) și domeniul dinamic (DR), de ordinul și implicit de complexitatea modulatorului. Pe baza acestei dependențe se poate alege ordinul modulatorului care asigură la un OSR dat parametrii doriți.

Trecerea în revistă a arhitecturilor convertoarelor arată corelația între complexitate și stabilitate pentru modulatorile de ordin mai mare decât doi și cerințele restrictive de proiectare.

Această direcție de analiză conduce la următoarele concluzii:

- performanțele convertoarelor $\Delta\Sigma$ sunt direct corelate cu ordinul modulatorului
- asigurarea stabilității convertoarelor de ordin superior impune cerințe restrictive de proiectare care se regăsesc în final în creșterea complexității convertorului.

Concluziile desprinse pe cele două direcții de analiză permit alegerea arhitecturii potrivite pentru convertorul delta-sigma de mare viteză necesar în structura formatorului de fascicul ultrasonic pentru satisfacerea următoarelor cerințe:

-
- un factor de supraeșantionare mai mare sau egal cu 32 , pentru a asigura formarea cu acuratețe a fasciculului și diminuarea influenței lobilor laterali dați de eroarea de cuantificare pentru o arie cu 128 elemente.
 - asigurarea unui domeniu dinamic de 80-100 dB pentru procesarea unei game largi de semnale inclusiv a reflexiilor provenite din mediul sangvin aflat în mișcare
 - păstrarea complexității în limite rezonabile pentru a conserva șansa de integrare într-un singur circuit a convertoarelor pentru o arie de 64, 128 sau 256 elemente.

Într-o variantă optimă pentru aplicațiile cu traductoare având limita superioară a benzii de 5 MHz, un modulator de ordinul 3 cu un OSR de 32 funcționează la o frecvență de 320 MHz și asigură domeniul dinamic de 65dB. Restul de 20 dB ceruți de aplicație se asigură prin mecanismul de compensare automată a câștigului la nivelul amplificatorului de intrare. Modulatorul de ordinul 3 care funcționează la o frecvență de 320 MHz și poate fi implementat în tehnologie BiCMOS. Arhitectura modulatorului se alege între varianta cu o singură buclă și cascada 2-1. Prima variantă este relativ mai simplă pe când a doua are avantajul stabilității rezultat din stabilitatea modulatorilor de ordinul I și II care îl compun.

CAPITOLUL IV

FORMATOR DE FASCICUL ULTRASONIC CU MODULAȚIE DELTA-SIGMA

Capitolul tratează problematica formatoarelor de fascicul de tip întârziere-însumare care au la bază modulația delta-sigma. Sunt supuse analizei comparative variantele de realizare cu eșantionare uniformă și respectiv neuniformă. Având ca referință formatorul convențional multibit se urmărește alegerea unei soluții constructive care să asigure performanțe comparative cu cele ale formatorului clasic dar cu avantajul major al simplificării structurii hardware. Analiza se bazează pe modelarea Matlab cu date de intrare reale a celor trei variante constructive: clasic, modulație delta-sigma uniformă și modulație delta-sigma neuniformă, și cuantificarea performanțelor fiecărui tip de formator.

4.1 LOCUL FORMATORULUI DE FASCICUL ÎN STRUCTURA SISTEMULUI IMAGISTIC

Structura simplificată a unui sistem imagistic ultrasonic în mod B include următoarele blocuri componente - figura 4.1:

- **Aria de traductoare ultrasonice** cu N elemente folosită pentru a converti o mulțime de semnale electrice în semnale ultrasonice care pot fi aplicate mediului de investigat și pentru a converti semnalele ultrasonice recepționate în semnale electrice. Elementele ariei sunt dispuse într-o configurație liniară, numărul lor N putând fi 64, 96, 128, 256 funcție de aplicație.
- **Generatorul de impulsuri** care produce semnalele electrice, cu forma și durata dorită, folosite pentru activarea elementelor ariei
- **Formatorul de fascicul pentru emisie** – care generează secvența de întârziere pentru semnalele emise astfel ca undele acustice generate de elementele ariei să se însumeze în punctul focal dorit.
- **Comutatorul Emisie/Recepție** – cu rol de a conecta elementele ariei fie la generatorul de semnal electric fie la blocul de recepție conform secvenței de funcționare.
- **Circuitele pentru preamplificarea semnalelor recepționate și controlul temporal al câștigului** (TGC – Time Gain Control)
- **Formatorul de fascicul la recepție** – care generează fasciculul recepționat prin întârzierea și însumarea semnalelor recepționate de la elementele ariei pentru fiecare punct focal de pe o direcție dată.
- **Procesorul de semnal** pentru prelucrarea informației livrate de formatorul receptor
- **Controlerul digital** care comandă și coordonează operațiile tuturor blocurilor electronice enumerate mai sus

Funcționarea sistemului este inițializată de controlerul digital care comandă producerea unui set de N impulsuri de către generatorul de impulsuri. Acest set de impulsuri este întârziat de către formatorul emițător pentru a dirija și focaliza aria de traductoare pe direcția și la distanța dorită. Semnalele electrice întârziate de la ieșirea formatorului sunt aplicate elementelor ariei prin intermediul comutatorului Emisie/Recepție, care este plasat de către controller în modul emisie, și sunt convertite de acestea în unde ultrasonice. Pe durata propagării undele ultrasonice interacționează cu mediul prin reflexie, împrăștiere sau absorbție, ceea ce conduce la apariția de semnale ecou ce se propagă înapoi spre aria de

traductoare în timp ce o fracțiune din undele emise își continuă propagarea prin mediu. Pe de altă parte, în urma transmiterii undelor ultrasonice în mediu controlerul trece comutatorul Emisie/Recepție în mod recepție pentru a direcționa semnalele electrice recepționate spre blocul de preamplificare și control a amplificării. Cele N semnale electrice recepționate care reprezintă N semnale ecou sunt preamplificate și apoi sunt livrate spre amplificatoarele cu câștig variabil în timp pentru a compensa atenuarea introdusă de mediu. [Lie00b] Semnalele amplificate sunt aplicate formatorului receptor pentru a fi întârziate corespunzător și însumate generând astfel un fascicul sumă dirijat și focalizat. Deoarece un sector de scanare de 90° este compus dintr-o mulțime de fascicule dirijate pe direcții succesive, procedura expusă mai sus se repetă pentru diferite unghiuri de dirijare până când toate direcțiile sunt scanate.

Procesorul de semnal realizează operațiile de detecție de anvelopă, conversie a formatului de scanare și compresie logaritmică asupra fasciculului sumă pentru a produce imaginea mod B care va fi afișată pe unitatea de display.

- Detecția de anvelopă este necesară deoarece imaginea mod B este anvelopa fasciculului de radiofrecvență dirijat și focalizat.
- Deoarece fasciculele ultrasonice sunt generate în coordonate polare, neconvenabile pentru afișare, este necesară conversia formatului de scanare pentru a transforma datele din format polar (r, θ) în format rectangular (x, y) . Aceasta se realizează prin interpolarea fiecărui punct (x, y) de pe grid din punctele învecinate din matricea (r, θ) .
- Reprezentarea rectangulară (x, y) a imaginii este compresată logaritmic pentru domeniul dinamic dorit funcție de aplicație.

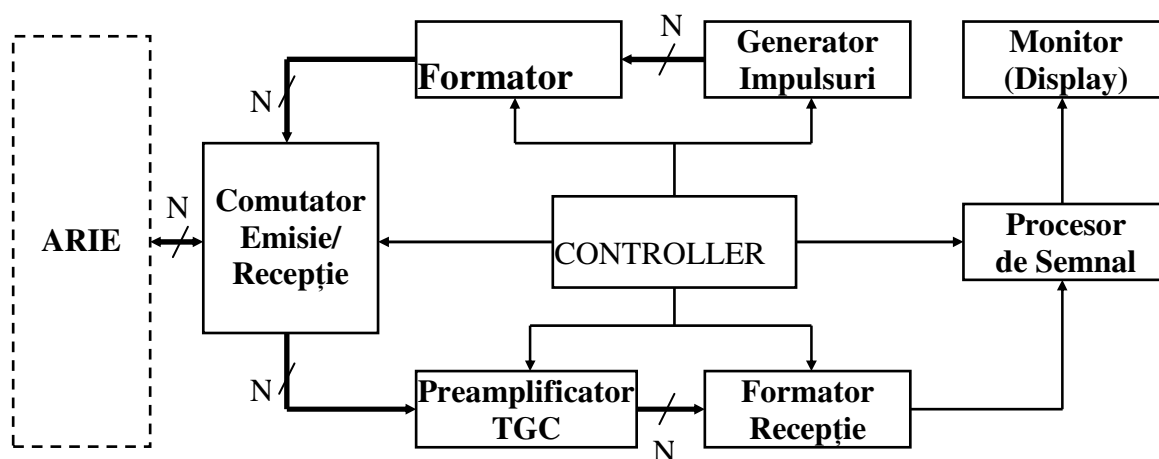


Fig. 4.1 Structura sistemului imagistic ultrasonic mod B

4.1.1 MECANISMUL DE SCANARE

Cea mai potrivită metoda de producere a imaginilor ultrasonice în mod B este prin folosirea ariei comandate prin fază la care toate elementele ariei de traductoare sunt active atât la emisia cât și la recepția fasciculului ultrasonic. Deoarece oferă un raport semnal zgomot de valoare mare și rezoluție spațială și de contrast foarte bună tehnicile de investigare cu arii fazate sunt deosebit de utile în imagistica medicală.

Sistemele de investigare în mod B cu arii fazate scanează în mod obișnuit un sector de 90 grade prin operații de sinteză de fascicul la emisie și respectiv la recepție, pentru a obține o secțiune transversală bidimensională în mediul de investigat. În tehnica de scanare sectorială formare fasciculelor este controlată electronic folosind linii de întârziere cu lungime variabilă. Concret, undele ultrasonice transmise sau recepționate pot fi dirijate și/sau focalizate pe direcția dată și în punctul dorit ajustând întârzierile elementelor ariei pentru a compensa diferențele de parcurs dus-întors. În procesul de formare la emisie se transmit în mediul

investigat pulsuri întârziate corespunzător iar în procesul de formare la recepție semnalele sunt întârziate și însumate pentru a se obține fasciculul sumă. Un sector este scanat prin operații de formare la transmisie și recepție pentru toate direcțiile unghiulare ce compun sectorul investigat – figura 4.2 Din această perspectivă este important de stabilit numărul direcțiilor de investigare și numărul punctelor focale pe fiecare direcție.

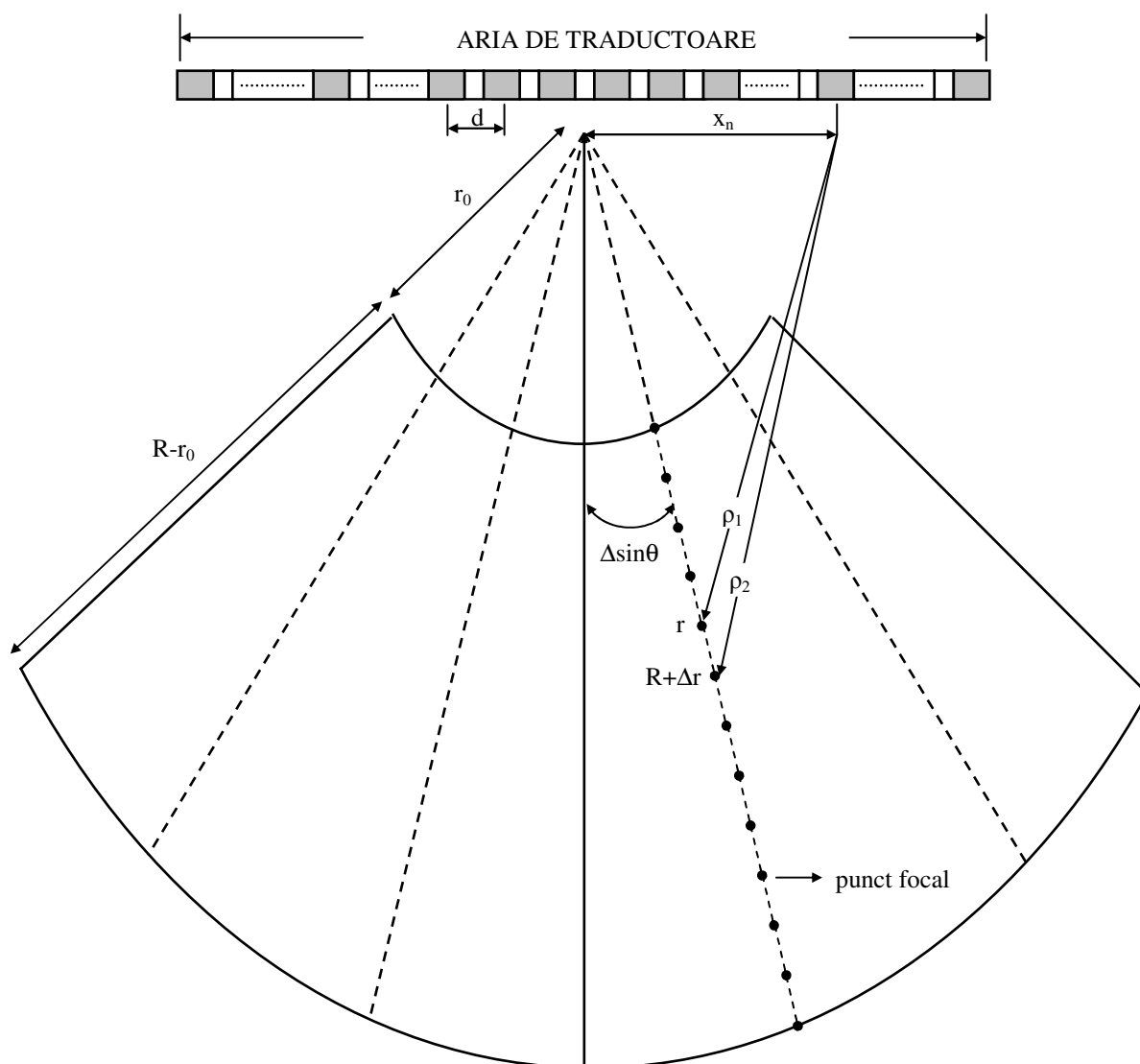


Fig. 4.2 Explicativă privind mecanismul de scanare sectorială

În imagistica ultrasonică sectorială în mod B numai pulsurile ultrasonice care sunt dirijate și focalizate într-un punct dat pe direcția dorită se pot propaga în regiunea de interes la un moment dat. Prin urmare odată ce a fost emis un set de pulsuri, următorul set poate fi transmis doar după un timp egal cu parcursul dus-întors al setului emis până la adâncimea maximă. Rezultă deci că pentru un sistem imagistic în timp real rata cadrelor, numărul de fascicule (linii) pe cadru și adâncimea de investigare nu pot fi alese arbitrar datorită vitezei finite de propagare a ultrasunetelor în mediu investigat. Pentru un sistem imagistic în timp real constrângerea se exprimă prin relația:

$$F \cdot B \cdot 2R \leq c \quad (4.1)$$

unde F , B , R și c reprezintă numărul de cadre pe secundă, numărul de fascicule pe cadru, adâncimea de investigare și viteza sunetului. Pentru $R=200\text{mm}$, $c=1540\text{ m/sec}$ și $F=20\text{ cadre/sec}$ numărul de fascicule pe cadru B trebuie să fie mai mic sau cel mult egal cu 192.

Pe de altă parte intervalul de eşantionare spaţială ($\Delta \sin \theta$) în planul fasciculului, care se defineşte prin diferenţa funcţiilor sinus pentru două direcţii de dirijare alăturate este determinat în concordanţă cu criteriul Nyquist. Pentru sistemele cu arii focalizate există a relaţie dată prin transformata Fourier între funcţia de apertură şi răspunsul spaţial al ariei în zona focală sau în câmp îndepărtat. Funcţia de apertură a unei arii cu N elemente separate între ele cu o distanţă d poate fi exprimată prin:

$$a(x) = \begin{cases} 1, & (-\frac{N}{2} + 0,5)d \leq x \leq (\frac{N}{2} - 0,5)d \\ 0, & \text{în rest} \end{cases} \quad (4.2)$$

Răspunsul unghiular al ariei pentru un parcurs dus-întors (emisie - recepţie) este dat prin următoarea relaţie bazată pe transformata Fourier:

$$U(\theta) = F\{a_t(x) * a_r(x)\} \quad (4.3)$$

unde a_t şi a_r sunt funcţiile de apertură la emisie şi recepţie, θ este unghiul direcţiei de dirijare cu normala la suprafaţa ariei, iar F şi $*$ indică operaţiile de transformare Fourier şi respectiv convoluţie. Deoarece funcţiile de apertură la emisie şi recepţie sunt identice pentru o arie fazată convenţională extinderea aperturii efective (dublul benzii de frecvenţă spaţială), rezultată din convoluţia ariilor emiţătoare şi receptoare cu N elemente, este $(2N-1)d/\lambda$. Frecvenţa spaţială se consideră uzual în unităţi normalizate corespunzătoare cu lungimea de undă λ . Frecvenţa de eşantionare spaţială minimă $1/(\Delta \sin \theta)$ trebuie să fie dublul benzii de frecvenţă spaţială $(2N-1)d/\lambda$ [Goodman92]. În consecinţă criteriul Nyquist pentru eşantionarea spaţială se exprimă prin:

$$\Delta \sin \theta \leq \frac{\lambda}{(2N-1)d} \quad (4.4)$$

Pentru un sector de 90 grade, adică $-1/\sqrt{2} \leq \sin \theta \leq 1/\sqrt{2}$, relaţia între numărul de fascicule (linii) şi perioada de eşantionare spaţială poate fi exprimată prin:

$$B \cdot \Delta \sin \theta = \sqrt{2} \quad (4.5)$$

Combinând ecuaţiile 4.4 şi 4.5 obţinem valoarea limită inferioară a numărului de fascicule:

$$B \geq \frac{\sqrt{2}d(2N-1)}{\lambda} \quad (4.6)$$

Relaţia arată că pentru $d=\lambda/2$ numărul de linii (fascicule) per cadru trebuie să fie mai mare sau egal cu de $\sqrt{2}$ ori numărul elementelor ariei N , în ipoteza că N este mult mai mare decât 1. Pentru o arie cu 128 elemente numărul liniilor trebuie să fie mai mare sau egal cu 182. Din combinarea relaţiilor 4.1 şi 4.6 se obţin valorile limita pentru numărul de fascicule per cadru

$$\frac{\sqrt{2}d(2N-1)}{\lambda} \leq B \leq \frac{c}{2 \cdot F \cdot R} \quad (4.7)$$

În concluzie pentru un sistem de investigare în timp real care foloseşte o arie cu 128 elemente cu distanţa interelement $d=\lambda/2$, la o frecvenţă a cadrelor de 20 Hz şi o adâncime de investigare de 200mm se poate asigura pentru un sector 90 grade un număr de 192 linii de scanare cu un increment spaţial $\Delta \sin \theta = 0,0078$.

Numărul punctelor focale de pe o direcţie defineşte frecvenţa de formare a fasciculului f_{BF} şi impune o valoare limită inferioară pentru frecvenţa de eşantionare.

4.1.2 ALEGEREA TIPULUI DE FOCALIZARE

Funcția principală a focalizării este aceea de a îmbunătăți rezoluția laterală. Așa cum s-a arătat în capitolul 2 întârzierea datorată focalizării τ^f este dependentă de distanța r între punctul focal curent și originea sistemului de coordonate. Funcție de modalitatea de fixare a punctului focal focalizarea poate fi fixă, dinamică sau compusă.

Pentru *focalizarea fixă* punctele focale sunt localizate la o anumită distanță R . Un punct de focalizare tipic este situat în mijlocul regiunii de vizualizare. Fasciculul focalizat are diametru minim la distanța R , iar dincolo de această distanță, diametrul lateral al fasciculului crește.

În *focalizarea dinamică*, întârzierea de focalizare este ajustată în funcție de raza r . Focalizarea dinamică la recepție este de obicei folosită pentru ca adâncimea câmpului să fie extinsă fără reducerea frecvenței cadrelor. Recepția focalizată dinamic mărește electronic distanța focală a fasciculului recepționat funcție de timp. Focalizarea dinamică poate de asemenea să fie aplicată la transmisia fasciculelor. Se poate arăta că rezoluția laterală a imaginilor poate îmbunătățită și că lobii laterali pot fi suprimați dacă focalizarea dinamică este folosită atât pentru transmisie cât și pentru recepție. Focalizarea dinamică în ambele sensuri poate fi folosită numai în vizualizarea prin apertură sintetică. În vizualizarea cu apertură reală implementarea focalizării dinamice la transmisie este impracticabilă deoarece ar necesita un timp de achiziție a datelor nelimitat.

Focalizarea compusă este un caz particular al focalizării dinamice, care folosește un număr mai mic de puncte de focalizare. Aceasta mărește adâncimea câmpului prin transmiterea câtorva secvențe de pulsuri, fiecare secvență focalizând la un punct diferit. Fasciculele cu lungimi focale diferite pot fi transmise numai una câte una. Cel de al doilea fascicul trebuie să fie transmis după ce toate ecourile produse de fasciculul anterior se întorc la traductor. Imaginile obținute prin transmisii cu lungimi focale diferite sunt decupate în jurul lungimii lor focale și montate una după alta formând astfel un nou cadru de imagine. Este evident că acest proces crește timpul de formare a cadrului de imagine. În practică trebuie avut în vedere un compromis între calitatea imaginilor și rata cadrelor.

Alegerea tipului de focalizare se face funcție de caracteristicile impuse sistemului de investigare și se traduce la final în constrângeri asupra resurselor hardware și software ale sistemului. Pentru sistemele de investigare cu apertură sintetică, la care constrângerile de timp real sunt eliminate, se poate folosi orice tip de focalizare cu observația că perioada de timp pentru sinteza unui cadru de imagine este diferită. O soluție des utilizată pentru sistemele care se apropie de modul de operare în timp real este folosirea focalizării fixe la transmisie și a focalizării dinamice la recepție. Această alegere presupune ca pentru fiecare cadru de imagine să se realizeze un număr de transmisii egal cu numărul de fascicule, și pe fiecare perioadă de recepție să se formeze un fascicul. Constrângerea care se aplică pentru această situație este cea dată de relația 4.1. Realizarea unui sistem care să opereze efectiv în timp real presupune conectarea în paralel a unui număr de structuri de recepție și formare de fascicul egal cu numărul de fascicule definit pentru sectorul explorat. Evident o astfel de soluție nu este practică dar folosirea de formatoare multifascicul poate în anumite cazuri să asigure un raport optim între complexitatea formatorului și frecvența cadrelor.

Pentru formatoarele prezentate în lucrare se vor folosi următoarele două combinații de focalizare:

- focalizare fixă la transmisie (FFTr) / focalizare fixă la recepție (FFRec)
- focalizare fixă la transmisie (FFTr) / focalizare dinamică la recepție (FDRec)

4.1.3 METODA DE EȘANTIONARE

Structura formatorului digital este definită pe lângă tipul de focalizare și de modalitatea de prelevare a eșantioanelor din semnalele ecou furnizare de canalele de recepție.

Eșantionarea este de fapt prima etapă din procesul de generare a fasciculului care mai cuprinde pentru formatoarele convenționale: **conversia analog numerică, întârzierea și însumarea**. În cazul **eșantionării uniforme** etapele sunt parcurse în ordinea descrisă mai sus fiind implementate fiecare pe suportul hard necesar. La **eșantionarea neuniformă** practic se poate vorbi de simultaneitatea etapelor de eşantionare și întârziere care sunt urmate de conversie și însumare. Este de așteptat ca la nivelul suportului hardware contopirea celor două etape să conducă la o reducere a complexității. Pe de altă parte expandarea structurii formatorului spre o configurație multifascicul pentru eşantionarea uniformă se realizează doar prin multiplicarea resurselor pentru întârziere și însumare pe când eşantionarea neuniformă cere multiplicarea practic a întregului canal de formare inclusiv a convertoarelor analog numerice.

4.1.4 CONVERSIA FORMATULUI DE SCANARE

Prin dirijarea și focalizarea fasciculului ultrasonic peste o secțiune transversală din mediul investigat se formează o imagine polară în coordonate r – adâncimea de investigare și θ - direcția fasciculului. Conversia formatului de scanare este necesară pentru trecerea de la coordonatele polare asociate informației ultrasonice la coordonatele carteziene. Imaginea carteziană este un spațiu imagine împărțit în regiuni elementare, de obicei rectangulare denumite pixeli. Dimensiunea pixelilor se fixează funcție de claritatea dorită pentru imagine. Pentru a evita „aspectul de bloc – blocky apperance” caracteristic imaginilor cu un număr redus de pixeli imaginea trebuie să conțină cel puțin 256x256 pixeli (512x512 de preferat) [Zar94]. Valoarea fiecărui pixel (x,y) se poate obține printr-o operație de mapare între imaginea carteziană și cea polară. Pixelul este mai întâi identificat în imaginea carteziană și apoi coordonatele sale sunt mapate fracționar în imaginea polară. Pentru fiecare pixel (x,y) din imaginea carteziană poziția corespunzătoare în imaginea polară poate fi obținută din următoarele ecuații:

$$r = \sqrt{x^2 + y^2} \quad (4.8)$$

$$\theta = \arcsin(x/r) \quad (4.9)$$

iar valoarea semnalului în poziția (r, θ) se obține prin interpolare deoarece din ecuațiile de transformare se obțin valori fracționare. Literatura referitoare la ultrasunete furnizează o cantitate redusă de informație privitoare la reconstrucția și reasamblarea imaginilor polare în imagini rectangulare. Teoria referitoare la interpolarea și reconstrucția optimă a semnalelor uni și bidimensionale este binecunoscută [Openheim89] dar puține studii au fost focalizate pe interpolarea imaginilor ultrasonice. Pentru implementările în timp real sunt de preferat dintre soluțiile de interpolare existente interpolarea biliniară sau interpolarea de tip ”cel mai apropiat vecin – nearest-neighbour” întrucât sunt cele mai eficiente din punctul de vedere al calculelor. Eforturile recente de îmbunătățire a vitezei de calcul s-au concretizat în algoritmi implementați software [Berkhof94] sau hardware în procesoare specializate pentru conversia de coordonate [Richard94]. Analiza efectuată în [Berkhof94] pentru diferite tipuri de interpolatoare arată de asemenea diferențe minime în ceea ce privește calitatea imaginii și superioritatea interpolării biliniare în ceea ce privește timpul necesar pentru calcule. Pe aceste considerente în emularea software din lucrare a fost folosită interpolarea biliniară. **Este de reținut, în ideea globală a lucrării de îmbunătățire a performanțele sistemelor de investigare ultrasonică prin folosirea de soluții hardware de viteză mare și complexitate rezonabilă, posibilitatea implementării hardware de procesoare convertoare de coordonate cu viteză mare de calcul.**

Metoda interpolării biliniare este ilustrată în figura 4.3. Pentru fiecare pixel din imaginea carteziană I2 se calculează cu relațiile 4.8 și 4.9 poziția corespunzătoare din

imaginea polară I_1 . Din I_1 sunt utilizate pentru interpolare patru eșantioane S_1, S_2, S_3, S_4 cele mai apropiate de această poziție. Valoarea pixelului se determină ca sumă ponderată a valorilor eșantioanelor. Presupunând că pozițiile eșantioanelor S_1, S_2, S_3, S_4 sunt date prin perechile de coordonate $(r_1, \theta_1), (r_1, \theta_2), (r_2, \theta_1), (r_2, \theta_2)$, valoarea pixelului în urma interpolării este:

$$p = S_1(\theta_2 - \theta)(r_2 - r) + S_2(\theta_2 - \theta)(r - r_1) + S_3(\theta - \theta_1)(r_2 - r) + S_4(\theta - \theta_1)(r - r_1) \quad (4.10)$$

unde (r, θ) este poziția pixelului p

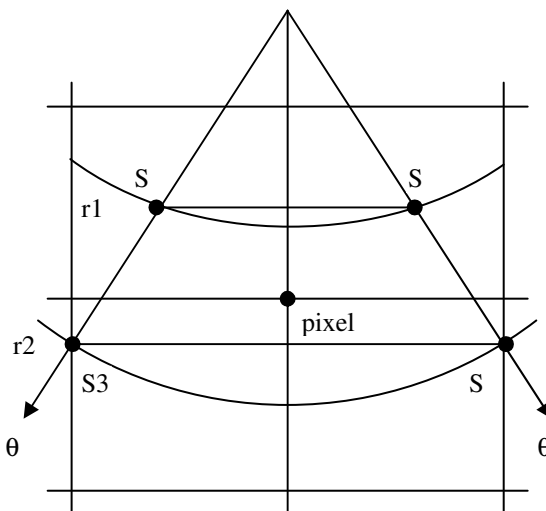


Fig. 4.3 Interpolarea biliniară

Programele pentru testarea diverselor variante de formatoare fac apel la o rutină specializată care modelează conversia formatului de scanare prin interpolare neliniară. În acest mod rezultatul scanării în coordonate polare este disponibil pentru evaluare calitativă sub forma imaginii în coordonate carteziane. Problema implementării hardware a unui procesor pentru conversia de format se plasează în afara obiectivelor acestei lucrări și nu a fost abordată.

4.2 FORMATORUL DE FASCICUL CU MODULAȚIE DELTA-SIGMA

Generarea fasciculelor de ultrasunete prin folosirea modulației sigma – delta își găsește justificarea în necesitatea simplificării structurii hardware a canalului receptor în condițiile folosirii de arii cu un număr tot mai mare de elemente (128; 256) pentru a obține imagini de calitate superioară. Această tehnică presupune prezența convertoarelor cu supraeșantionare în componența blocului digital de intrare și scheme simple de procesare pentru întârzierea și însumarea coerentă.

Folosind tehnici de supraeșantionare pot fi obținute imagini cu o calitate similară cu a celor obținute cu tehnicile digitale multi-bit de formare a fasciculelor, în condițiile reducerii semnificative a complexității și gabaritului sistemului. Formatoarele ce au la bază convertoare $\Delta\Sigma$ prezintă două avantaje majore față de metodele tradiționale multi bit.

I. Simplificarea structurii CAN și posibilitatea integrării pe același chip împreună cu blocurile de procesare digitală care creează fasciculul. Circuitele necesare pentru a implementa CAN $\Delta\Sigma$ sunt mult mai simple decât cele folosite în CAN flash multi-bit (câteva amplificatoare operaționale și un comparator față de 255 comparatoare necesare pentru un CAN flash pe 8 biți), fapt ce se regăsește în reducerea resurselor de interconectare, a gabaritului și puterii consumate.

II. Se pot asigura simplu întârzieri precise prin manipularea eșantioanelor prelevate cu o rată de eșantionare ridicată. Circuitele complicate pentru procesare în banda de bază sau

pentru interpolare / decimare sunt înlocuite de o memorie FIFO convențională poate asigura întârzierea cu rezoluția dorită înainte de însumare. În plus, însumarea pentru întreaga arie se simplifică întrucât primul nivel de sumatoare este alcătuit din sumatoare pe un bit.

Datorită domeniului dinamic larg necesar pentru aplicațiile ultrasonore, CAN $\Delta\Sigma$ trebuie să fie un modulator de ordinul doi sau trei pentru a menține un raport semnal zgomot adecvat. Pentru a asigura atât un raport semnal zgomot ridicat cât și o acuratețe adecvată a întârzierilor s-a ales o frecvență de eșantionare de cel puțin 32 ori mai mare decât frecvența purtătoare ceea ce revine la un factor de supraeșantionare $OSR = 16$. Cuantificatorul de la ieșirea modulatorului se alege în mod obișnuit cu două niveluri de cuantizare datorită simplității sale. Adăugarea de niveluri de cuantificare îmbunătățește raportul semnal zgomot al sistemului dar complică circuitul de reacție, în special CNA – ul din componența modulatorului.

În varianta cu eșantionare uniformă, figura 4.4, ieșirea digitală a modulatorului se aplică liniilor digitale de întârziere cu lungime variabilă, implementate ca memorii FIFO sau registre de deplasare, care realizează întârzierea dinamică a semnalelor. Aceste registre de deplasare trebuie să opereze la frecvența de eșantionare și conțin câteva sute de celule pentru a găzdui întârzierile dinamice pe durata fiecărui fascicul. Lungimea liniei de întârziere depinde de următorii parametrii:

- frecvența de eșantionare,
- geometria ariei,
- unghiul de dirijare,
- adâncimea maximă de investigare,
- frecvența purtătoare (frecvența centrală a traductorului).

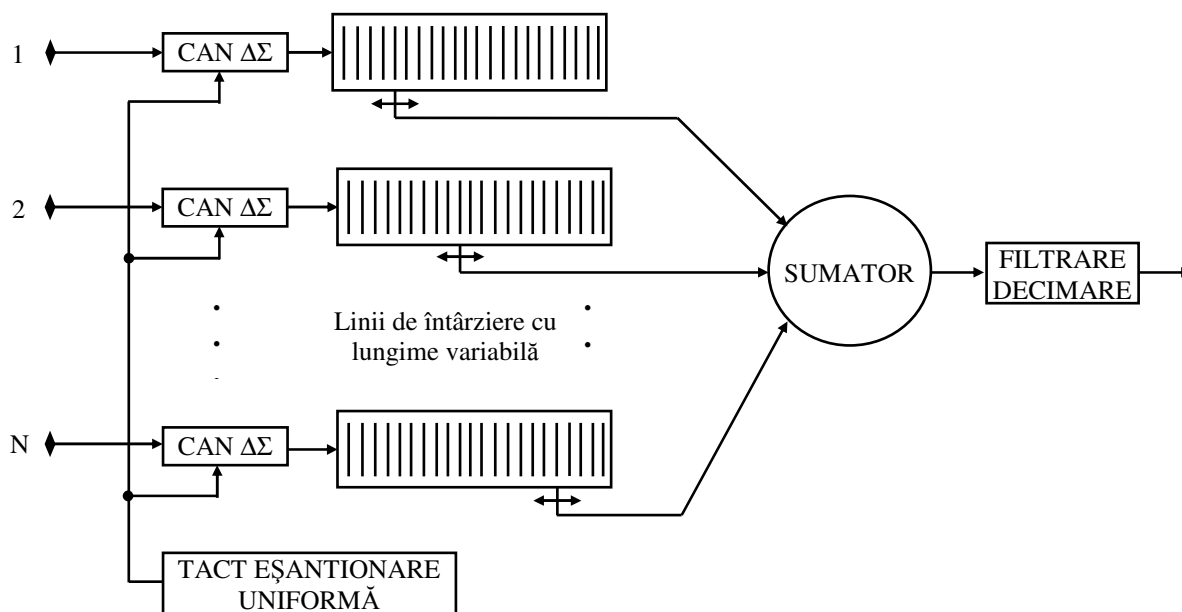


Fig. 4.4 Formator cu modulație delta-sigma uniformă

Când lungimea liniei de întârziere se modifică pentru a produce o nouă zonă focală, eșantioanele trebuie repetate, inserate sau eliminate la intrarea, ieșirea sau în punctul median a registrului.

Eșantioanele întârziate de la toate elementele ariei sunt apoi însumate digital pentru a forma fasciculul. Această însumare trebuie realizată la frecvența de eșantionare și necesită un set de sumatoare digitale pipeline pentru întreaga arie. Odată fasciculul generat prin sumare, semnalul se aplică unui filtru trece jos cu pantă abruptă pentru a elimina zgomotul de cuantizare $\Delta\Sigma$. Acest filtru de ordin superior poate fi foarte lung și complicat și cu un anumit consum dar este necesar un singur astfel de filtru pentru generarea unui fascicul. Pentru

folosirea în continuare a blocurilor de procesare tradiționale (dectecție de anvelopă, conversie format de scanare) este necesară reducerea ratei de eşantionare prin decimarea şirului filtrat de eşantioane.

4.3 EMULAREA SOFTWARE A FORMATOARELOR DE FASCICUL

Pentru a evidenția performanțele diverselor arhitecturi de formatoare de fascicul, acestea au fost emulate software folosind semnalele ultrasonice eşantionate și înregistrate în “Biomedical Ultrasonics Laboratory” al Universității din Michigan, disponibile pe internet la adresa <http://bul.eecs.umich.edu>.

Semnalele ultrasonice provin de la o configurație de test alcătuită din 6 fire plasate într-un vas cu apă, la distanțe de 34, 48, 65, 83, 101 și 121 mm față de traductor, perpendicular pe planul de scanare.

Datele au fost achiziționate cu un sistem experimental bazat pe o arie de 128 traductoare cu frecvența de 3.5 MHz și distanța interelement egală aproximativ cu $\lambda/2$ (0.22 mm). Toate elementele au fost folosite atât pentru transmisie cât și pentru recepție. O înregistrare conține 128 secvențe în care un element emite și toate cele 128 elemente recepționează semnalele ecou. Combinația elementelor ca emițător / receptor dă un total de $128 * 128$ linii de scanare în mod A. Fiecare scanare mod A a fost eşantionată la 13,89 MHz cu un offset de 29,448 uS, și conține 2048 eşantioane corespunzător unei extinderi spațiale de 130 mm. Pentru a reduce zgomotul introdus de sistemul de achiziție a datelor s-a recurs la medierea semnalelor rezultate din achiziții succesive iar datele finale au fost stocate folosind cuvinte pe 16 biți. Figura 4.5 prezintă trei semnale mod A recepționate de elementele 2, 64 și 127 în situația când emite elementul 1. Se observă modificarea pozițiile ecourilor provenite de la cele 6 fire ca urmare a modificării drumului parcurs de unda ultrasonică în cele trei configurații geometrice. Semnalele mod A indică prezența celor 6 puncte de reflexie (împrăștiere) localizate pe circumferințele unor cercuri având ca centru elementul receptor și raza proporțională cu numărul eşantionului. În mod evident asimilarea elementului de arie cu o sursă de radiație sferică omnidirecțională nu permite localizarea spațială a reflectorului prin investigarea în modul A.

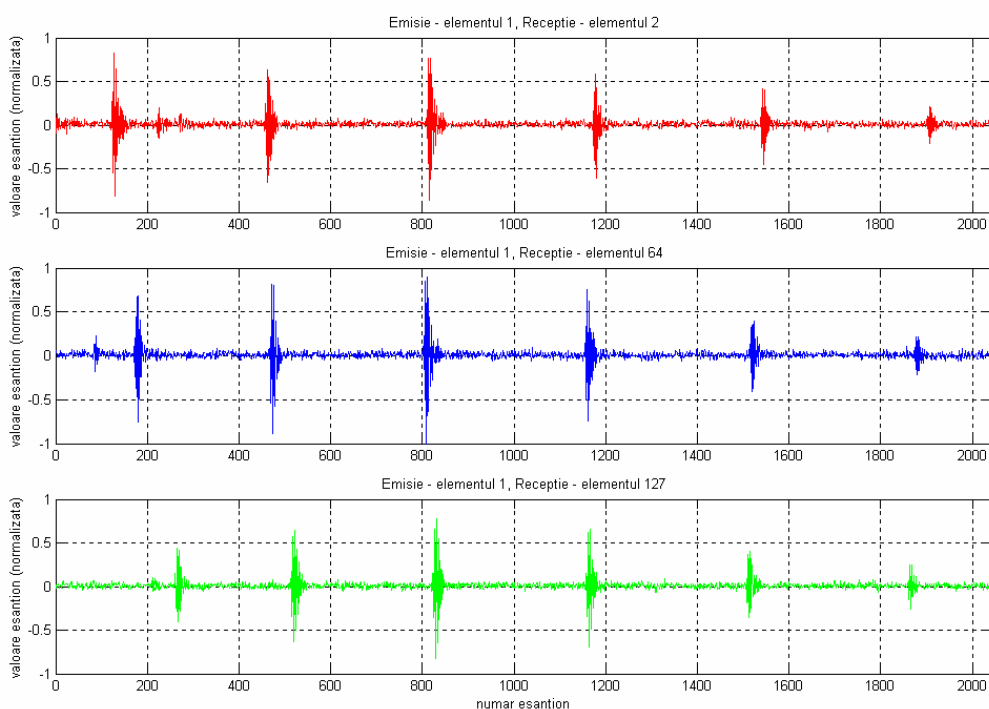


Fig.4.5 Semnalele mod A recepționate de elementele 2, 64 și 127 când emite elementul 1

Analiza densităților puterii spectrale pentru un set de date (128 linii mod A) arată distribuția de puterii spectrale a semnalelor recepționate în banda de frecvență a ariei de traductoare (figura 4.8) și indică datele de proiectare a unui filtru pentru eliminarea zgomotului adițional.

S-a implementat pentru aceasta un filtru FIR trece bandă. Cea mai simplă metodă de proiectare a unui astfel de filtru este metoda ferestrei care constă în trunchierea răspunsului ideal la impuls pentru o fereastră de lungime finită. Ferestrele folosite în mod curent sunt Hamming, Hanning, Blackman, Bartlet și Kaiser.

Fereastră Kaiser, folosită pentru proiectarea filtrului FIR în această lucrare, este definită prin:

$$w[n] = \begin{cases} \frac{I_0\left[\beta\sqrt{1-\left[(n-\alpha)/\alpha\right]^2}\right]}{I_0(\beta)}, & 0 \leq n \leq M \\ 0 & \text{în rest} \end{cases} \quad (4.11)$$

unde M este dimensiunea ferestrei, $\alpha=M/2$, iar $I_0(\cdot)$ este funcția Bessel de spița 1 modificată, de ordinul zero. Prin modificarea lui M și a parametrului β pot fi ajustate dimensiunea și forma ferestrei pentru a obține un raport optim între lățimea lobului principal și amplitudinea lobului lateral – figura 4.6. Deoarece în imagistica ultrasonică este preferat un nivel scăzut al lobilor laterali se alege pentru β o valoare mare ($\beta = 7$)

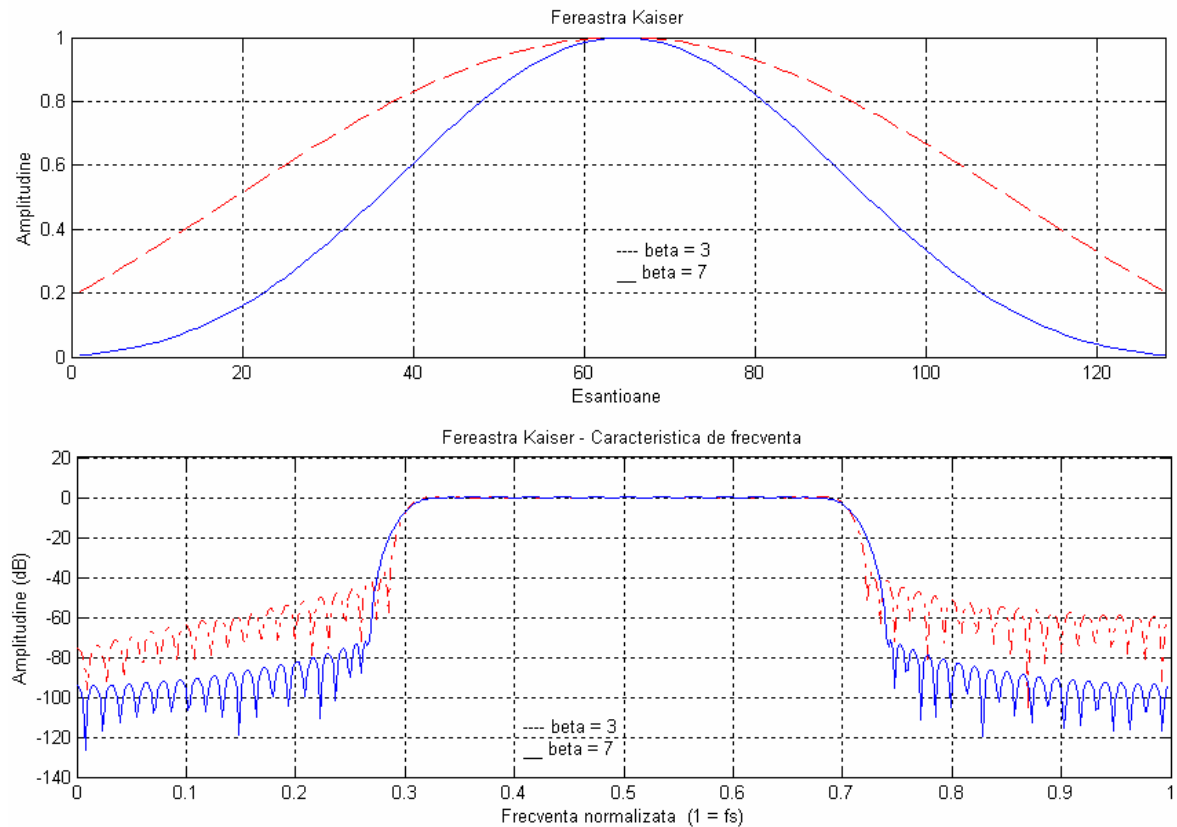


Fig.4.6 Sus: Fereastră Kaiser pentru $M=128$, $\beta = 3$ și $\beta = 7$. Jos: caracteristica de frecvență

Din figura 4.8 se observă ca energia semnalului este distribuită aproximativ simetric între 0,3 și 0,7 în domeniul frecvențelor normalizate ceea ce corespunde la 2,1MHz și 4,9MHz în frecvența reală. Caracteristicile filtrului Kaiser tip trece bandă cu frecvențele de tăiere indicate mai sus, de dimensiune $M=128$ și cu $\beta = 7$ sunt date în figura 4.7. Așa cum era de așteptat

spectrul semnalului filtrat – fig.4.8 – nu mai conține zgomot adițional în exteriorul benzii utile.

Cu ajutorul datelor filtrate a fost posibilă comparația între diverși algoritmi de formare de fascicule, identificarea și testarea unor soluții de implementare hardware a structurilor de prelucrare de viteză mare și complexitate acceptabilă.

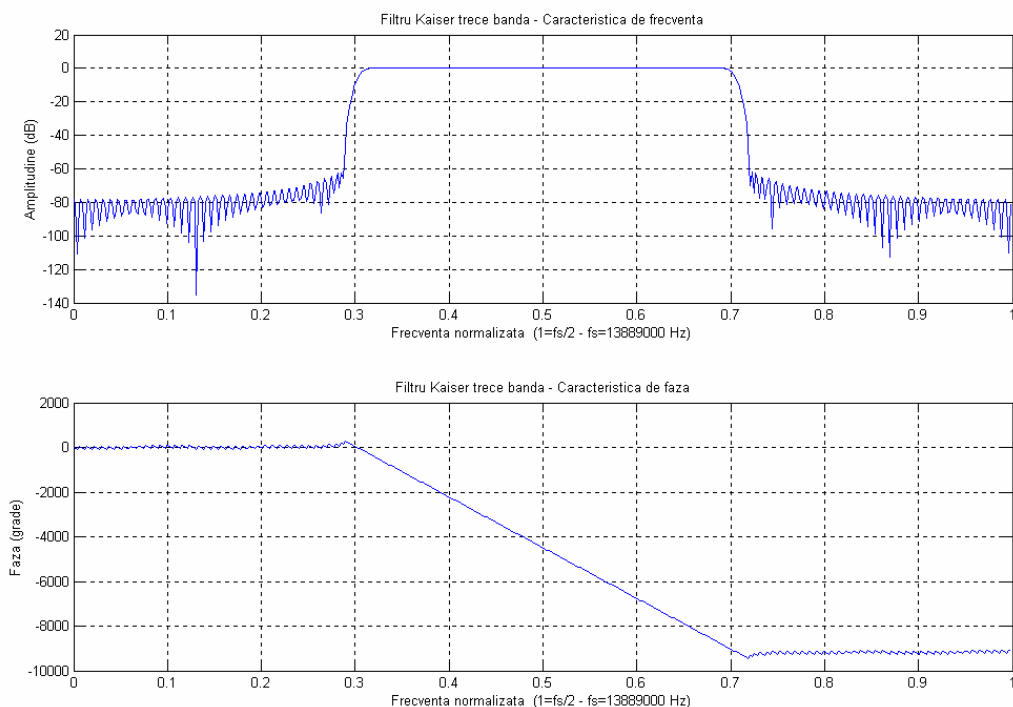


Fig.4.7 Caracteristicile filtrului Kaiser trece bandă

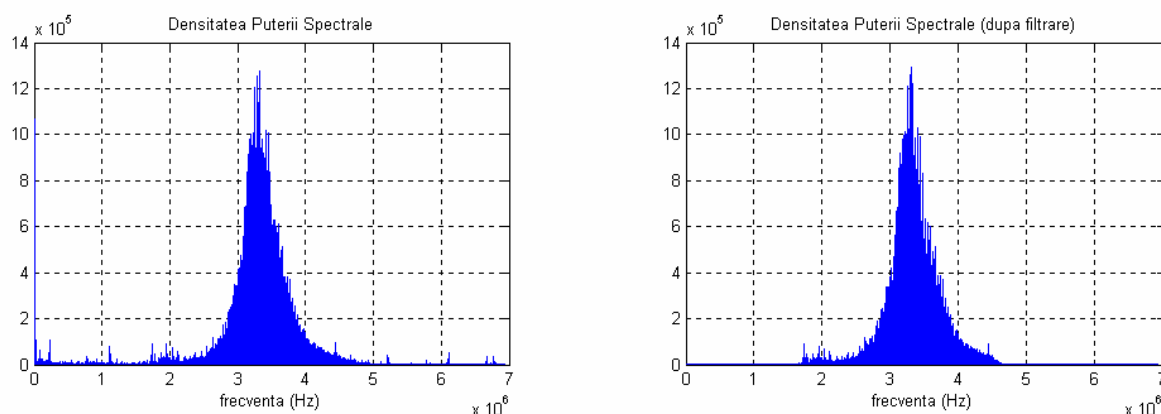


Fig.4.8 Spectrul semnalelor mod A înainte (stânga) și după (dreapta) filtrare

4.4 Simularea unui formator de fascicul – varianta convențională

Calea de semnal a unui sistem imagistic ultrasonic conține subsistemele de prelucrare a informației de la conversia semnalelor acustice în semnale electrice la nivelul ariei de traductoare și până la obținerea unei imagini într-un format accesibil utilizatorului (coordonate carteziene).

Din punct de vedere al tipului informației prelucrate există un **nivel de procesare analogică** care include etajul preamplificator, amplificatorul cu câștig variabil și blocul de conversie analog numerică și un **nivel de procesare digitală** având ca blocuri principale formatorul de fascicul și convertorul de coordonate. Blocurile care fac obiectul lucrării de față: formatorul

de fascicul și convertorul delta-sigma pot fi testate soft folosind setul de date în format numeric rezultate în urma conversiei analog numerice multibit (10 biți), un program care implementează algoritmul de formare a fascicului ultrasonic și o rutină care descrie software conversia de la coordonatele polare la cele carteziane. Rutinele au fost scrise în Matlab deoarece mediul dispune de resurse puternice de prezentare a informației în format grafic (imagini și grafice).

Scanarea mediului se face în coordonate polare pentru un număr de direcții (fascicule) NBeam dat de relația 4.6 și un număr de eșantioane pe fiecare direcție $N_{es} = 2048$ egal cu numărul de eșantioane dintr-o înregistrare mod A.

S-a luat ca referință două variante ale formatorului clasic tip întârziere-însumare (DSBF Delay-Sum BeamFormer) care operează cu datele de intrare livrate de convertoare analog numerice multibit :

- cu focalizare fixă atât la transmisie cât și la recepție DSBF-FFTrFFRec (Delay-Sum BeamFormer – Fixed Focus Transmission, Fixed Focus Reception)
- cu focalizare fixă la transmisie și focalizare dinamică la recepție DSBF-FFTrFDRec (Delay-Sum BeamFormer – Fixed Focus Transmission, Dynamic Focus Reception).

O primă rutină a programului calculează, pornind de la configurația geometrică a ariei și de la datele de scanare, matricea întârzierilor față de centrul ariei, pentru fiecare punct focal de pe fiecare direcție și pentru fiecare element, folosind relațiile geometrice date în capitolul 2 (2.25-27). Informația stocată în această matrice tridimensională ($N_{Beam} * N_{Eșantion} * N_{Elem}$) permite calculul parcursului dus-întors (Emisie-Recepție) pentru orice pereche de elemente și întârzierea față de elementul din centrul ariei.

Algoritmul de formare a fascicului constă în:

- selecția punctului curent pe o direcție dată,
- determinarea întârzierii față de centrul ariei pentru un element dat,
- identificarea printr-un proces de interpolare a eșantionului recepționat de elementul dat din punctul considerat,
- însumarea contribuțiilor recepționate de fiecare element al ariei.

Procesul de reia pentru toate punctele de pe o direcție și pentru toate direcțiile. Pentru a crește precizia de formare a fascicului s-a procedat la supraeșantionarea setului original de date printr-un proces de interpolare care crește frecvența de interpolare de un număr de ori. Codul sursă pentru toate variantele de formatoare este indicat în Anexa 1.

4.5 SIMULAREA FORMATORULUI CU MODULAȚIE DELTA-SIGMA

Pentru simularea formatorului cu modulație delta-sigma $\Sigma\Delta$ DSBF (Delta-sigma Delay-Sum BeamFormer) se includ în structura programului de simulare pentru formatorul clasic DSBF următoarele faze suplimentare:

- extinderea cu zerouri urmată de interpolare pentru a asigura fluxul de date conform cu factorul de supraeșantionare (OSR) dorit
- normalizarea datelor de intrare pentru a evita saturația modulatorului Delta-sigma
- conversia Delta-sigma
- filtrarea trece jos pentru eliminarea zgomotului de înaltă frecvență introdus de modulatorul Delta-sigma
- decimarea șirului de date rezultat la ieșirea formatorului pentru a asigura o frecvență a datelor compatibilă cu ieșirea unui formator convențional

Exceptând conversia Delta-sigma pentru care s-au folosit rutinele dedicate acestui scop din ToolBox pentru toate fazele enumerate mai sus au fost concepute module originale în Matlab.

Imaginile în coordonate carteziane obținute ca rezultat al simulării permit o apreciere calitativă asupra procesului de formare a fascicului. Pentru a evalua cantitativ performanțele formatoarelor s-a folosit analiza spectrală la nivel de fascicul și reprezentarea fascicului în domeniul timp la scară logaritmică.

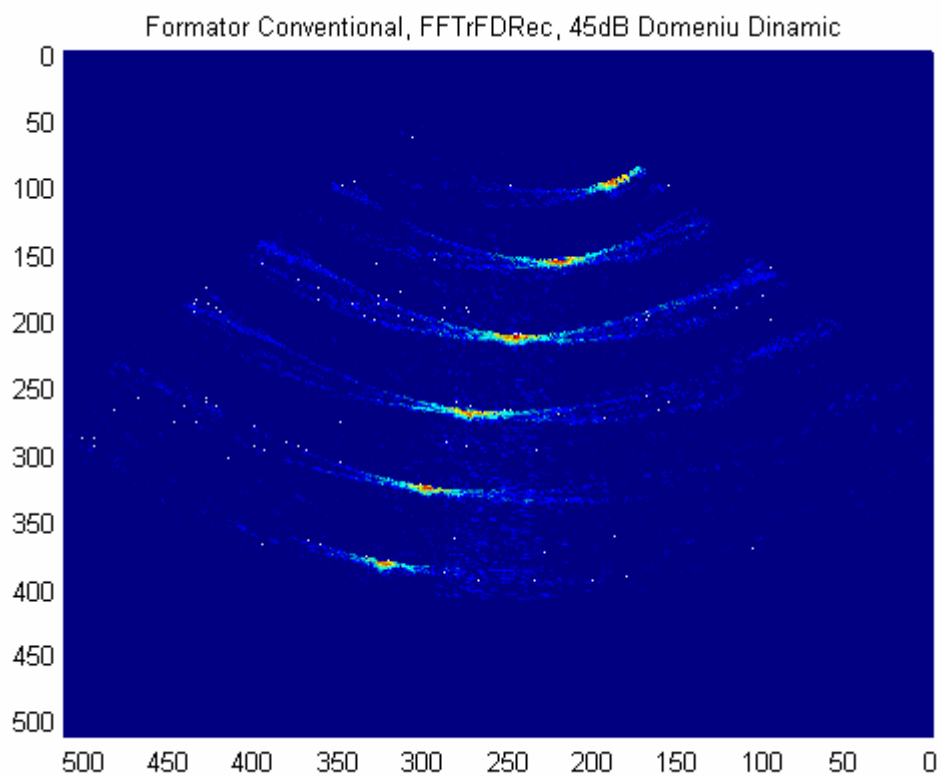


Fig.4.9 Imagine 512x512 generată cu un formator convențional

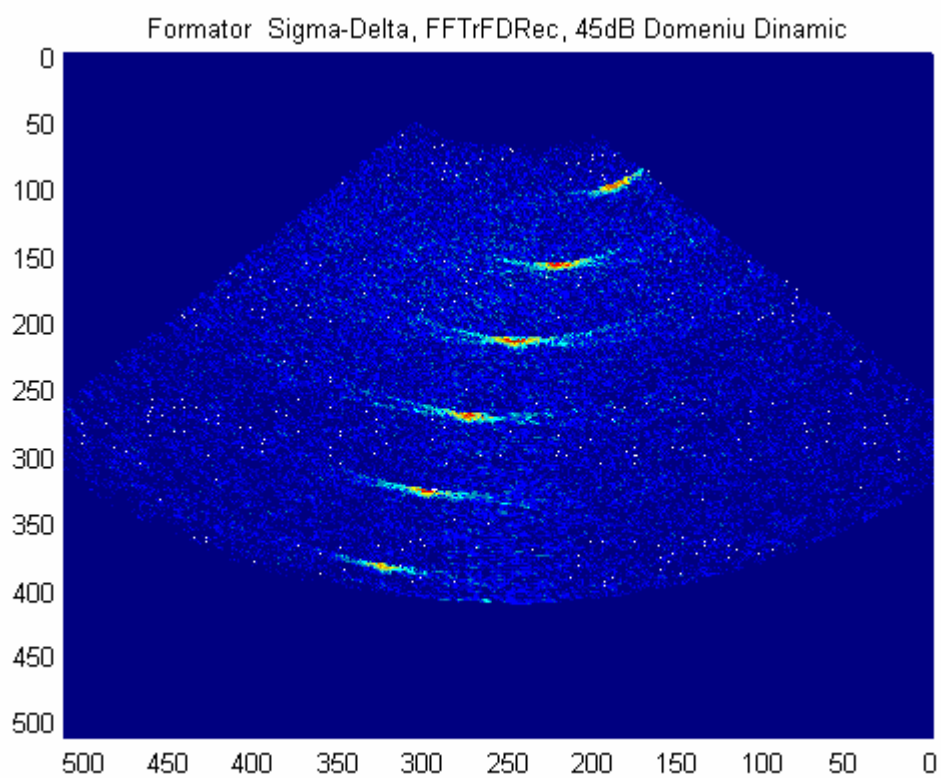


Fig.4.10 Imagine 512x512 generată cu un formator Delta-sigma cu eșantionare uniformă

Figura 4.9 prezintă o imagine generată cu formatorul convențional multibit care prelucrează eşantioane codate pe 10 biți, iar figura 4.10 prezintă aceeași imagine generată cu un formator Delta-sigma cu eşantionare uniformă. Este evidentă prezența „neașteptată” a unui zgomot suplimentar în imaginea generată cu formatorul delta-sigma.

Cauza apariției acestui zgomot a fost detectată a fi modificarea dinamică a întârzierilor în timpul procesului de formare de fascicul. În cele ce urmează sunt indicate modalități de compensare a influenței focalizării dinamice asupra calității imaginilor produse cu formatoare delta-sigma.

Convertorul analog numeric CAN $\Delta\Sigma$ împreună cu filtru de reconstrucție trece jos constituie o pereche modulator – demodulator. Între modulator și demodulator pot fi inserate blocuri care execută operații liniare fără ca aceasta să conducă la degradarea performanțelor (în principal în ceea ce privește raportul semnal zgomot).

Procesul de formare al fasciculului ultrasonic prin algoritmul întârziere – însumare poate fi descompus în două etape distincte :

- întârzierea – care constă în decalarea temporală a informației furnizate de elementele ariei de traductoare
- însumarea semnalelor decalate

Focalizarea undelor acustice în câmp apropiat necesită, așa cum rezultă din capitolul 2, întârzierea după o lege parabolică. Prin urmare formarea fasciculelor ultrasonice se realizează printr-o operație neliniară de întârziere urmată de operația liniară de însumare.

Folosirea convertoarelor delta-sigma cu modulație uniformă pentru formarea fasciculelor presupune intercalarea între modulator și demodulator a procesului neliniar de întârziere urmat de procesul liniar de însumare – figura 4.11. Neliniaritatea întârzierii conduce la reeşantionarea semnalelor livrate de elementele de arie care se traduce în folosirea repetată a unor eşantioane sau ne folosirea altora. Repetarea sau eliminarea de eşantioane din şirul rezultat în urma conversiei delta-sigma cu rată uniformă duce la desincronizarea ansamblului modulator / demodulator. Preponderența unuia sau a altuia dintre mecanismele de desincronizare este funcție de raportul între frecvența de eşantionare (f_M) și frecvența de formare (f_{BF})

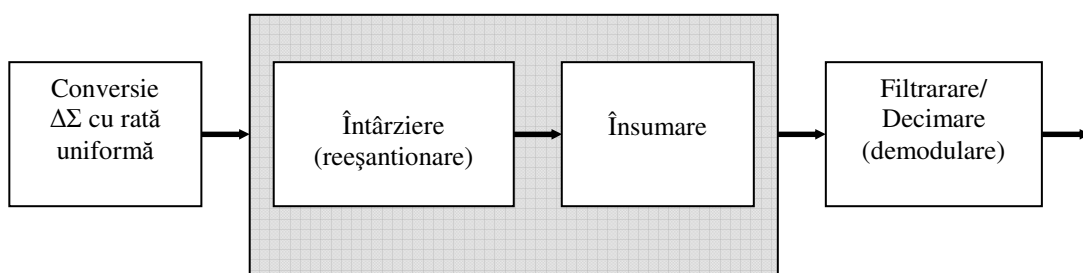


Fig. 4.11 Principiul formării fasciculelor prin modulație delta-sigma uniformă

Figurile 4.12 și 4.13 ilustrează procesul de reeşantionare și dependența de raportul f_M/f_{BF} (rezultatele au fost obținute prin simulare). Așa cum era de așteptat pentru $f_M \leq f_{BF}$ predomină fenomenul de neutilizare de eşantioane, pe când pentru $f_M \geq f_{BF}$ preponderentă este utilizarea de eşantioane în mod repetat. Situația este descrisă cantitativ în tabelul 4.1.

În condiții de operare normală diferențele introduse de funcționarea modulatorului sunt eliminate de filtrul demodulator, rezultând în final o reprezentare exactă a semnalului de intrare. Filtrul demodulator interpretează apariția unui eşantion suplimentar ca urmare a repetării în procesul de formare a fasciculului sau lipsa unui eşantion ca o diferență dată de modulator dar care de fapt nu există. Prin urmare filtrul injectează sau extrage o cantitate suplimentară de energie în/din semnalul de ieșire, dar care nu este prezentă în semnalul de intrare.

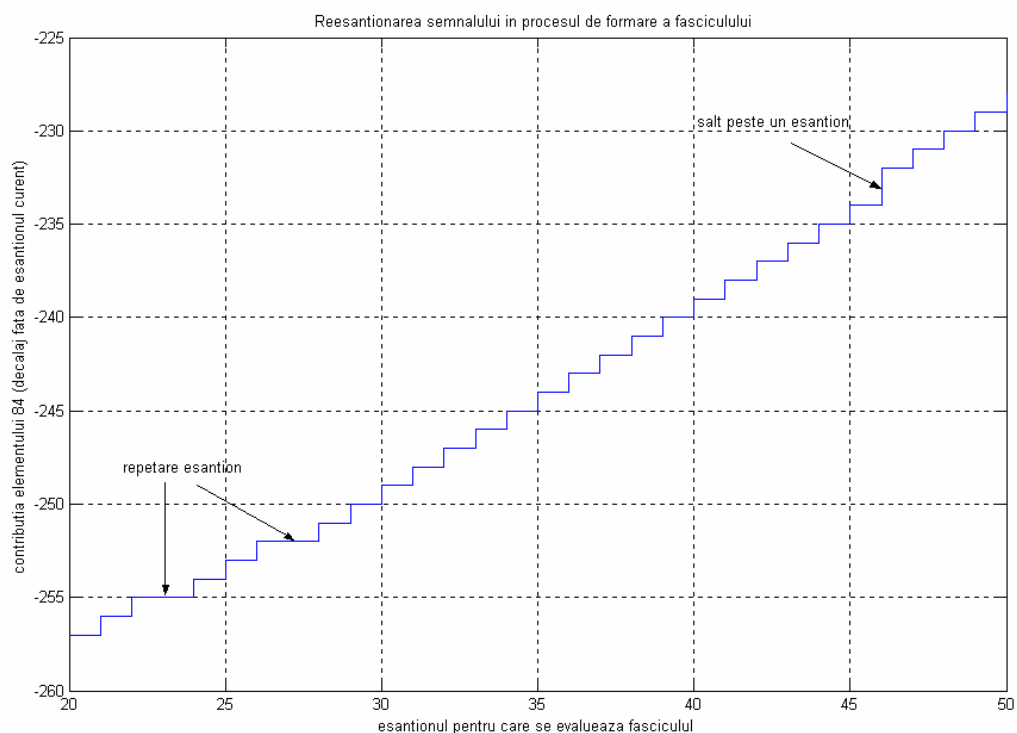


Fig. 4.12 Reesantionarea semnalului la formarea fasciculelor prin modulație delta-sigma uniformă

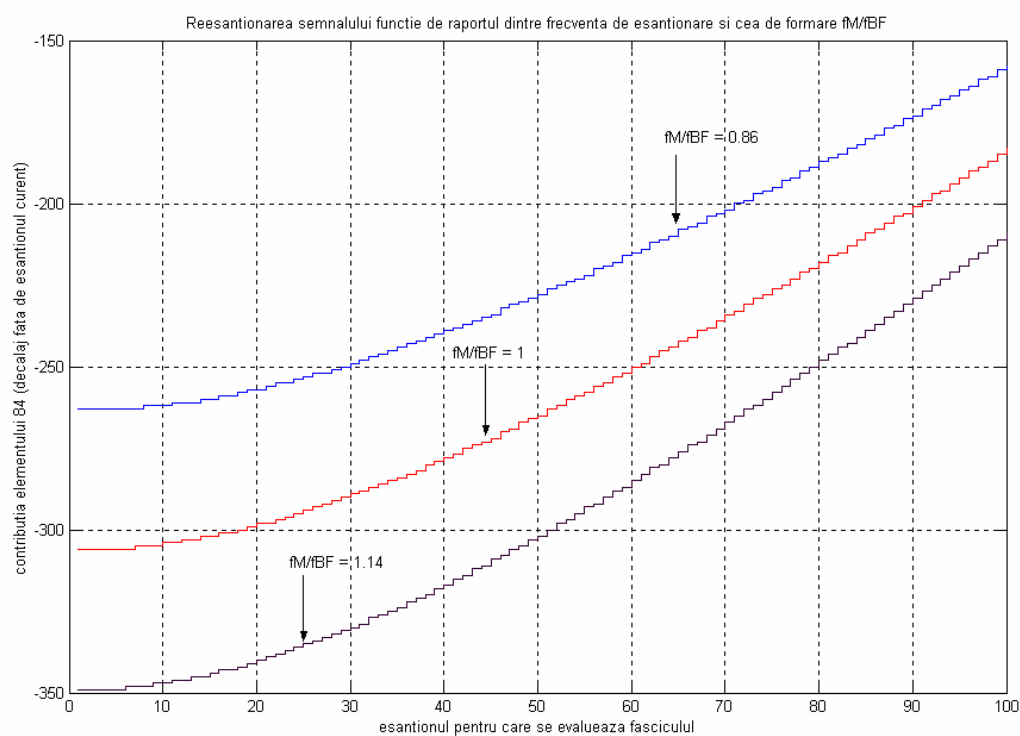


Fig. 4.13 Dependența procesului de reesantionare de raportul f_M/f_{BF}

Tabel 4.1 Dependența desincronizărilor convertor $\Sigma\Delta$ - filtru de raportul f_M/f_{BF}

f_M/f_{BF}	Eșantioane Repetate	Eșantioane neutilizate	Eșantioane sincrone	Total eşantioane
1,25	172	5581	18822	24575
1,14	279	2986	21310	24575
1,04	477	728	23370	24575
1	731	0	23844	24575
0,96	1713	0	22862	24575
0,86	4169	0	20406	24575
0,75	6869	0	17706	24575

O soluție simplă este să se forțeze modulatorul să țină seama în funcționare de eşantionul repetat sau eliminat asigurând prezența acestuia în sirul de eşantioane ce rezultă în urma modulării. Pentru aceasta se include în bucla de reacție un multiplexor care alege între amplitudine zero, amplitudine normală sau amplitudine multiplicată cu 2. Eşantioanele care sunt repetate ulterior în blocul de întârziere trebuie să aibă o pondere dublă pe calea de reacție deoarece fiecare eşantion apare de două ori la demodulator. Eşantioanele eliminate în procesul de formare a fasciculului au pondere zero pe calea de reacție deoarece acestea nu sunt “văzute” de demodulator. Evident, eşantioanele normale au un factor de reacție unitar. Acest tip de modulator care include multiplexorul în reacție este denumit modulator delta-sigma compensat – figura 4.14.

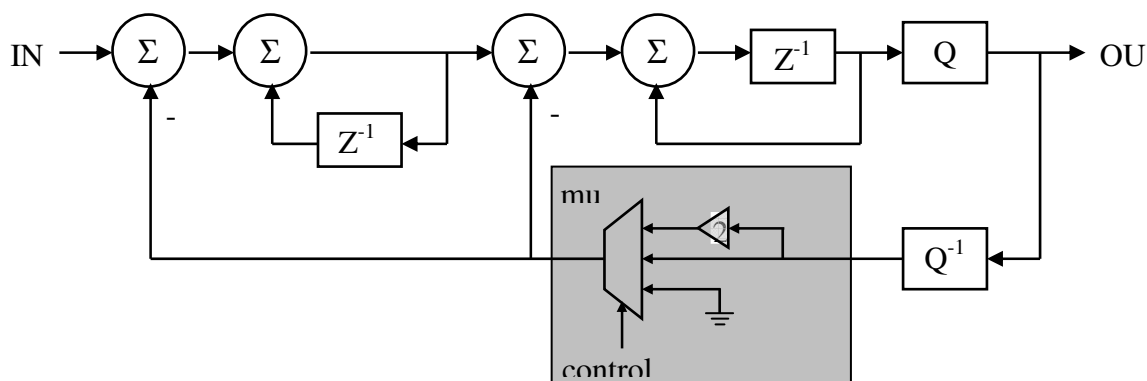


Fig. 4.14 Modulator delta-sigma compensat

Extinderea şirului de eşantioane la nivelul formatorului şi implicit creşterea rezoluţiei întârzierilor se realizează prin una din următoarele metode:

- repetarea unui eşantion;
- inserarea unui eşantion nul;
- descompunerea (scindarea) eşantionului în două părţi egale;

Pentru situațiile în care se descompune eşantionul sau se introduce un eşantion nul nu este necesară intervenția asupra modulatorului pentru a compensa şirul de eşantioane. Aceste metode nu afectează puterea medie a semnalului şi ca urmare modulatorul şi filtrul de reconstrucție rămân sincronizate. Prin contrast, repetiția unui eşantion conduce la existența a două eşantioane de aceeaşi mărime şi ca urmare modulatorul trebuie compensat. Aceste trei situații au fost testate pe datele achiziționate pentru o linie de investigare în mod A – fig. 4.15. Semnalul original a fost supus unui ciclu modulare-demodulare în care s-au implementat pe rând cele trei metode de extindere a şirului de eşantioane. Atât reprezentarea în domeniul timp – figura 4.15 cât şi spectrul semnalelor reconstruite - figura 4.16 confirmă necesitatea compensării în cazul repetării eşantioanelor.

Metodele de menținere a sincronismului prin „inserarea unui eşantion nul” sau prin „descompunerea eşantionului în două părți egale” sunt mai simple decât compensarea directă deoarece necesită doar o simplă recodare a eşantioanelor la ieșire pentru a modifica șirul eşantioanelor întârziate.

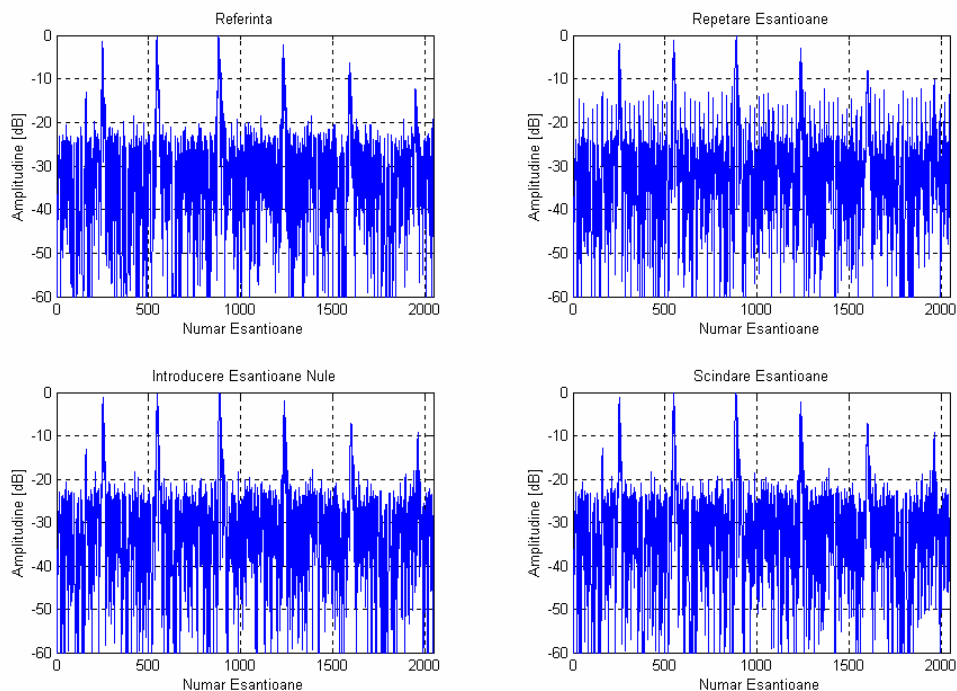


Fig. 4.15 Compararea tehnicilor de extindere a șirului de eşantioane – reprezentare în domeniul timp

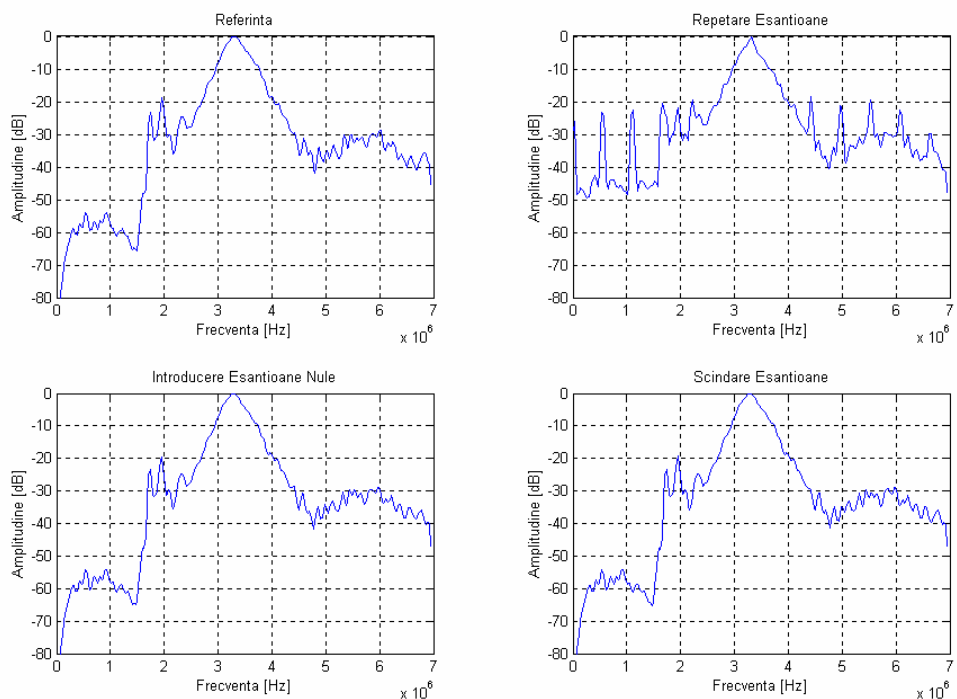


Fig. 4.16 Compararea tehnicilor de extindere a șirului de eşantioane – spectrul de frecvență

Pentru recodarea digitală ieșirea pe un singur bit a modulatorului, reprezentând +1 sau -1 trebuie modificată astfel ca noua valoare digitală multibit să reprezinte noile nivele necesare

pentru focalizarea dinamică. La inserarea unui eşantion noul se adaugă un al treilea nivel - zero – şi este necesar la nivel digital un singur bit suplimentar. Cele trei nivele pot fi codificate în două moduri folosind bitul suplimentar – tabelul 4.2

Tabelul 4.2

Ieşire binară $\Sigma\Delta$	Nivel digital	Cod nou (fără semn)	Cod nou (cu semn)
1	+1	10 (2)	01 (1)
0	-1	00 (0)	11 (-1)
Nivel nou	0	01 (1)	00 (0)

Tehnica prin „descompunerea eşantionului în două părţi egale” necesită doi biţi adiţionali datorită poziţionării noilor nivele între nivelele de +1 şi -1. (tabelul 4.3)

Tabelul 4.3

Ieşire binară $\Sigma\Delta$	Nivel digital	Cod nou (fără semn)	Cod nou (cu semn)
1	+1	100 (4)	010 (2)
Nivel nou	+0.5	011 (3)	001 (1)
Nivel nou	-0.5	001 (1)	111 (-1)
0	-1	000 (0)	110 (-2)

Deoarece, la performanţe egale, este mai simplă, metoda de introducere de eşantioane nule este preferată pentru modificarea dinamică a întârzierilor la formatoarele bazate pe modulaţia delta-sigma cu eşantionare uniformă.

Toate metodele de păstrare a sincronizării modulator – demodulator indicate presupun intervenţia asupra structurii hardware a modulatorului şi circuite de control suplimentare pentru detecţia desincronizărilor. Extragerea procesului neliniar de întârziere şi plasarea lui înaintea modulatorului delta-sigma elimină problema desincronizării dintre convertor şi filtru. Acest lucru este posibil dacă se foloseşte eşantionarea neuniformă.

4.6 FORMATOR CU MODULAŢIE DELTA-SIGMA NEUNIFORMĂ

Formarea de fascicul prin conversie sigma – delta cu rată neuniformă constă în extragerea procesului neliniar de întârziere din ansamblul modulator / demodulator şi plasarea acestuia înaintea modulatorului. În acest mod se asigură sincronizarea perechii modulator / demodulator întrucât între acestea este intercalată doar operaţia liniară de însumare – figura 4.17.

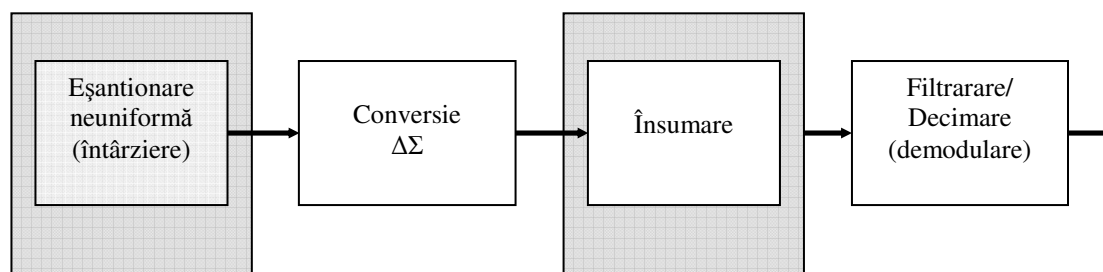


Fig. 4.17 Principiul formării fasciculelor prin modulație delta-sigma neuniformă

Formatoarele digitale care se bazează pe eşantionarea neuniformă realizează focalizarea dinamică prin eşantionarea semnalelor ecou la momente de timp care asigură condiţia de însumare coerentă la recepţie. Tactul pentru eşantionarea neuniformă este generat pe seama setului de întârzieri stocat într-o memorie digitală operată cu un tact uniform f_M .

Generatorul tactului de eşantionare produce semnale neuniforme de tact care sunt diferite pentru fiecare element al ariei, conform expresiei [Song90]:

$$c_n(t) = \sum_{p=0}^{P-1} m(t) \delta \left(t - \frac{1}{f_m} Q \left[p \frac{f_m}{f_{BF}} - f_m \tau_n(r_p, \theta) \right] \right) \quad (4.12)$$

unde $r_p = r_0 + p(c/2f_s)$, iar P este numărul total de puncte focale din domeniul r_0 la R cu incrementul Δr . Funcțiile $Q[\cdot]$ și $\delta(\cdot)$ reprezintă rotunjirea la „cel mai mare întreg mai mic decât” și respectiv impulsul Delta Dirac, iar $m(t) = 1$ pentru $0 \leq t < 1/(2f_m)$, $m(t) = 0$ în rest.

Eșantioanele întârziate dinamic sunt apoi digitizate de un set de modatoare delta-sigma pe un bit, înainte de însumarea coerentă, fiecare modulator fiind activat cu tactul neuniform produs de generator – figura 4.18. Ieșirea fiecărui modulator este un șir de date pe un bit care include semnalul original întârziat plus zgomotul de cuantificare de înaltă frecvență. Aceste reprezentări pe un bit ale eşantioanelor ecou apar la diverse momente de timp pe canalele de recepție deoarece nu există o regularitate la sosirea ecourilor. Dacă perioada de timp între două puncte focale consecutive $1/f_{BF}$ este mai mare decât întârzierea maximă dintre elemente ($\max \{\tau_n(r, \theta)\}$), atunci pentru orice punct focal, eşantionul cerut în procesul de formare de la fiecare element apare înaintea eşantionului necesar pentru punctul focal următor. Parametrii sistemului trebuie aleși astfel ca $1/f_{BF} < \max \{\tau_n(r, \theta)\}$, și, prin urmare toate eşantioanele care apar în intervalul de timp $\max \{\tau_n(r, \theta)\}$ trebuie memorate temporar. În acest scop structura conține pe fiecare canal un buffer FIFO pentru cuvinte pe un bit având lungimea dată de $(\max \{\tau_n(r, \theta)\})f_{BF}$.

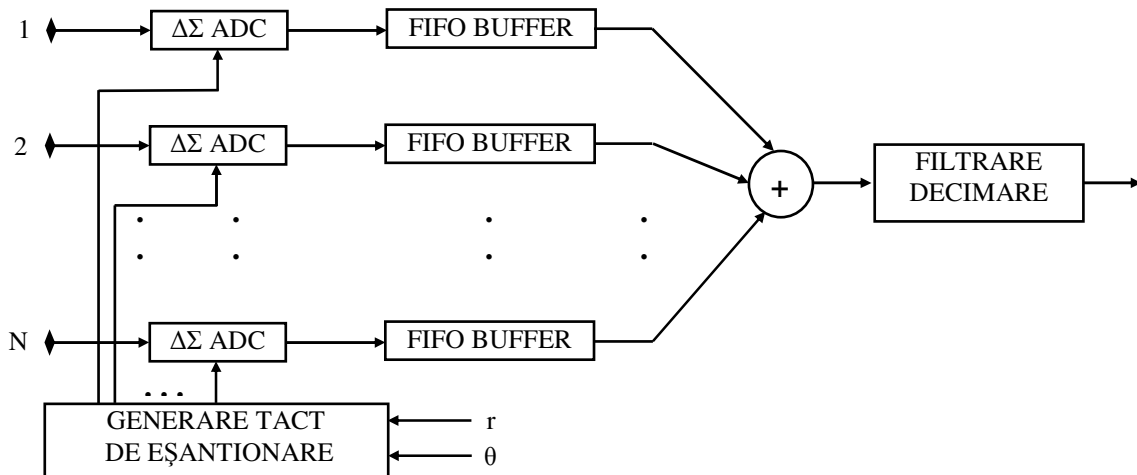


Fig. 4.18 Structura formatorului cu modulație delta-sigma neuniformă

Eșantioanele codate pe un bit care apar în diverse momente de timp la ieșirile modatoarelor sunt aliniate prin bufferele FIFO de pe fiecare canal și vor fi transmise simultan sumatorului care generează versiunea grosieră a fasciculului ce conține suma ecourilor întârziate plus zgomotul de cuantificare. Fasciculul este procesat de filtrul de reconstrucție care elimină zgomotul de cuantificare și reduce rata de eşantionare aproape de rata Nyquist

4.6.1 EVITAREA REPETIȚIEI EȘANTIOANELOR ÎN PROCESUL DE FORMARE A FASCICULULUI ULTRASONIC

Figura 4.2 indică distanțele de la două puncte focale consecutive r și $r + \Delta r$ la elementul n notate ρ_1 și respectiv ρ_2 . Evident undele ultrasonice reflectate r și $r + \Delta r$ ating elementul n al

ariei la momente diferite de timp, decalajul temporal Δt fiind dat de expresia $\Delta t = 2(\rho_1 - \rho_2)/c$. Din considerente geometrice ρ_1 și ρ_2 pot fi scrise astfel:

$$\rho_1 = \sqrt{r^2 + x_n^2 - 2rx_n \cos\left(\frac{\pi}{2} - \theta\right)} \quad (4.13)$$

$$\rho_2 = \sqrt{(r + \Delta r)^2 + x_n^2 - 2(r + \Delta r)x_n \cos\left(\frac{\pi}{2} - \theta\right)} \quad (4.14)$$

unde x_n este distanța de la centrul ariei la elementul n și are expresia $x_n = (n - N/2 - 0.5)d$. Pentru simplificare se dezvoltă în serie binomială $(1+x)^k = 1+kx+k(k-1)x^2/2! + \dots$, dacă $|x| < 1$ și folosind aproximarea de paraxialitate se obțin următoarele expresii pentru ρ_1 și ρ_2 :

$$\rho_1 \approx r + \frac{x_n^2}{2r} - x_n \sin \theta \quad (4.15)$$

$$\rho_2 \approx r + \Delta r + \frac{x_n^2}{2(r + \Delta r)} - x_n \sin \theta \quad (4.16)$$

Prin urmare diferența de timp Δt se exprimă astfel:

$$\Delta t = \frac{2}{c} \left[\Delta r - \frac{x_n^2}{2} \frac{\Delta r}{r(r + \Delta r)} \right] \quad (4.17)$$

Dacă se impune condiția $\Delta t = 0$ rezultă $r(r + \Delta r) = \frac{x_n^2}{2}$. Pentru formatoarele de fascicule

receptoare bazate pe CAN cu supraeșantionare, Δr se alege suficient de redus în comparație cu r pentru a asigura raportul semnal zgomot dorit și prin urmare este rezonabilă omiterea sa din

ultima ecuație. Ca urmare se obține $\Delta t = 0$ pentru $r \approx \frac{x_n}{\sqrt{2}}$, ceea ce înseamnă că pentru

punctele focale consecutive situate la $r \approx \frac{x_n}{\sqrt{2}}$ și $r + \Delta r \approx \frac{x_n}{\sqrt{2}} + \Delta r$ ecourile reflectate pot sosi

exact în același timp la elementul n al ariei. Altfel spus, din considerente geometrice aceste două puncte focale reclamă folosirea aceluiași eșantion de pe canalul n în procesul de formare a fasciculului prin focalizare dinamică. Din fericire folosirea ponderării de tip FN (f number) pentru îmbunătățirea calității imaginii face imposibilă îndeplinirea condiției de mai sus. Ponderarea FN se exprimă astfel: $FN \leq r/D$ unde FN este o constantă întreagă predeterminată ($FN = 1, 2, 3, \dots$) iar D reprezintă lungimea ariei. Este evident din această relație că pentru punctele focale apropiate de arie nu toată aria ci doar o porțiune din ea este autorizată să participe la formarea fasciculului receptor, iar în exteriorul unui domeniu dat toate elementele ariei sunt activate la recepție. Cu alte cuvinte domeniul punctului focal receptor este întotdeauna mai mare decât dimensiunea ariei fiind legat de aceasta printr-un factor constant

($r \geq FN \cdot D$), și prin urmare condiția $r \approx \frac{x_n}{\sqrt{2}}$ care ar impune folosirea aceluiași eșantion pentru

mai multe puncte focale nu este niciodată îndeplinită.

Un alt aspect al problemei repetiției eșantioanelor este legat de cuantificarea temporală. Repetiția eșantioanelor datorată cuantificării temporale poate fi eliminată dacă se alege o rezoluție temporală T_m mai mică decât întârzierea minimă $(\Delta t)_{\min}$: $T_m \leq (\Delta t)_{\min}$. Valoarea $(\Delta t)_{\min}$ se obține din minimizarea relației (4.17) funcție de parametrii n și r :

$$(\Delta t)_{\min} = \frac{2}{c} \left[\Delta r - \frac{(x_n)_{\max}^2}{2} \frac{\Delta r}{r_0(r_0 + \Delta r)} \right] \quad (4.18)$$

unde $(x_n)_{\max}$ este distanța de la cel mai îndepărtat element la centrul ariei, iar r_0 primul punct focal dorit. Dacă se substituie $\Delta r = c/(2f_{BF})$, f_{BF} – frecvența de lucru a formatorului, în ecuația (4.18) se obține:

$$(\Delta t)_{\min} = \frac{1}{f_{BF}} \left[1 - \frac{(x_n)_{\max}^2}{2} \frac{1}{r_0(r_0 + \Delta r)} \right] \quad (4.19)$$

Prin urmare frecvența minimă de tact, $f_m = 1/T_m$, care rezolvă problema repetiției eșantioanelor în procesul de formare a fasciculului, poate fi exprimată astfel:

$$\frac{f_m}{f_{BF}} \geq \frac{1}{1 - \frac{(x_n)_{\max}^2}{2} \frac{1}{r_0(r_0 + \Delta r)}} \quad (4.20)$$

Figura 4.19 indică dependența raportului dintre frecvența de tact și cea de formare funcție de poziția primului punct focal (exprimată funcție de dimensiunea D a ariei) pentru diverse valori ale FN.

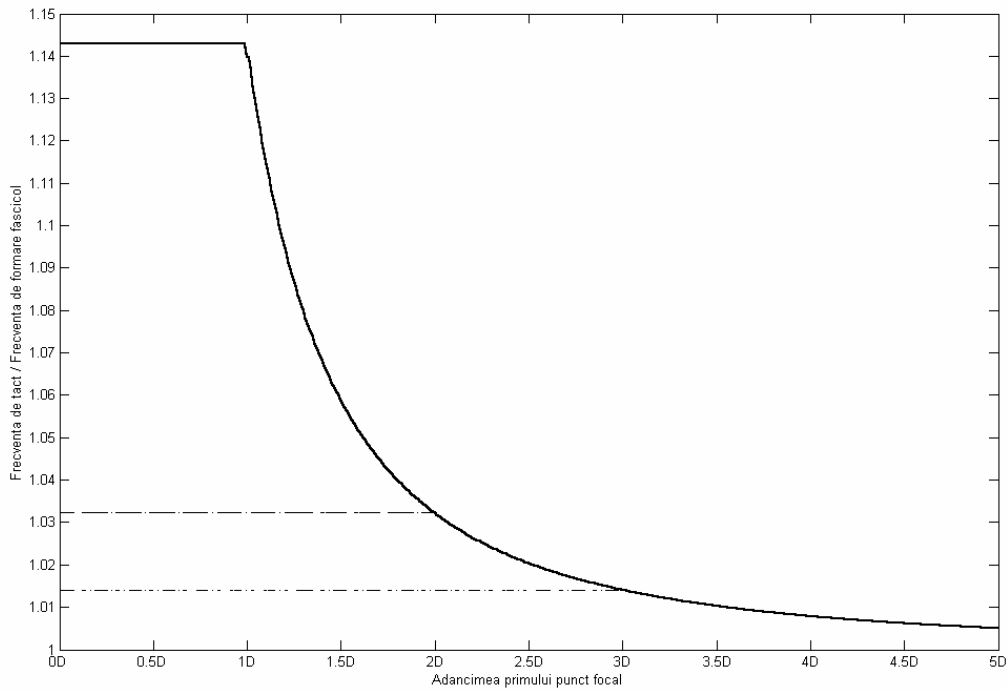


Fig. 4.19 Dependența raportului frecvență de tact / frecvență de formare de poziția primului punct focal

Se observă din figură că în situația cea mai defavorabilă, care se obține când adâncimea primului punct focal este mai mică decât lungimea ariei multiplicată cu FN ($r_0 \leq FN \cdot D$), cele două frecvențe sunt într-un raport constant iar pentru valori mai mari ale lui r_0 constrângerea asupra lui f_m se reduce. În mod clar pentru un FN fixat există o cea mai defavorabilă situație în care raportul frecvențelor f_m și f_{BF} depinde numai de FN pentru orice alegere a lui r_0 . Valoarea acestui raport se poate deduce substituind în ecuația (##) $(x_n)_{\max}$ cu $r_0 / 2FN$, deoarece $-D/2 \leq x_n \leq D/2$, și neglijând pe Δr comparativ cu r_0 . Rezultă o valoare independentă de r_0 :

$$\frac{f_m}{f_{BF}} \geq \frac{8(FN)^2}{8(FN)^2 - 1} \quad (4.21)$$

În concluzie, prin alegerea unei frecvențe de tact f_m de cel puțin $\frac{8(FN)^2}{8(FN)^2 - 1}$ ori mai mare decât frecvența de formare f_{BF} se evită repetarea eșantioanelor în procesul de formare a fasciculului la recepție pentru ponderarea cu constanta FN.

4.7 DISCUȚIE

Pentru evaluarea cantitativă a celor trei variante de formare a fasciculului s-a recurs la simularea cu date reale în condițiile descrise în prima parte a capitolului. S-a folosit pentru cuantificarea performanțelor reprezentarea în domeniul timp și analiza spectrală la nivel de fascicul. Figurile 4.20 – 4.23 prezintă spectrul semnalului de ieșire pentru același fascicul generat cu patru structuri de formare. Prima figură arată spectrul fasciculului referință sintetizat de un formator convențional cu focalizare fixă la transmisie / focalizare dinamică la recepție. Următoarele două caracteristici spectrale au fost obținute cu formatorul cu modulație delta-sigma uniformă în configurația FFTr/FFRec și respectiv FFTr/FDRec. Nivelul zgomotului de joasă frecvență mai mare cu 20dB la focalizarea dinamică confirmă ipoteza că acesta se datorează desincronizărilor modulator/demodulator la modificarea dinamică a întârzierilor.

Pentru formatorul cu modulație delta-sigma neuniformă cu FFTr/FDRec, care respectă condiția de evitare a repetiției eșantioanelor, spectrul semnalului de ieșire indică un nivel al zgomotului de joasă frecvență comparabil cu cel rezultat pentru FFTr/FFRec și doar cu 10 dB peste nivelul indicat de caracteristica de referință. Pentru modulatorul sigma delta spectrul a fost evaluat înaintea procedurii de filtrare, fapt ce explică prezenta zgomotului de frecvență ridicată ($> 5\text{MHz}$). Reprezentările în domeniul timp din figurile 4.24 – 4.26 sunt date pentru semnale de ieșire filtrate și decimate. Reducerea de 15dB a nivelului de zgomot pentru formatorul cu modulație neuniformă comparativ cu cel cu modulație uniformă necompensat se datorează păstrării sincronismului modulator – demodulator și evitării repetiției eșantioanelor la formatorul cu eșantionare neuniformă.

Alegerea soluției constructive pune în discuție la performanțe similare complexitatea hardware a fiecărei variante. Din acest punct de vedere soluția cu modulație neuniformă este mai simplă deoarece nu presupune modificarea modulatorului ci doar asigurarea raportului de frecvențe pentru a se evita repetiția. În ceea ce privește generarea și controlul întârzierilor suportul de memorie necesar este practic același în cele două situații. Reducerea în continuare a complexității structurii hardware are în vedere tocmai înlocuirea blocului de memorie folosit pentru stocarea întârzierilor cu un mecanism de generare „on line” a secvențelor de întârziere

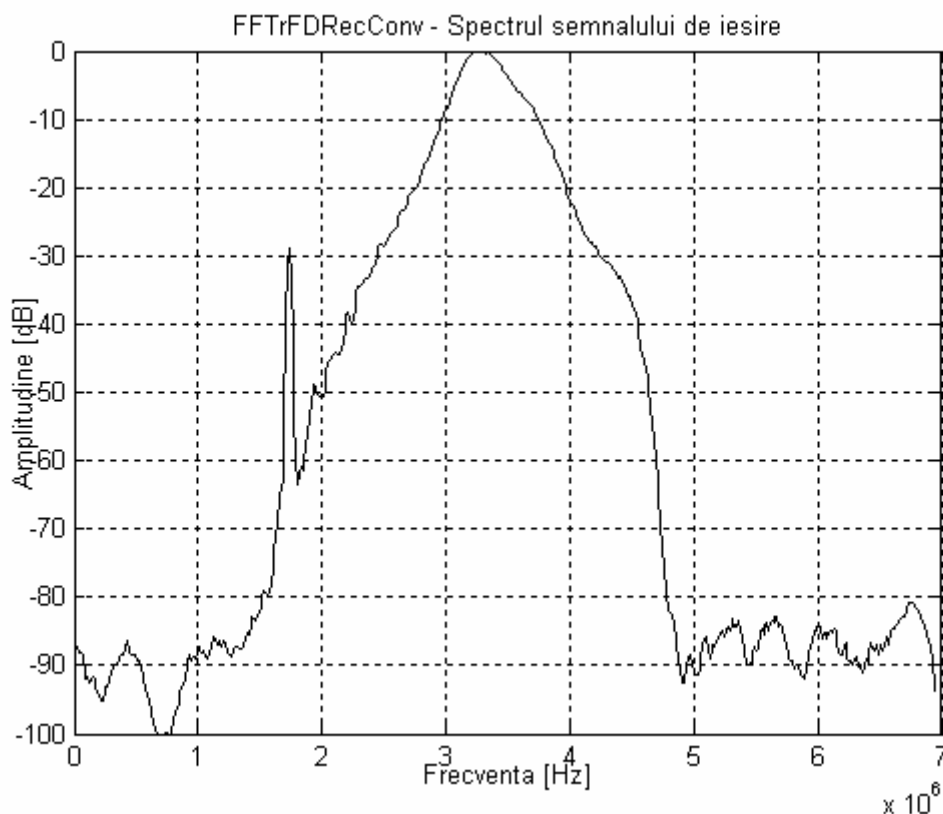


Fig. 4.20 Formator convențional FFTr/FDRec – Spectrul unui fascicul

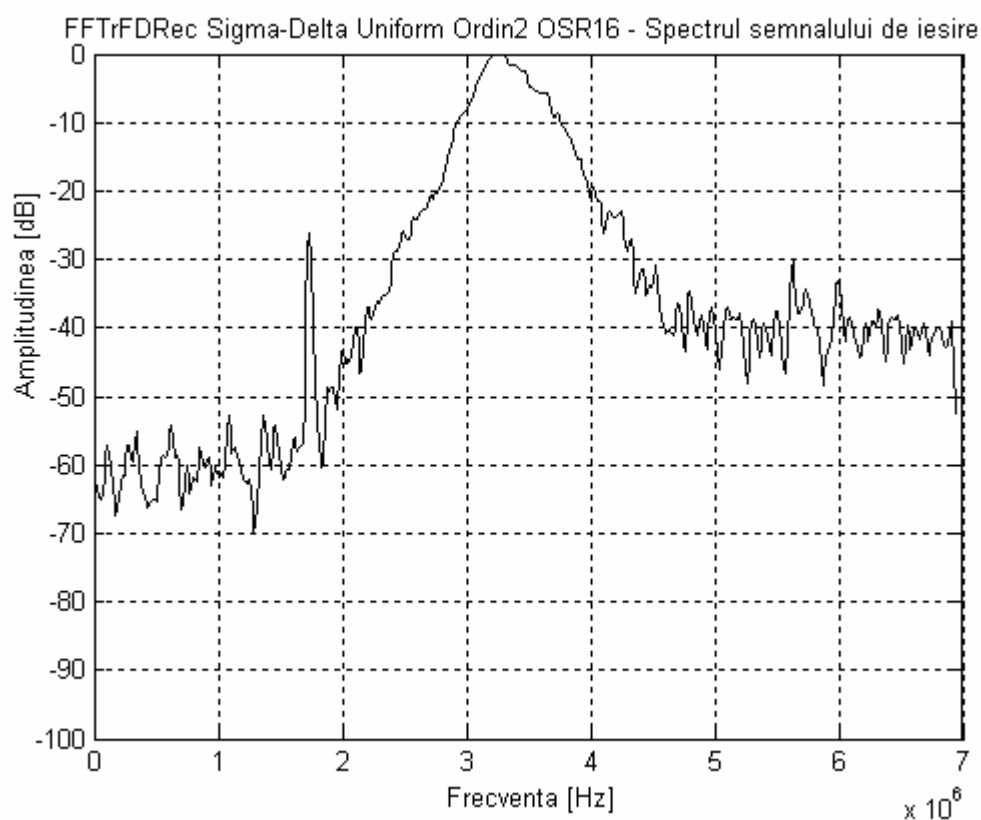


Fig. 4.21 Formator cu modulație sigma –delta uniformă FFTr/FDRec – Spectrul unui fascicul

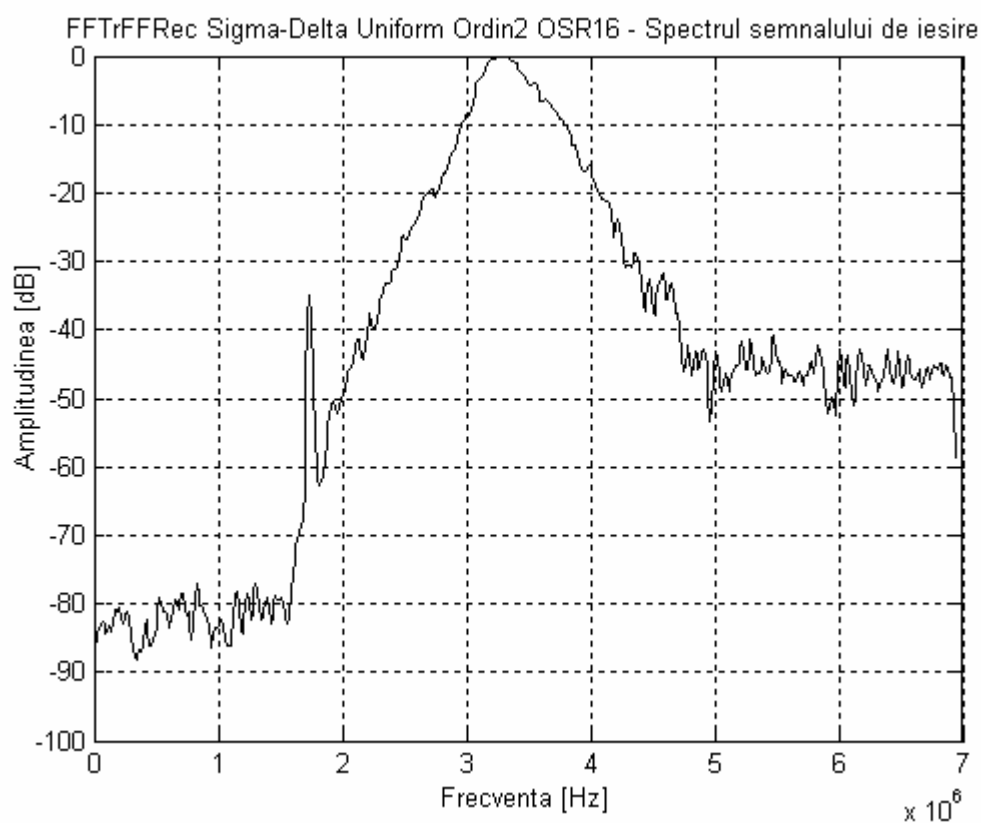


Fig. 4.22 Formator cu modulație sigma –delta uniformă FFTr/FFRec – Spectrul unui fascicul

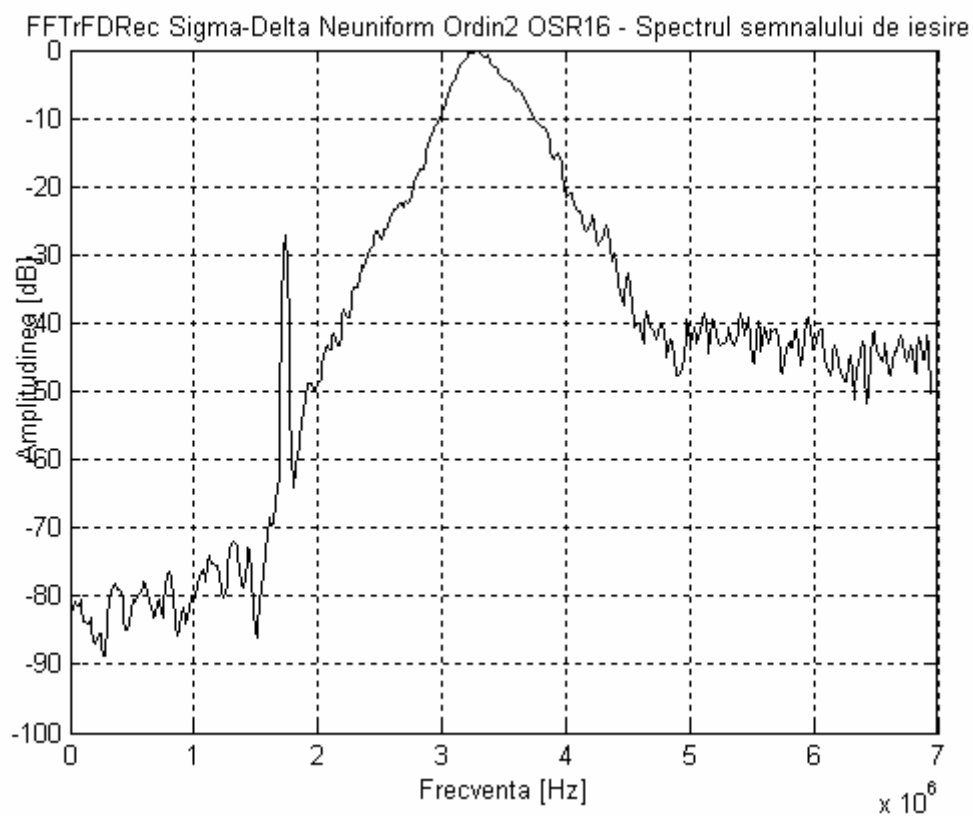


Fig. 4.23 Formator cu modulație sigma –delta neuniformă FFTr/FDRec – Spectrul unui fascicul

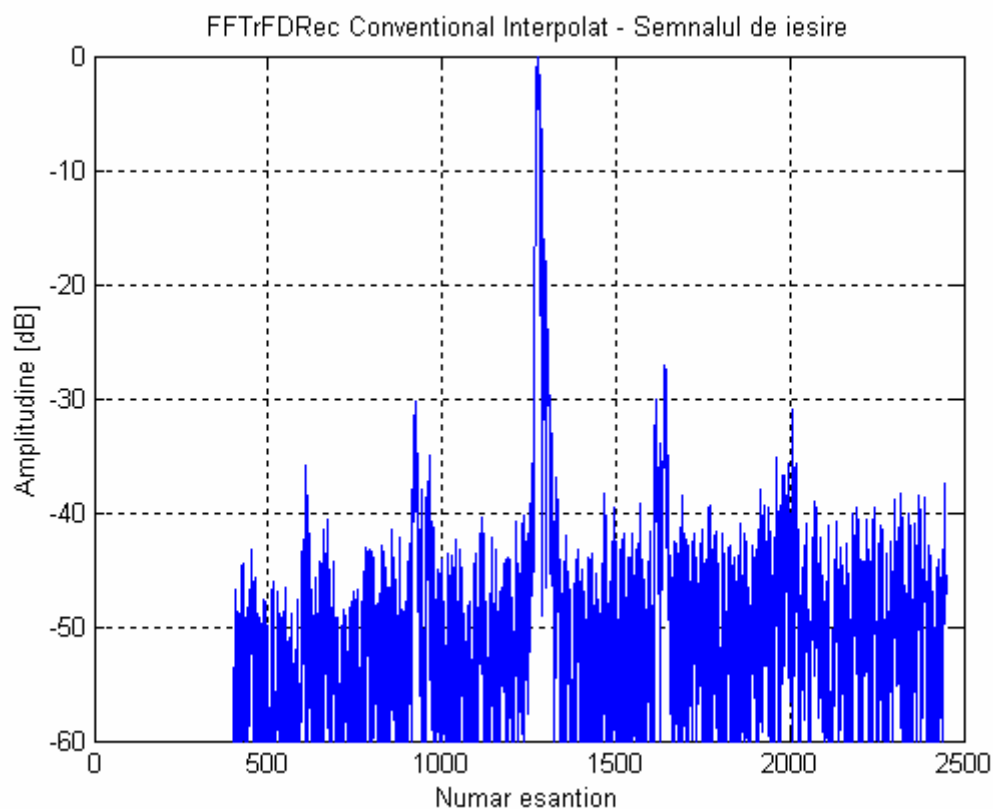


Fig. 4.24 Formator convențional FFTr/FDRec – Reprezentarea unui fascicul în domeniul timp

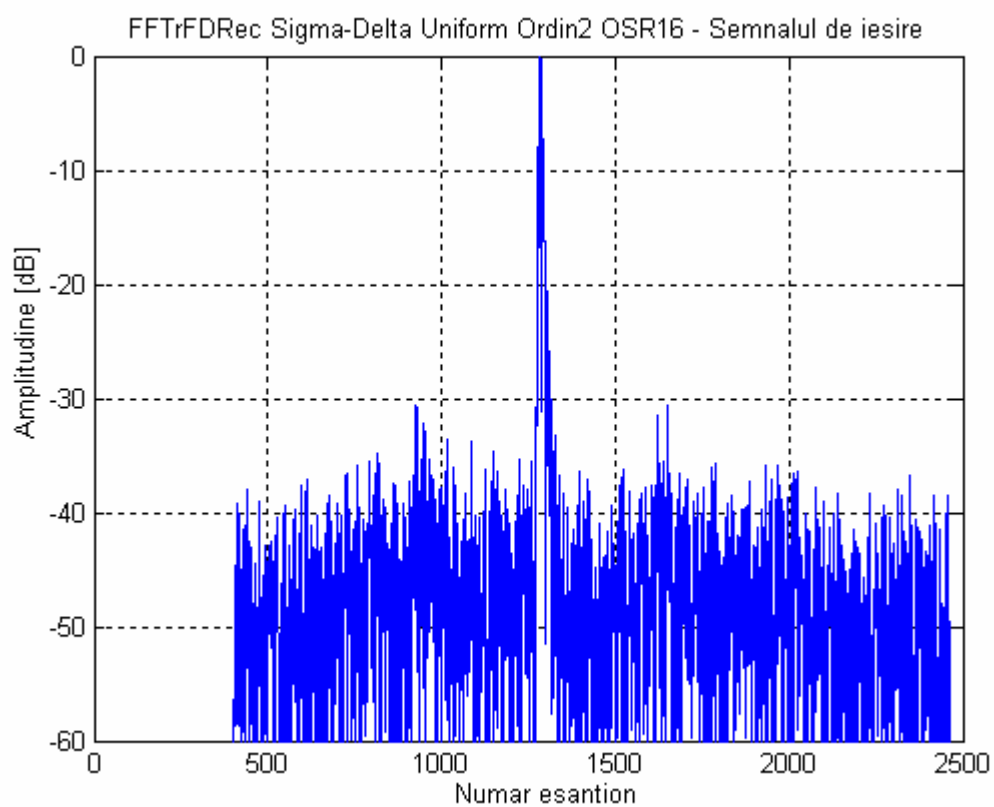


Fig. 4.25 Formator cu modulație sigma –delta uniformă FFTr/FDRec – Reprezentarea unui fascicul în domeniul timp

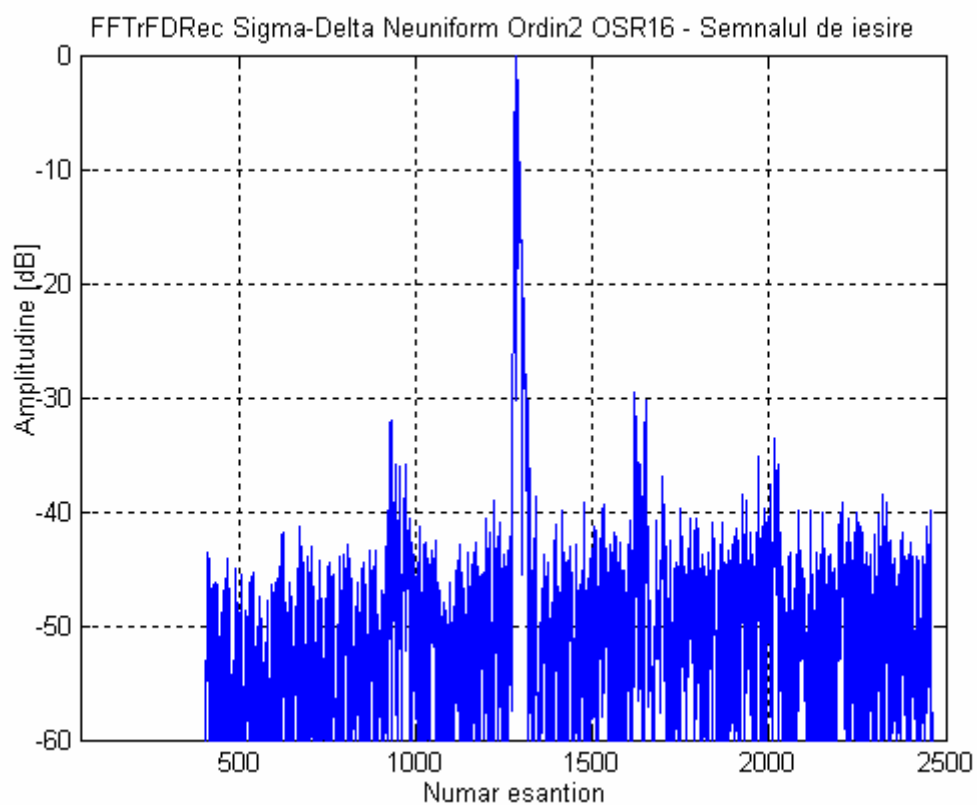


Fig. 4.26 Formator cu modulație sigma –delta neuniformă, FFTr/FDRec – Reprezentarea unui fascicul în domeniul timp

4.8 CONCLUZII

În cadrul capitolului a fost abordată problematica formatoarelor de fascicul de tip întârziere-însurare care au la bază modulația delta-sigma. Au fost supuse analizei comparative variantele de realizare cu eșantionare uniformă și respectiv neuniformă. Având ca referință formatorul convențional multibit s-a urmărit alegerea unei soluții constructive care să asigure performanțe comparative cu cele ale formatorului clasic în condițiile simplificării structurii hardware. Analiza s-a bazat pe modelarea Matlab cu date de intrare reale a celor trei variante constructive: clasic, modulație delta-sigma uniformă și modulație delta-sigma neuniformă, și cuantificarea performanțelor fiecărui tip de formator.

Evaluarea performanțelor fiecărui tip de formator a necesitat într-o primă fază precizarea locului formatorului de fascicul în structura sistemului imagistic. Pe baza analizei structurale a sistemului imagistic ultrasonic a fost posibilă construcția unui model Matlab pentru sistem și simularea cu date reale a structurii hardware.

Structura formatorului digital este definită prin tipul de focalizare și modalitatea de prelevare a eșantioanelor din semnalele ecou furnizate de canalele de recepție. **Eșantionarea** este prima etapă din procesul de generare a fasciculului care mai cuprinde pentru formatoarele convenționale: **conversia analog numerică, întârzierea și însumarea.**

- La **eșantionarea uniformă** etapele sunt parcurse în ordinea enumerată mai sus fiind implementate fiecare pe suportul hard necesar. Expandarea structurii formatorului spre o configurație multifascicul se realizează prin multiplicarea resurselor pentru întârziere și însumare, procesul de eșantionare și conversie fiind comun pentru toate fasciculele generate simultan.
- la **eșantionarea neuniformă** eșantionarea și întârzierea se desfășoară simultan, fiind urmate de conversie și însumare. Contopirea celor două etape conduce la reducerea complexității suportului hardware. Expandarea structurii formatorului spre o configurație multifascicul necesită multiplicarea întregului canal de formare inclusiv a convertoarelor analog numerice.

Alegerea **tipului de focalizare** se face funcție de cerințele de rezoluție impuse sistemului de investigare și se traduce la final în constrângeri asupra resurselor hardware și software. Soluția utilizată pentru sistemele care operează în timp cvasi-real este folosirea focalizării fixe la transmisie și a focalizării dinamice la recepție, variantă în care pentru fiecare cadru de imagine se realizează un număr de transmisii egal cu numărul de fascicule, și pe fiecare perioadă de recepție se formează un fascicul. Realizarea unui sistem care să opereze efectiv în timp real necesită conectarea în paralel a unui număr de structuri de recepție și formare de fascicul egal cu numărul de direcții unghiulare care definesc sectorul explorat. Evident o astfel de soluție nu este practică dar folosirea de formatoare multifascicul poate, în anumite cazuri, să asigure un raport optim între complexitatea formatorului și frecvența cadrelor.

Generarea fasciculelor de ultrasunete prin folosirea modulației sigma – delta se justifică prin necesitatea simplificării structurii hardware a canalului receptor în condițiile folosirii de arii cu un număr tot mai mare de elemente (128; 256) pentru a obține imagini de calitate superioară. Această tehnică presupune folosirea convertoarelor cu supraeșantionare în componența blocului digital de intrare și scheme simple de procesare pentru întârzierea și însumarea coerentă.

Folosind tehnici de supraeșantionare pot fi obținute imagini cu o calitate similară cu a celor obținute cu tehnicile digitale multi-bit de formare a fasciculelor, în condițiile reducerii semnificative a complexității și gabaritului sistemului. Formatoarele ce au la bază convertoare $\Delta\Sigma$ prezintă două avantaje majore față de metodele tradiționale multi bit.

- Simplificarea structurii CAN și posibilitatea integrării pe același chip împreună cu blocurile de procesare digitală care creează fasciculul. Circuitele necesare pentru a

implementa CAN $\Delta\Sigma$ sunt mult mai simple decât cele folosite în CAN flash multi-bit (câteva amplificatoare operaționale și un comparator față de 255 comparatoare necesare pentru un CAN flash pe 8 biți), fapt ce se regăsește în reducerea resurselor de interconectare, a gabaritului și puterii consumate.

- Asigurarea de întârzieri precise prin manipularea eșantioanelor prelevate cu o rată de eșantionare ridicată și simplificarea blocului sumator, primul nivel de însumare fiind alcătuit din sumatoare pe un bit.

Domeniul dinamic larg necesar pentru aplicațiile ultrasonore, necesită un CAN $\Delta\Sigma$ de ordinul doi sau trei pentru a menține un raport semnal zgomot adecvat. Pentru a asigura atât un raport semnal zgomot ridicat cât și o acuratețe adecvată a întârzierilor este necesar un factor de supraeșantionare $OSR \geq 16$. Cuantificatorul de la ieșirea modulatorului se alege în mod obișnuit cu două niveluri de cuantizare datorită simplității sale. Adăugarea de niveluri de cuantificare îmbunătățește raportul semnal zgomot al sistemului dar complică circuitul de reacție, în special CNA – ul din componența modulatorului.

Calea de semnal a unui sistem imagistic ultrasonic conține subsistemele de prelucrare a informației de la conversia semnalelor acustice în semnale electrice la nivelul ariei de traductoare și până la obținerea unei imagini într-un format accesibil utilizatorului.

Din punct de vedere al tipului informației prelucrate există un **nivel de procesare analogică** care include etajul preamplificator, amplificatorul cu câștig variabil și blocul de conversie analog numerică și un **nivel de procesare digitală** având ca blocuri principale formatorul de fascicul și convertorul de coordonate. Blocurile care fac obiectul lucrării de față: formatorul de fascicul și convertorul delta-sigma au fost testate soft folosind setul de date în format numeric rezultate în urma conversiei analog numerice multibit, un program care implementează algoritmul de formare a fascicului ultrasonic și o rutină care descrie software conversia de la coordonatele polare la cele carteziane.

Pentru simularea formatoarelor cu modulație delta-sigma au fost incluse în structura programului de simulare pentru formatorul clasic faze suplimentare pentru: extinderea cu zerouri + interpolare, conversia delta-sigma, filtrarea trece jos și decimarea șirului de date rezultat. Exceptând conversia delta-sigma pentru care s-au folosit rutinele dedicate acestui scop din ToolBox pentru toate fazele enumerate mai sus au fost concepute module originale în Matlab.

Imaginile în coordonate carteziane obținute ca rezultat al simulării au permis o apreciere calitativă asupra procesului de formare a fasciculului. Pentru a evalua cantitativ performanțele formatoarelor s-a folosit analiza spectrală la nivel de fascicul și reprezentarea fasciculului în domeniul timp la scară logaritmică.

Evaluarea comparativă a versiunilor unei imagini generate cu un formator conventional multibit și respectiv cu un formator Delta-Sigma cu eșantionare uniformă a indicat prezența „neșteptată” a unui zgomot suplimentar în imaginea generată cu formatorul Delta-sigma. Cauza apariției acestui zgomot a fost detectată a fi modificarea dinamică a întârzierilor în timpul procesului de formare de fascicul.

Convertorul analog numeric CAN $\Delta\Sigma$ împreună cu filtru de reconstrucție trece jos constituie o pereche modulator – demodulator între care pot fi inserate blocuri care execută operații liniare fără ca aceasta să conducă la degradarea raportului semnal zgomot.

Procesul de formare al fasciculului ultrasonic prin algoritmul întârziere – însumare a fost descompus în două etape distincte :

- întârzierea – decalarea temporală a informației recepționate
- însumarea semnalelor decalate

Focalizarea undelor acustice în câmp apropiat necesită întârzierea după o lege parabolică, adică formarea fasciculelor ultrasonice se realizează printr-o operație neliniară de întârziere urmată de operație liniară de însumare.

Folosirea convertoarelor delta-sigma cu modulație uniformă pentru formarea fasciculelor constă în intercalarea între modulator și demodulator a procesului neliniar de întârziere urmat de procesul liniar de însumare. Neliniaritatea întârzierii conduce la reeșantionarea semnalelor livrate de elementele de arie care se traduce în folosirea repetată a unor eșantioane sau nefolosirea altora. Repetarea sau eliminarea de eșantioane din șirul rezultat în urma conversiei delta-sigma cu rată uniformă duce la desincronizarea ansamblului modulator / demodulator. Preponderența unuia sau a altuia dintre mecanismele de desincronizare este funcție de raportul între frecvența de eșantionare (f_M) și frecvența de formare (f_{BF}).

Au fost evaluate din punct de vedere al complexității soluțiile de sincronizare a ansamblului modulator-demodulator. Extinderea șirului de eșantioane la nivelul formatorului și implicit creșterea rezoluției întârzierilor se realizează prin una din următoarele metode:

- repetarea unui eșantion;
- inserarea unui eșantion nul;
- descompunerea (scindarea) eșantionului în două părți egale

O primă soluție de sincronizare forțează modulatorul să țină seama în funcționare de eșantionul repetat sau eliminat asigurând prezența acestuia în șirul de eșantioane ce rezultă în urma modulării. Pentru aceasta se include în bucla de reacție un multiplexor care alege între amplitudine zero, amplitudine normală sau amplitudine multiplicată cu 2. Se obține astfel un nou tip de modulator, care include multiplexorul în reacție, denumit modulator delta-sigma compensat.

Pentru situațiile în care se descompune eșantionul sau se introduce un eșantion nul nu este necesară intervenția asupra modulatorului pentru a compensa șirul de eșantioane. Aceste metode nu afectează puterea medie a semnalului și ca urmare modulatorul și filtrul de reconstrucție rămân sincronizate. Prin contrast, repetiția unui eșantion conduce la existența a două eșantioane de aceeași mărime și ca urmare modulatorul trebuie compensat.

Metodele de menținere a sincronismului prin „inserarea unui eșantion nul” sau prin „descompunerea eșantionului în două părți egale” sunt mai simple decât compensarea directă deoarece necesită doar o simplă recodare a eșantioanelor la ieșire pentru a modifica șirul eșantioanelor întârziate. Deoarece, la performanțe egale este mai simplă, necesitând un singur bit suplimentar pentru recodarea eșantioanelor la ieșire, metoda de introducere de eșantioane nule este preferată pentru modificarea dinamică a întârzierilor la formatoarele bazate pe modulația delta-sigma cu eșantionare uniformă.

Toate metodele de păstrare a sincronizării modulator – demodulator indicate presupun intervenția asupra structurii hardware a modulatorului și circuite de control suplimentare pentru detecția desincronizărilor.

Formarea de fascicul prin conversie sigma – delta cu rată neuniformă constă în extragerea procesului neliniar de întârziere din ansamblul modulator / demodulator și plasarea acestuia înaintea modulatorului. În acest mod se asigură sincronizarea perechii modulator / demodulator întrucât între acestea este intercalată doar operația liniară de însumare.

Formatoarele digitale care se bazează pe eșantionarea neuniformă realizează focalizarea dinamică prin eșantionarea semnalelor ecou la momente de timp care asigură condiția de însumare coerentă la recepție. Tactul pentru eșantionarea neuniformă este generat pe seama setului de întârzieri stocat într-o memorie digitală operată cu un tact uniform f_M . Din considerente geometrice s-a dedus relația ce trebuie îndeplinită de frecvența de generare a întârzierilor (f_M) și frecvența de formare (f_{BF}) pentru a evita repetiția eșantioanelor în procesul de generare a fasciculului

În evaluarea cantitativă a celor trei variante de formare a fasciculului s-a recurs la simularea lor cu date reale. S-a folosit pentru cuantificarea performanțelor reprezentarea în domeniul timp și analiza spectrală la nivel de fascicul. Nivelul zgomotului de joasă frecvență mai mare cu 20dB la focalizarea dinamică confirmă ipoteza că acesta se datorează desincronizărilor modulator/demodulator la modificarea dinamică a întârzierilor.

Pentru formatorul cu modulație delta-sigma neuniformă cu FFTr/FDRec, care respectă condiția de evitare a repetiției eșantioanelor, spectrul semnalului de ieșire indică un nivel al zgomotului de joasă frecvență comparabil cu cel rezultat pentru FFTr/FFRec și doar cu 10 dB peste nivelul indicat de caracteristica de referință. Reprezentările în domeniul timp date pentru semnale de ieșire filtrate și decimate arată reducerea de 15dB a nivelului de zgomot pentru formatorul cu modulație neuniformă comparativ cu cel cu modulație uniformă necompensat datorită păstrării sincronismului modulator – demodulator și evitării repetiției eșantioanelor la formatorul cu eșantionare neuniformă.

Alegerea soluției constructive pune în discuție la performanțe similare complexitatea hardware a fiecărei variante. Din acest punct de vedere soluția cu modulație neuniformă este mai simplă deoarece nu presupune modificarea modulatorului ci doar asigurarea raportului de frecvențe pentru a se evita repetiția. În ceea ce privește generarea și controlul întârzierilor suportul de memorie necesar este practic același în cele două situații. Reducerea în continuare a complexității structurii hardware are în vedere tocmai înlocuirea blocului de memorie folosit pentru stocarea întârzierilor cu un mecanism de generare „on line” a secvențelor de întârziere

IMPLEMENTAREA FORMATORULUI CU MODULAȚIE DELTA - SIGMA NEUNIFORMĂ

5.1. STRUCTURA SECȚIUNII DIGITALE

Formatoarele digitale care se bazează pe eșantionarea neuniformă realizează focalizarea dinamică prin eșantionarea semnalelor ecou la momente de timp care asigură condiția de însumare coerentă la recepție. Tactul pentru eșantionarea neuniformă este generat pe seama setului de întârzieri stocat într-o memorie digitală operată cu un tact uniform f_M . Generatorul tactului de eșantionare produce semnale neuniforme de tact care sunt diferite pentru fiecare element al ariei.

Eșantioanele întârziate dinamic sunt apoi digitizate de un set de modulatatoare delta-sigma pe un bit, înainte de însumarea coerentă, fiecare modulator fiind activat cu tactul neuniform produs de generator – figura 5.1. Ieșirea fiecărui modulator este un șir de date pe un bit care include semnalul original întârziat plus zgomotul de cuantificare de înaltă frecvență. Aceste reprezentări pe un bit ale eșantioanelor ecou apar la diverse momente de timp pe canalele de recepție deoarece nu există o regularitate la sosirea ecourilor.

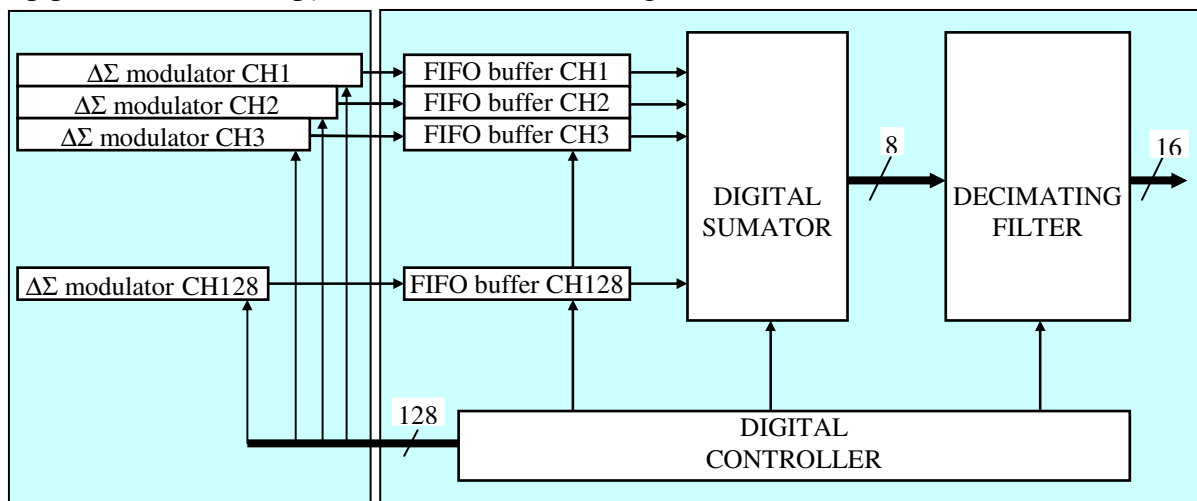


Fig. 5.1 Structura formatorului cu modulație delta-sigma neuniformă

Eșantioanele codate pe un bit care apar în diverse momente de timp la ieșirile modulatatoarelor sunt aliniate prin bufferele FIFO de pe fiecare canal și vor fi transmise simultan sumatorului. Acesta generează versiunea grosieră a fasciculului care conține suma ecourilor întârziate plus zgomotul de cuantificare. Fasciculul este procesat în continuare de filtrul de reconstrucție cu scopul de a elimina zgomotul de cuantificare și a reduce rata de eșantionare la o valoare apropiată de rata Nyquist.

Din punct de vedere structural formatorul digital de fascicul cu modulație sigma delta neuniformă se compune din două secțiuni:

- Secțiunea analogică constituită din modulatările delta sigma
- Secțiunea digitală alcătuită din următoarele entități:
 - controllerul digital
 - memoria FIFO
 - sumatorul

- filtrul decimator

Entitatea centrală este controlerul digital care asigură semnalele de comandă și coordonare pentru toate celelalte componente ale formatorului. Controlerul digital îndeplinește următoarele funcții:

- generează tactul de eșantionare neuniformă pentru comanda modulatorilor delta-sigma și încărcarea bufferelor FIFO.
- asigură semnalele de comandă pentru descărcarea FIFO, pentru blocul sumator și pentru filtrul decimator.

5.1.1 IMPLEMENTARE FPGA SAU ASIC

Alegerea tehnologiei de implementare a secțiunii digitale necesită o analiză comparativă privind avantajele și dezavantajele tehnologiilor de implementare cu structuri logice programabile – FPGA, respectiv cu circuite integrate dedicate - ASIC. Tabelul de mai jos prezintă comparativ aspectele cheie pentru producția de prototip și serie mică.

Tabelul 5.1

Tehnologia	Rutare	Arie	Putere	Cost	Programare
ASIC	Nelimitată	Nelimitată	Redusă	Ridicat	Singulară
FPGA	Limitată	Limitată	Ridicată	Scăzut	Nelimitată

Numărul de porți disponibile la proiectarea cu circuite programabile este limitat și depinde de dimensiunea dispozitivului fapt ce constituie un dezavantaj față de implementarea bazată pe ASIC-uri care oferă mai multă libertate din acest punct de vedere. De asemenea FPGA-urile au dezavantajul de a fi limitate la o structură particulară a elementului logic pe când în proiectele ASIC poate fi implementată orice poartă. Această limitare a circuitelor programabile structurate din elemente logice fixe cunoaște o modificare graduală așa cum se poate observa din paragraful 5.1.2 (referitor la structura dispozitivelor ALTERA). Arhitectura viitoarelor structuri logice programabile nu poate fi prezisă cu precizie dar fără îndoială acestea vor rămâne competitive.

Rutarea ridică anumite probleme pentru proiectele implementate în FPGA deoarece magistralele fixe de interconectare au capacități limitate. În plus interconexiunile cresc semnificativ consumul de putere la nivelul întregului dispozitiv. Dispozitivele ALTERA folosesc magistrale de interconectare care traversează întregul dispozitiv în timp ce alți producători de FPGA-uri cum este, de exemplu, Xilinx folosesc interconectarea pe segmente pentru a reduce consumul de putere și a crește resursele de rutare. Comparativ, pentru proiectele realizate în tehnologie ASIC volumul magistrelor de interconectare depinde exclusiv de proiect.

Consumul relativ ridicat de putere al dispozitivelor FPGA le face puțin atractive pentru proiectele care impun consumuri reduse. O măsură a puterii atât pentru FPGA-uri cât și pentru ASIC-uri este dată de proporționalitatea cu produsul între capacitatea dispozitivului, frecvența de lucru și tensiunea de alimentare. În condițiile în care cerința de sisteme de calcul rapide este în continuă creștere nu poate fi vorba de o reducere a consumului pe seama vitezei de operare. Pe de altă parte nu poate fi vorba nici de o reducere a capacității dispozitivelor întrucât se doresc implementări compacte pe un singur chip de proiecte tot mai complexe. Întră în discuție doar aspectul reducerii volumului de resurse de rutare prin folosirea interconectării pe segmente și scăderea tensiunii de alimentare la valori de 3,3 2,5 sau 1,8 Volți. Aceleași principii privind reducerea consumului se aplică și pentru dispozitivele ASIC dar este evident că FPGA-urile vor consuma mai mult datorită arhitecturii programabile.

FPGA-urile au însă două avantaje importante comparativ cu tehnologia ASIC:

- pot fi programate de mai multe ori (ASIC – ul se programează o singură dată)
- costul dispozitivelor folosite pentru implementare este mult mai redus.

În mod obișnuit FPGA-urile se programează la punerea sub tensiune dar există și variante reconfigurabile în circuit. Spre exemplu pentru formatorul de fascicul același FPGA poate fi programat într-o primă instanță pentru a implementa secvența de eșantionare neuniformă și stocarea datelor în memorie iar apoi poate fi reconfigurat pentru a realiza sinteza fasciculului prin însumarea și filtrarea datelor stocate anterior, lucru imposibil de realizat cu un ASIC. Costul dispozitivelor realizate în tehnologie ASIC este un factor prohibitiv în special în etapa de realizare a prototipurilor.

Un argument suplimentar pentru folosirea tehnologiei de realizare cu circuite programabile, în situația dată, îl constituie apariția pe piață a familiilor de dispozitive care conțin blocuri de memorie încorporate. Utilizarea acestor dispozitive crește șansele de realizare a unei implementări compacte unicip a secțiunii digitale a formatorului. Familiile FLEX, APEX și STRATIX realizate de firma ALTERA sunt opțiuni indicate pentru faza de prototip.

În concluzie, o metodologie adecvată de proiectare cuprinde realizarea prototipului folosind un FPGA și utilizarea tehnologiei ASIC ca soluție de implementare finală pentru producția de serie.

5.1.2 CARACTERISTICILE FAMILIEI DE CIRCUITE FPGA FLEX 10K.

Descriere Generală

Dispozitivele Altera FLEX 10K sunt primele PLD-uri industriale cu blocuri de memorie încorporate. Bazată pe elemente reconfigurabile de tip RAM CMOS, arhitectura Matricii Flexibile de Elemente Logice (FLEX – Flexible Logic Element MatriX) încorporează toate trăsăturile care sunt necesare matricilor de porți în vederea implementării megafuncțiilor. Cu peste 250000 de porți, familia FLEX 10K pune la dispoziție densitate, viteză și facilitățile necesare integrării de sisteme complete, inclusiv magistrale multiple de 32 de biți, într-un singur dispozitiv.

Dispozitivele FLEX 10K sunt reconfigurabile, ceea ce permite o testare completă înainte de livrare. Drept rezultat, utilizatorul nu trebuie să genereze vectori de test în scopul descoperirii de erori. În plus, proiectantul nu trebuie să fie preocupat de inventarierea diferitelor proiecte tip ASIC; dispozitivele FLEX 10K pot fi configurate direct pe placă pentru funcționalitatea specifică cerută.

Tabelul de mai jos arată performanțele FLEX 10K pentru câteva proiecte uzuale. Toate valorile performanțelor au fost obținute cu funcții parametrice de bibliotecă (LPM – Library Parametrized Module). Nu a fost necesară nici o tehnică specială de proiectare pentru a implementa aplicațiile; proiectantul nu a trebuit decât să instanțieze o funcție într-un fișier Verilog, VHDL, AHDL (Altera Hardware Description Language) sau într-un fișier în mod grafic.

Tabelul 5.2

Aplicația	Resurse folosite		Performanța Viteza				Unități
	LEs	EABs	-1	-2	-3	-4	
numărător setabil 16-biți	16	0	240	166	125	95	MHz
acumulator 16-biți	16	0	204	166	125	95	MHz
multiplexor 16-la-1	10	0	4,2	5,8	6.0	7.0	Ns
RAM 256 × 8 Viteza de Citire	0	1	172	145	108	84	MHz
RAM 256 × 8 Viteza de Scriere	0	1	106	86	68	63	MHz

Arhitectura FLEX 10K este similară celei a matricilor de porți încorporate, segmentul cel mai prolific al pieței matricilor de porți. Ca și matricile standard de porți, matricile încorporate de porți implementează logica generală într-o arhitectură convențională de tip “sea-of-gates” - mare de porți. În plus, matricile încorporate de porți, au zone dedicate pentru implementarea unor funcții specializate. Prin realizarea funcțiilor în siliciu, matricile încorporate de porți ocupă suprafețe reduse pe plachetă și asigură o viteză mai mare decât cea a matricilor standard de porți. Totuși, mega-funcțiile astfel realizate nu pot fi modificate după bunul plac, limitând opțiunile designerului. În contrast, dispozitivele FLEX 10K sunt programabile punând la dispoziția designerului un control total asupra mega-funcțiilor încorporate și asupra logicii generale facilitând în același timp modificări iterative de proiect în timpul validării acestuia.

Fiecare dispozitiv FLEX 10K conține o matrice încorporată și o matrice logică. Matricea încorporată este folosită pentru implementarea unei varietăți de funcții de memorie sau a unor funcții logice complexe, cum ar fi procesarea digitală de semnal (DSP), microcontrollere, manipularea unor magistrale extinse de date și funcții de conversie de date. Matricea logică execută aceleași funcții pe care le execută “sea-of-gates”(marea de porți) din matricea de porți: este folosită pentru implementarea de funcții logice generale, cum ar fi număratoarele, sumatoarele, automatele de stare și multiplexoarele. Combinația celor 2 tipuri de matrici conduce la performanțe înalte și la o densitate mare a matricii încorporate de porți, dând astfel posibilitatea utilizatorilor de a implementa un întreg sistem pe un singur dispozitiv.

Dispozitivele FLEX 10K sunt configurate la punerea sub tensiune a sistemului cu datele stocate într-un dispozitiv serial de configurare Altera sau puse la dispoziție de un controller de sistem. Datele de configurare pot fi descărcate și prin RAM-ul sistemului sau prin cablul serial de configurare Altera BitBlaster sau cablul de port paralel ByteBlaster, sau ByteBlasterMV. După configurarea dispozitivului FLEX 10K, el poate fi reconfigurat în circuit prin resetarea dispozitivului și încărcarea de noi date. Deoarece reconfigurarea necesită mai puțin de 320 ms, pot fi făcute schimbări în timp real în timpul operării sistemului.

Dispozitivele FLEX 10K conțin o interfață optimizată care permite microprocesoarelor să configureze dispozitivele FLEX 10K serial sau paralel și sincron sau asincron. Interfața permite de asemenea microprocesoarelor să trateze dispozitivul FLEX 10k drept memorie și să-l configureze prin scrierea într-o locație virtuală de memorie, proiectantului fiindu-i astfel mai ușor să reconfigureze dispozitivul.

Dispozitivele FLEX 10K sunt suportate de mediile de dezvoltare Quartus și MAX+PLUS II; un pachet integrat unitar care oferă facilități de introducere a proiectelor în format grafic, text (incluzând AHDL) sau formă de undă, compilare și sinteză logică, simulare completă, analiză de timp pentru cele mai defavorabile cazuri și configurare de dispozitiv. Mediile Quartus și MAX+PLUS II oferă interfețe EDIF200 și 300, LPM, VHDL, Verilog HDL și altele pentru introducere de proiecte adiționale și pentru suport de simulare de la alte unelte de proiectare electronică automată (EDA).

5.2. GENERATORUL TACTULUI DE EȘANTIONARE NEUNIFORMĂ

Pentru realizarea focalizării dinamice semnalele recepționate de elementele ariei de traductoare sunt amplificate și întârziate pe canale de procesare separate, iar apoi sunt combinate în formatorul receptor. Întârzierea pentru fiecare canal este selectată astfel încât fasciculul recepționat să fie dirijat și focalizat pe direcția cerută și la adâncimea dorită. Întârzierile trebuie modificate dinamic astfel ca focalizarea să se producă la adâncimi incrementate progresiv pe măsură ce energia ultrasonică este recepționată. Fasciculul transmis scanează o regiune din mediu iar semnalele generate de formator sunt procesate pentru a produce o imagine a regiunii scanate. O arie tipică de traductoare ultrasonice este structurată din minim 64 elemente. Prin urmare determinarea întârzierii necesare fiecărui element traductor spre a asigura focalizarea dinamică pentru fiecare punct dintr-un cadru imagine necesită efectuarea cu rapiditate a unui mare număr de calcule complexe în timp real sau un volum de stocare considerabil pentru memorarea întârzierilor calculate “off-line”.

5.2.1 NECESARUL DE MEMORIE PENTRU FORMATORUL DE FASCICUL

Formatorul digital de fascicul care realizează focalizarea dinamică la recepție necesită pentru fiecare canal la fiecare tact o valoare actualizată pentru întârziere. Într-o abordare clasică generarea tactului de eșantionare neuniformă se realizează prin citirea unei memorii care conține informația de întârziere într-un format specific. Pentru o adâncime de scanare $R=20\text{cm}$, o frecvență de eșantionare $f_s = 40\text{MHz}$ și o viteză de propagare $c = 1540\text{ms}^{-1}$ numărul de întârzieri este

$$N_{\Delta l} = \frac{2 \cdot R \cdot f_s}{c} = 10390$$

Această valoare multiplicată cu numărul de canale dă necesarul de memorie pentru formarea unei linii de imagine. Pentru a calcula necesarul de memorie pentru întreaga imagine rezultatul trebuie multiplicat cu numărul de linii care compun imaginea. Pentru un sistem de performanță modestă cu $N = 64$ elemente și $n_l = 90$ de linii per cadru imagine necesarul de memorie este:

$$N_{\Delta f} = N \cdot n_l \cdot \frac{2 \cdot R \cdot f_s}{c} = 59844155 \cong 60 \cdot 10^6$$

Exploatarea simetriei la formatoarele cu arii fazate permite reducerea la jumătate a volumului de memorie necesar dar chiar și în această situație o implementare compactă a secțiunii digitale a formatorului într-un singur cip este foarte dificilă. O soluție pentru reducerea necesarului de memorie constă în compresia informației pentru focalizare. În acest sens pentru generarea tactului de eșantionare neuniformă s-a investigat posibilitatea calculării iterative în timp real, “în circuit”, a informației de întârziere pe baza algoritmului punctului median aplicat unei ecuații care derivă din geometria de focalizare.

5.2.2 ALGORITMUL PUNCTULUI MEDIAN

Figura 5.2. prezintă geometria folosită pentru determinarea întârzierilor de focalizare pentru o arie fazată. Centrul ariei este localizat în originea sistemului de referință, punctul P indică punctul focal, r este distanța din punctul focal la centrul ariei, θ este unghiul de dirijare al fasciculului iar x descrie poziția unui element traductor față de centrul ariei. Într-un mediu fără atenuare întârzierea pentru focalizare a elementului x față de elementul de referință (din originea sistemului) este întârzierea relativă l/c , unde l este diferența de drum iar c viteza de propagare.

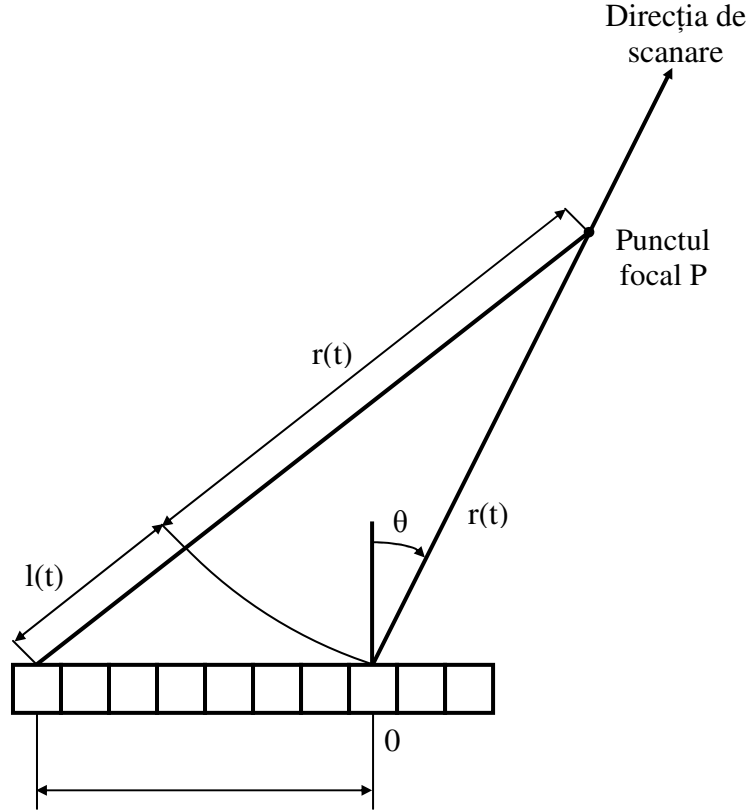


Fig. 5.2 Explicativă privind geometria focalizării

Diferența de drum se exprimă astfel:

$$l = (r^2 + x^2 + 2 \cdot r \cdot x \cdot \sin \beta)^{1/2} - r \quad (5.1)$$

unde distanța r este variabilă, unghiul θ este invariant pentru o direcție de scanare iar x este fix pentru un element dat. Calculul direct al întârzierii în timp real nu este practic deoarece implică operațiile consumatoare de timp de multiplicare și extragere a rădăcinii pătrate. Prin urmare este necesară aplicarea de metode numerice pentru a simplifica evaluarea ecuației (2.1). O astfel de metodă este algoritmul punctului median “midpoint algorithm” descris în [VanAken84]. Pe scurt utilizarea algoritmului pentru calculul întârzierii este explicat cu referire la curba l - r din figura 5.3. Variabila l reprezintă valoarea estimată cuantificată a întârzierii relative iar variabila indexată r descrie eșantionarea domeniului cu o perioadă $t_s = r/2v$. Aplicarea algoritmului presupune că ecuația curbei poate fi exprimată în forma $f(r,l) = 0$ și că satisface următoarele condiții:

- coeficienții ecuației $f(r,l) = 0$ sunt întregi

$$- r \text{ și } l \text{ satisfac ecuația } -1 \leq \frac{dl}{dr} \leq 0 \quad (5.2a)$$

$$\text{În cazul nostru } f(r,l) = l^2 + 2 \cdot l \cdot r - \alpha \cdot r - \beta, \text{ unde } \alpha = 2 \cdot x \cdot \sin \theta, \beta = x^2. \quad (5.2b)$$

$$\text{Expresia derivatei este: } \frac{dl}{dr} = \frac{r}{\sqrt{r^2 + \alpha \cdot r + \beta}} - 1 \quad (5.2c)$$

Se observă că la limită, când r tinde la zero, valoarea derivatei devine egală cu -1. Figura 5.4 arată variația derivatei întârzierii de domeniu în situația cea mai defavorabilă pentru elementul cel mai îndepărtat de originea ariei (64) și pentru unghiul maxim de scanare

(45 grade). Îndeplinirea condiției (5.2a) asigură aproximarea curbei întârzierii prin decrementarea valorii curente a acesteia cu cel mult o unitate.

Algoritmul punctului median estimează valoarea întârzierii l funcție de domeniul r din ecuația întârzierii folosind adunări și scăderi întregi. Euristic algoritmul punctului median alege valoarea cuantificată de pe grid, i, cea mai apropiată de curba reală $l \rightarrow r$. Deoarece valorile cuantificate sunt spațiate uniform valoarea cea mai apropiată i este una situată la mai puțin de jumătate de nivel de cuantificare de curbă.

Matematic selecția valorii i se face pe baza semnelui funcției decizie d_n definită prin:

$$d_n = f(r_{n+1}, i_n - 1/2) \quad (5.3)$$

Funcția decizie este evaluată în perioada de tact n și este folosită pentru a determina valoarea lui i în perioada de tact n+1. Punctele marcate în figura 5.3 reprezintă întârzierile estimate prin algoritmul punctului median. În conformitate cu algoritmul valoarea întârzierii i pentru o valoare cuantificată dată a domeniului se alege egală cu valoarea sa anterioară sau cu această valoare decrementată cu o unitate de timp (corespunzătoare unei unități de distanță). Decrementarea se realizează în situația $d_n > 0$.

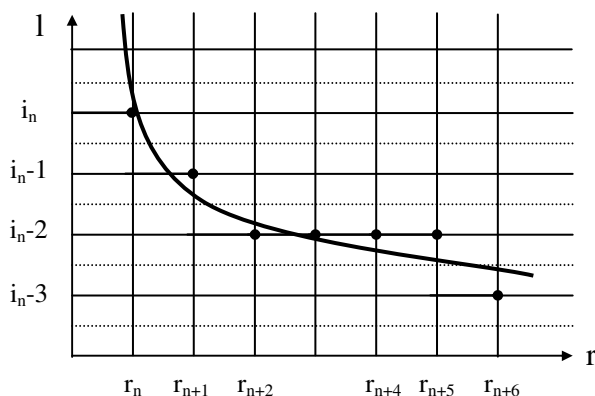


Fig.5.3 Explicativă privind algoritmul punctului median

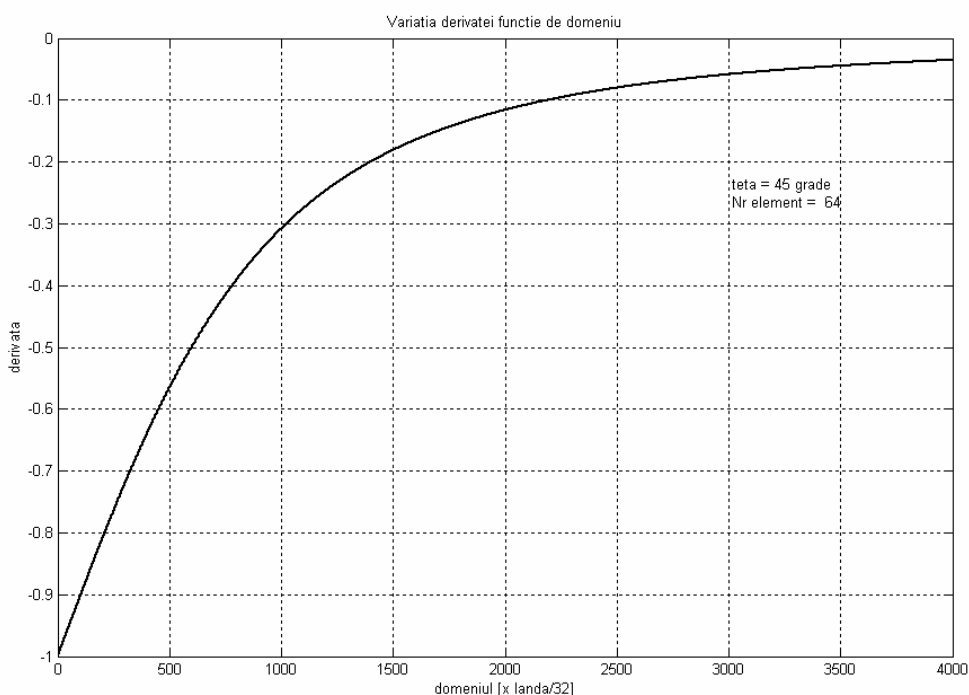


Fig. 5.4 Variația derivatei întârzierii funcție de domeniu

Algoritmul: Se notează cu i_n valoarea întreagă cea mai apropiată de valoarea reală a întârzierii l_n corespunzătoare punctului focal n situat la distanța r_n . Dacă r_n și i_n se dau ca și condiții inițiale și, în plus, este îndeplinită condiția $r_{n+1} = r_n + 1$, este posibil să se obțină o ecuație $i_{n+1} = i_n$ sau $i_{n+1} = i_n - 1$ datorită relației (5.2a). Astfel valoarea întreagă a întârzierii pentru al $n-1$ -lea punct focal se determină ca fiind i_n sau $i_n - 1$ funcție de valoarea variabilei decizie d_n dată de ecuația:

$$d_n = 4 \cdot f(r_n + 1, i_n - 0.5) = 4i_n^2 + 4i_n + 8i_n r_n - 4r_n(1 + \alpha) - 4\alpha - 4\beta - 3 \quad (5.4)$$

Variabila decizie rezultă prin multiplicarea cu 4 a ecuației (5.2b), pentru a lucra cu valori întregi și reprezintă valoarea ecuației pentru un punct median situat între punctele de coordonate (r_{n+1}, i_n) și $(r_{n+1}, i_n - 1)$. Așa cum rezultă și din figura 5.9 când variabila decizie este pozitivă deoarece întârzierea i_{n+1} este mai apropiată de valoarea $i_n - 1$ decât de valoarea i_n , întârzierea i_{n+1} se actualizează la valoarea $i_n - 1$. În caz contrar i_{n+1} se actualizează la valoarea i_n . Cazul $d_n = 0$ corespunde situației în care distanța între i_{n+1} și i_n sau $i_n - 1$ este aceeași și i_{n+1} va putea lua oricare dintre aceste două valori. În acest fel întârzierea i_n poate fi calculată pe seama domeniului focal r_n care se incrementează progresiv.

Calculul algoritmic al unei valori întregi a întârzierii pentru focalizare folosind relația (5.4) este evident mai eficient decât relația inițială dar include în continuare operații de multiplicare care necesită timp și resurse. Pentru a evita operațiile de multiplicare se ține seama de faptul că exprimarea incrementală a variabilei decizie conduce la următoarele relații:

$$\text{- pentru } d_n \leq 0, \quad d_{n+1} = 4 \cdot f(r_n + 2, i_n - 0.5) = d_n + 8i_n - 4(1 + \alpha) \quad (5.5a)$$

$$\text{- pentru } d_n > 0, \quad d_{n+1} = 4 \cdot f(r_n + 2, i_n - 1.5) = d_n + 8r_n - 4(3 + \alpha) \quad (5.5b)$$

Prin urmare dacă se cunosc valorile inițiale r_0 , i_0 și d_0 atunci întârzierea i_n corespunzătoare punctului focal situat la distanța r_n unde n este un întreg pozitiv, se calculează cu ajutorul ecuațiilor (5.5a) și (5.5b). Algoritmul descris mai sus poate fi exprimat astfel:

```
/*Se definesc constantele C1 și C2 astfel: */
C1 = -4(1+α);
C2 = -4(3+α);
/*Se inițializează r, i și d */
r = r0;
i = i0;
d = d0;
/*Descrierea algoritmului */
for (n=0; n<N; n++){
    output (r, i);
    if (d ≤ 0){
        d = d+8i+C1;
    }
    else{ d = d+8r+C2;
        i = i-1;
    }
    r = r+1;
}
```

O diagramă bloc funcțională care implementează algoritmul descris mai sus este dată în figura 5.5. Generatorul de coeficienți include două registre care stochează valorile coeficienților C1 și C2 și un multiplexor pentru selecția valorii dorite la un moment dat. Numărătorul N1 se încarcă cu valoarea inițială r_0 și contorizează prin numărare înainte valoarea incrementală a poziției punctului focal curent. Numărătorul N2 se încarcă cu

valoarea inițială i_0 și contorizează prin numărare înapoi valoarea incrementală a întârzierii pentru punctul focal curent.

Ieșirile celor două numărătoare sunt aduse la intrările multiplexorului MUX2. Multiplicatorul M multiplică cu +8 sau -8 ieșirea multiplexorului MUX2 și livrează rezultatul multiplicării sumatorului S1 care îl însumează cu ieșirea multiplexorului MUX1. Sumatorul S2 adună la conținutul registrului ieșirea sumatorului 1. În acest mod registrul conține valoarea actualizată a variabilei de decizie D el fiind inițializat în prealabil cu valoarea inițială d_0 a întârzierii. Generatorul tactului de eșantionare primește valoarea inițială a întârzierii și va genera inițial tactul de eșantionare i_0 care corespunde punctului focal situat la distanța r_0 . Cel mai semnificativ bit (MSB) al variabilei D stocate în registru la un moment dat reprezintă semnul variabilei de decizie D și va fi folosit pentru a selecta care dintre constantele C1, C2 se folosește pentru a actualiza valoarea lui D și pentru a actualiza valoarea întârzierii cu sau fără decrementare. În acest scop semnalul MSB este adus la intrarea de selecție a multiplexoarelor MUX1 și MUX2 și la intrarea de autorizare a numărării la numărătorul N2. Când MSB este la nivel scăzut multiplexorul MUX1 dă la ieșire constanta C2, iar numărătorul N2 își decrementează valoarea inițială i_0 cu o unitate. Numărătorul N1 primește valoarea inițială r_0 și se incrementează cu o unitate. Multiplexorul MUX2 livrează valoarea i_0 de la ieșirea lui N2 la multiplicatorul M în concordanță cu valoarea MSB. Multiplicatorul înmulțește această valoare cu +8 sau -8 funcție de valoarea lui MSB, înmulțirile reducându-se la simple deplasări la stânga sau la dreapta. Sumatorul S1 adună constanta cu rezultatul multiplicării ieșirea sa având valoarea $-8r+C2$. Sumatorul S2 adună ieșirea lui S1 la valoarea inițială d_0 stocată în registru și prin urmare conținutul registrului va fi $d-8r+C2$. Generatorul tactului de eșantionare primește data stocată în registru și o compară cu 0, apoi actualizează întârzierea la valoarea i_0-1 dacă valoarea din registru este mai mare ca zero și generează un tact de eșantionare întârziat cu valoarea i_0-1 . Dacă valoarea din registru este mai mică sau egală cu zero întârzierea se menține la valoarea i_0 și se generează un tact de eșantionare întârziat cu valoarea i_0 . Poziția punctului focal se incrementează cu o unitate și procesul de reia.

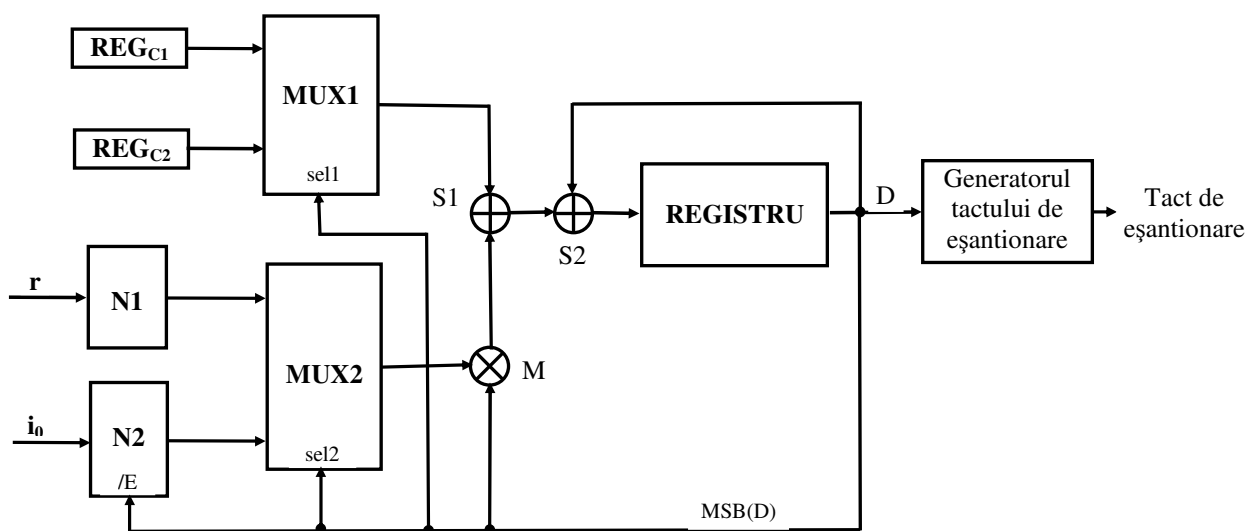


Fig.5.5 Diagrama bloc pentru implementarea algoritmului punctului median – varianta 1

Dacă definim unitatea de distanță ca fiind distanța între două puncte focale (proporțională cu timpul necesar parcurgerii acestuia) sistemul care implementează algoritmul prezentat trebuie să actualizeze variabila de decizie d la fiecare incrementare a lui r cu unitatea de distanță. Pentru situația când se doresc frecvențe de eșantionare ridicate volumul calculelor crește semnificativ. În această situație dacă valoarea lui i nu se calculează pentru fiecare $r = n$ ci

doar pentru $r = Kn$, K și n fiind întregi, volumul calculelor se reduce cu $1/K$. Pentru această situație condiția (5.2a) devine:

$$-\frac{1}{K} \leq \frac{dl}{dr} \leq 0 \quad (5.6)$$

iar algoritmul se modifică prin substituirea lui r cu Kr după cum urmează:

$$f(r, l) = l^2 + 2 \cdot l \cdot K \cdot r - \alpha \cdot K \cdot r - \beta \quad (5.7)$$

$$d_n = 4i_n^2 + 4i_n(2K - 1) + 8Ki_n r_n - 4Kr_n(1 + \alpha) - 4K(1 + \alpha) - 4\beta + 1 \quad (5.8)$$

$$\text{- pentru } d_n \leq 0, \quad d_{n+1} = d_n + 8Ki_n - 4K(1 + \alpha) \quad (5.9)$$

$$\text{- pentru } d_n > 0, \quad d_{n+1} = d_n - 8(1 - K)i_n - 8Kr_n - 4K\alpha - 20K + 8 \quad (5.10)$$

Algoritmul descris mai sus se generalizează după cum urmează:

```

/*Se definesc constantele C1 și C2 astfel: */
C1 = -4·K·(1+α);
C2 = -4·K·(5+α)+8;
/*Se inițializează r, i și d */
r = r0;
i = i0;
d = d0;
/*Descrierea algoritmului */
for (n=0; n<N; n++){
  output (r, i);
  if (d ≤ 0){
    d = d+8·K·i+C1;
  }
  else{ d = d-8·(1-K) i-8·K·r+C2;
    i = i-1;
  }
  r = r+1;
}

```

Diagrama bloc funcțională care implementează algoritmul descris mai sus este dată în figura 5.6.

Întrucât termenii $-4K(1 + \alpha)$ și $-4K\alpha - 20K + 8$ sunt constante ce pot fi inițializate, din ecuațiile de mai sus rezultă că pentru calculul funcției de decizie sunt necesare doar operații de adunare cu trei și respectiv patru termeni. Implementarea algoritmului doar cu operații de adunare cu doi termeni este posibilă dacă funcția de decizie se exprimă prin relațiile:

$$\text{- pentru } d_n \leq 0, \quad d_{n+1} = d_n + A_n \quad (5.11)$$

$$\text{- pentru } d_n > 0, \quad d_{n+1} = d_n + B_n \quad (5.12)$$

unde A_n și B_n au expresiile:

$$\text{- } A_n = 8Ki_n - 4K(1 + \alpha) \quad (5.13)$$

$$\text{- } B_n = -8(1 - K)i_n - 8Kr_n - 4K\alpha - 20K + 8 \quad (5.14)$$

Cu aceste modificări A_{n+1} și B_{n+1} pot fi exprimate incremental astfel:

$$\text{- pentru } d_n \leq 0, \quad A_{n+1} = A_n, \quad B_{n+1} = B_n - 8K \quad (5.15)$$

$$\text{- pentru } d_n > 0, \quad A_{n+1} = A_n - 8K, \quad B_{n+1} = B_n - 16K + 8 \quad (5.16)$$

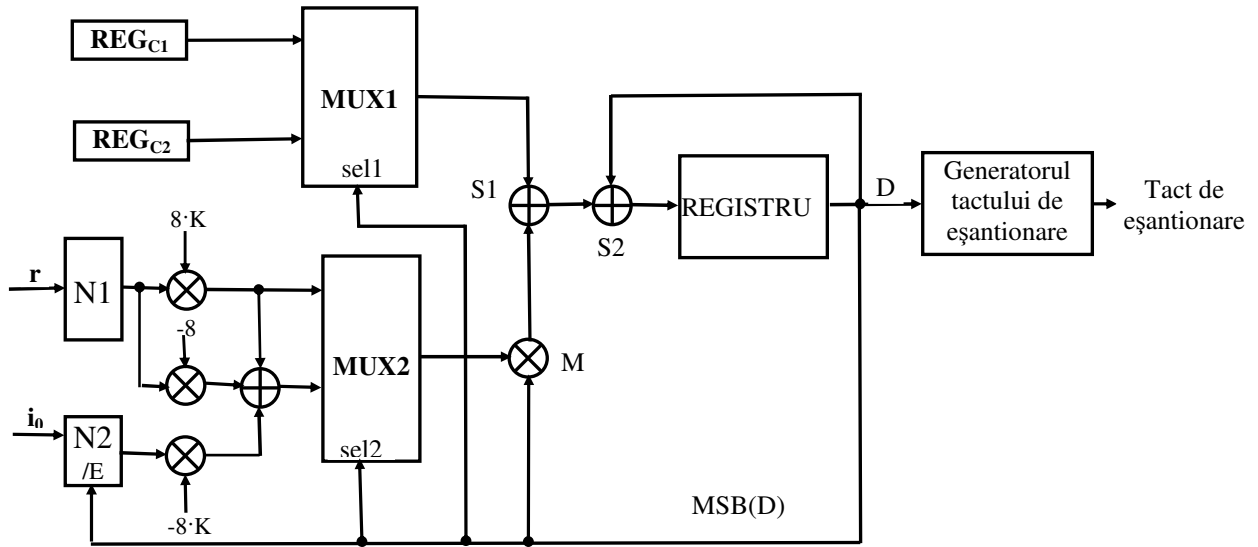


Fig. 5.6 Diagrama bloc pentru implementarea algoritmului punctului median – varianta 2

Prin urmare dacă se cunosc valorile inițiale A_0 și B_0 atunci A_n , B_n , d_n , r_n și i_n se pot obține secvențial când $n = 0, 1, 2, \dots$

Algoritmul descris poate fi exprimat astfel:

```

/*Se definesc constantele C1, C2 */
C1 = -8·K; C2 = -16·K+8;
/*Se inițializează r, i, d, A, B */
r = r0; i = i0; d = d0; A = A0; B = B0;
/* A0 = 8Ki0 -4K(1+α); B0 = -8(1-K)i0 -4Kα-20K+8; */
/*Descrierea algoritmului */
for (n=0; n<N; n++){
    output (r, i);
    if (d ≤ 0){
        d = d+A; B =B+C1;}
    else{ d = d+B;
        i = i-1; A =A+C1; B =B+C2;}
    r = r+1;}

```

Diagrama bloc funcțională care implementează algoritmul descris mai sus este dată în figura 5.7.

Schema bloc conține doar registre, multiplexoare și blocuri sumatoare cu două intrări. Valorile constantelor $C1$ și $C2$ stocate în REG_{C1} și REG_{C2} sunt disponibile la intrările multiplexorului MUX1. Registrul REG1 stochează valoarea actualizată a termenului A , iar REG2 stochează valoarea actualizată a termenului B . Actualizarea este comandată de semnul (MSB – cel mai semnificativ bit) variabilei de decizie d , care este conținută în registrul REG3. MSB(d) selectează totodată care din intrările multiplexorului MUX2 este adunată la conținutul registrului REG3 pentru a obține valoarea actuală a variabilei decizie. Generatorul

tactului de eşantionare decrementează valoarea întârzierii de fiecare dată când variabila d este pozitivă şi generează un impuls de eşantionare.

După cum se poate observa în figura 5.4 condiţia dată de relaţia (5.6) nu mai poate fi respectată pentru întreg domeniul pentru valori ale lui K mai mari decât 1. Tabelul 5.3 indică dependenţa domeniului minim pentru care este respectată condiţia (5.6) de valoarea lui K .

Tabelul 5.3

$K = 1$	$r_{\min} = 0$
$K = 2$	$r_{\min} = 27,4 \cdot \lambda$
$K = 3$	$r_{\min} = 51,9 \cdot \lambda$
$K = 4$	$r_{\min} = 75,5 \cdot \lambda$
$K = 5$	$r_{\min} = 98,8 \cdot \lambda$
$K = 6$	$r_{\min} = 121,8 \cdot \lambda$
$K = 7$	$r_{\min} = 144,8 \cdot \lambda$
$K = 8$	$r_{\min} = 167,7 \cdot \lambda$

Problema care se pune este alegerea valorii potrivite pentru K funcţie de cerinţele aplicaţiei date. Este de dorit adoptarea unei valori cât mai ridicate a lui K deoarece frecvenţa de efectuare a calculelor scade proporţional dar în acelaşi timp eroarea de aproximare a curbei întârziere nu trebuie să depăşască incrementul de domeniu (distanţa între două puncte focale) care este egal cu raportul dintre lungimea de undă şi factorul de supraeşantionare λ/OSR .

Folosirea ponderării de tip FN (f number) pentru îmbunătăţirea calităţii imaginii adaugă o cerinţă suplimentară la alegerea lui K . Ponderarea FN se exprimă prin relaţia: $\text{FN} \leq r/D$ unde FN este o constantă întreagă predeterminată ($\text{FN} = 1, 2, 3, \dots$), r poziţia punctului focal curent iar D reprezintă lungimea ariei şi arată că pentru punctele focale apropiate de suprafaţa ariei nu toată aria ci doar anumite elemente sunt autorizate să participe la sinteza fasciculului receptor, iar în exteriorul unui domeniu dat toate elementele ariei sunt activate la recepţie. Cu alte cuvinte domeniul punctului focal receptor este întotdeauna mai mare decât dimensiunea ariei fiind legat de aceasta printr-un factor constant $r \geq \text{FN} \cdot D$. Pentru situaţia cea mai defavorabilă când $\text{FN} = 1$ poziţia primului punct focal care include contribuţia celui mai îndepărtat element este la $r = 128 \cdot \lambda / 2 = 64 \cdot \lambda$. Din tabelul de mai sus se observă că se poate alege pentru K valoarea 3. Din motive de simplificare a circuitelor este de preferat să se lucreze cu $K = 2$. Figurile 5.8a – 5.8d indică modul de operare a algoritmului punctului median pentru trei valori ale lui K : 1, 2, 4, iar figura 5.9 arată comparativ eroarea de aproximare în cele trei situaţii. Rezultatele simulărilor MATLAB confirmă raţionamentul pentru alegerea valorii $K=2$ care asigură, pentru valori ale lui r mai mari de 32λ ($512 \lambda/16$) încadrarea erorii de aproximare a curbei reale în intervalul $[-0,5 - 0,5]$.

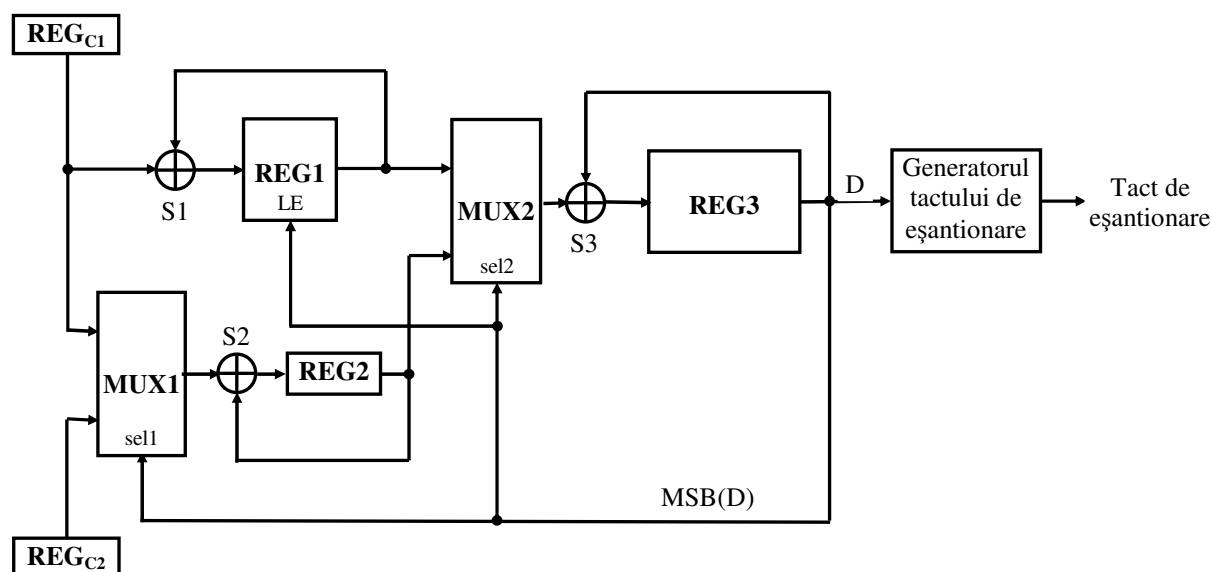


Fig. 5.7 Diagrama bloc pentru implementarea algoritmului punctului median – varianta 3

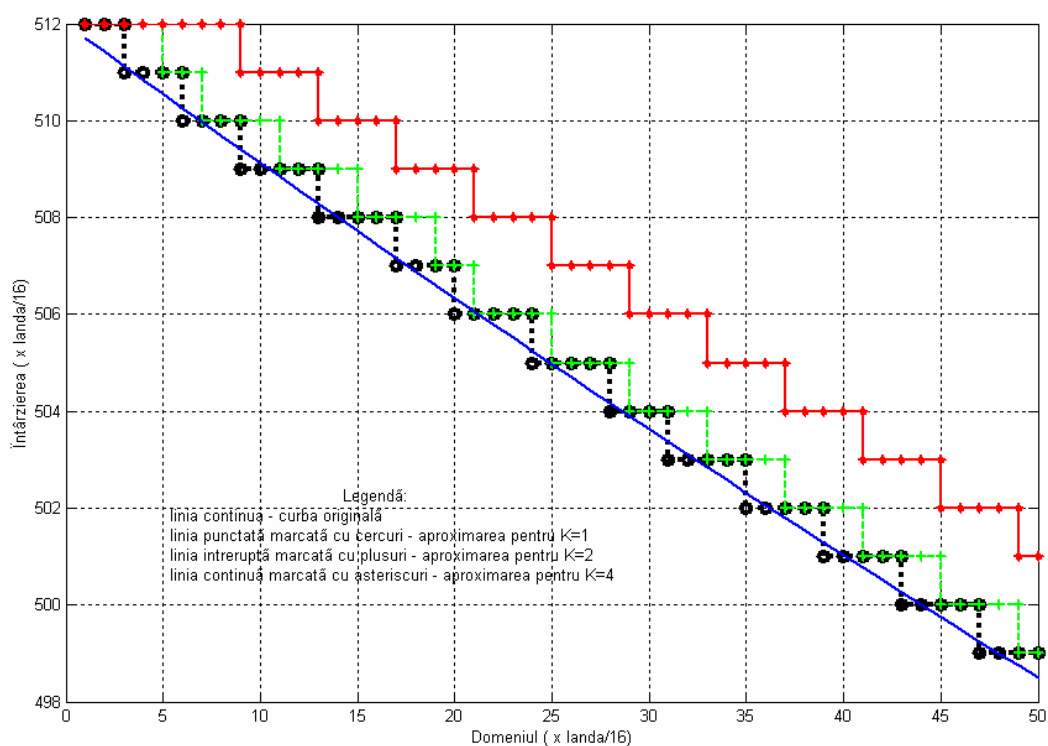


Fig. 5.8a Algoritmul punctului median – simulare (detaliu)

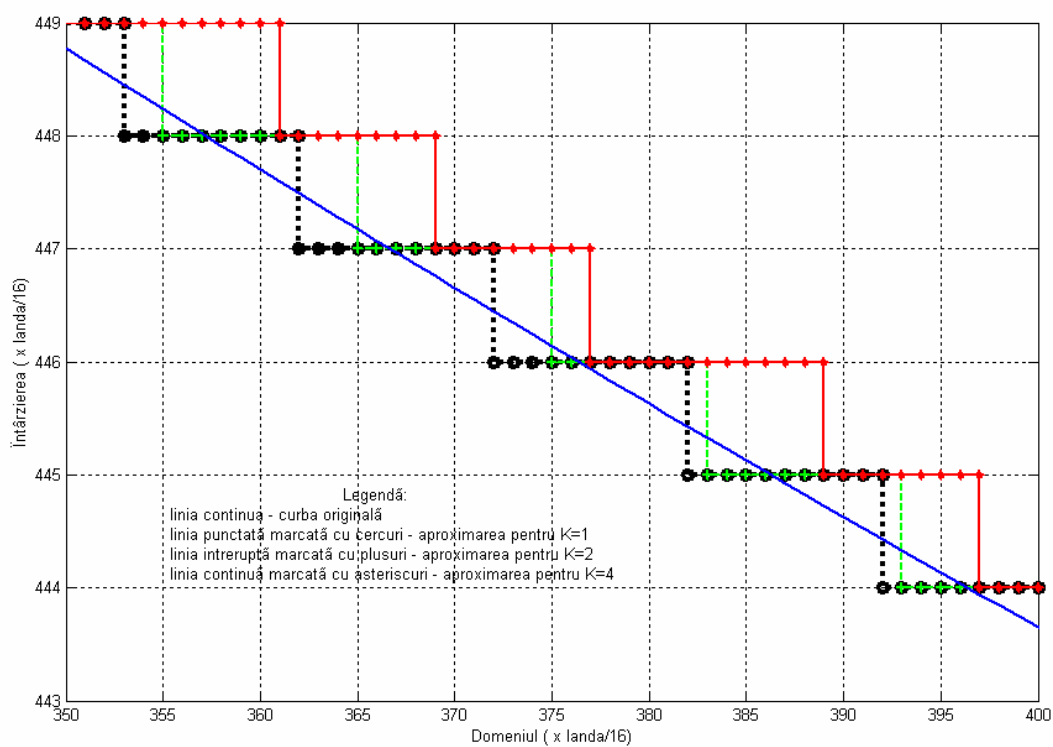


Fig. 5.8b Algoritmul punctului median – simulare (detaliu)

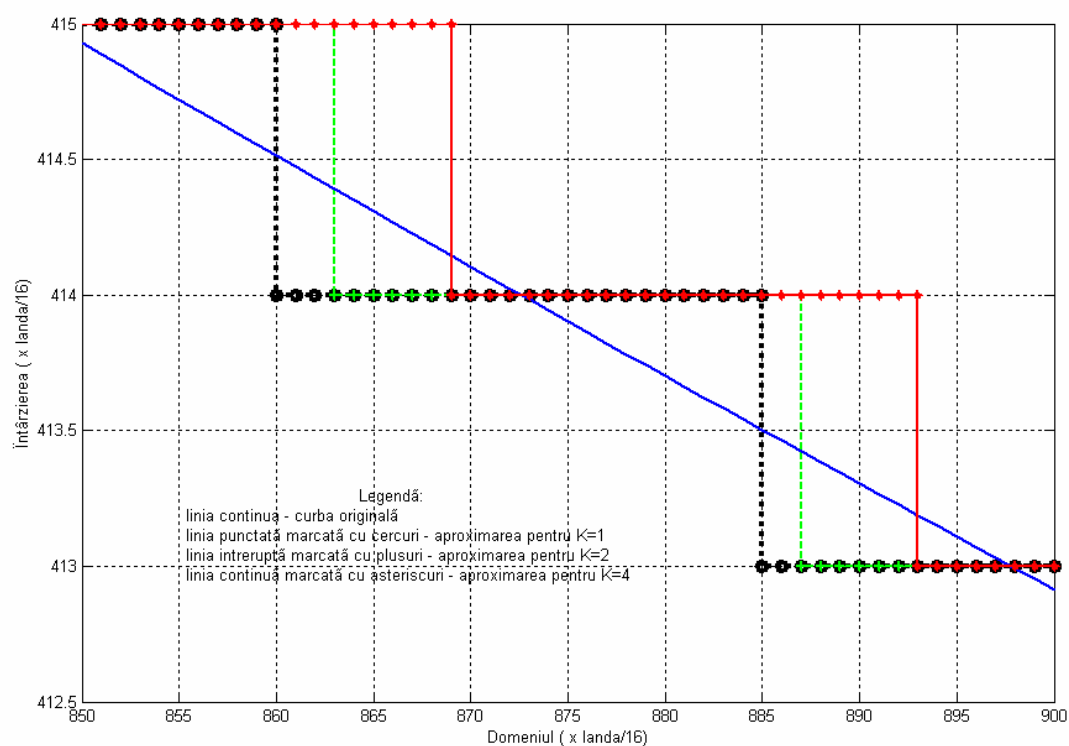


Fig. 5.8c Algoritmul punctului median – simulare (detaliu)

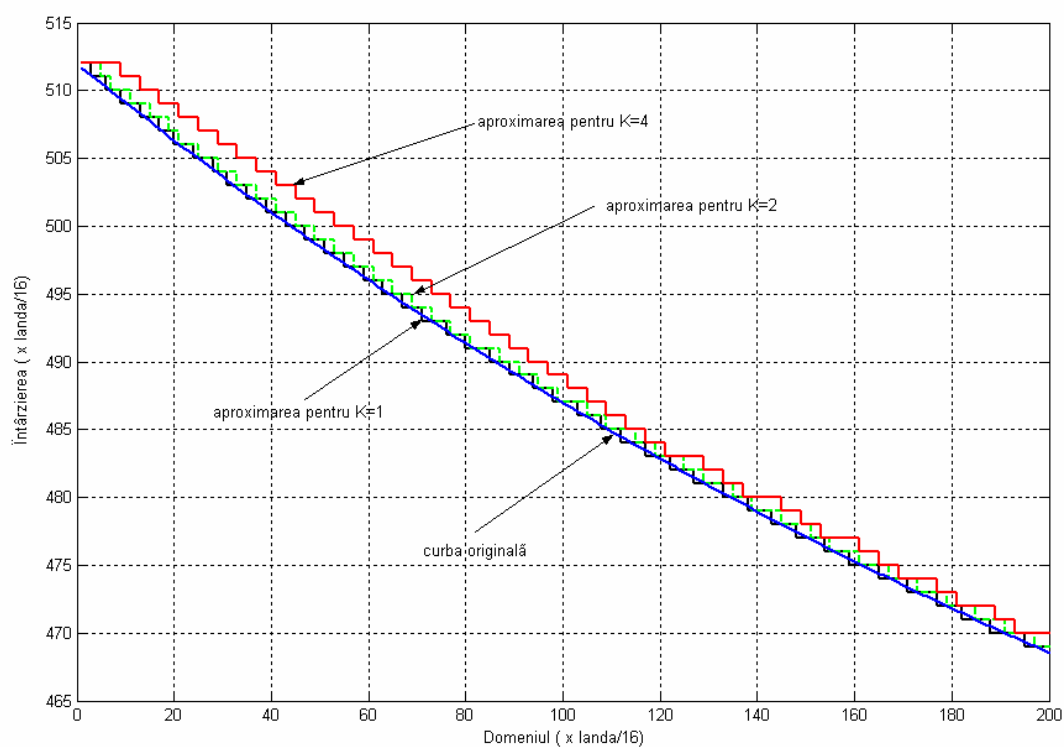


Fig. 5.8d Algoritmul punctului median – simulare (detaliu)

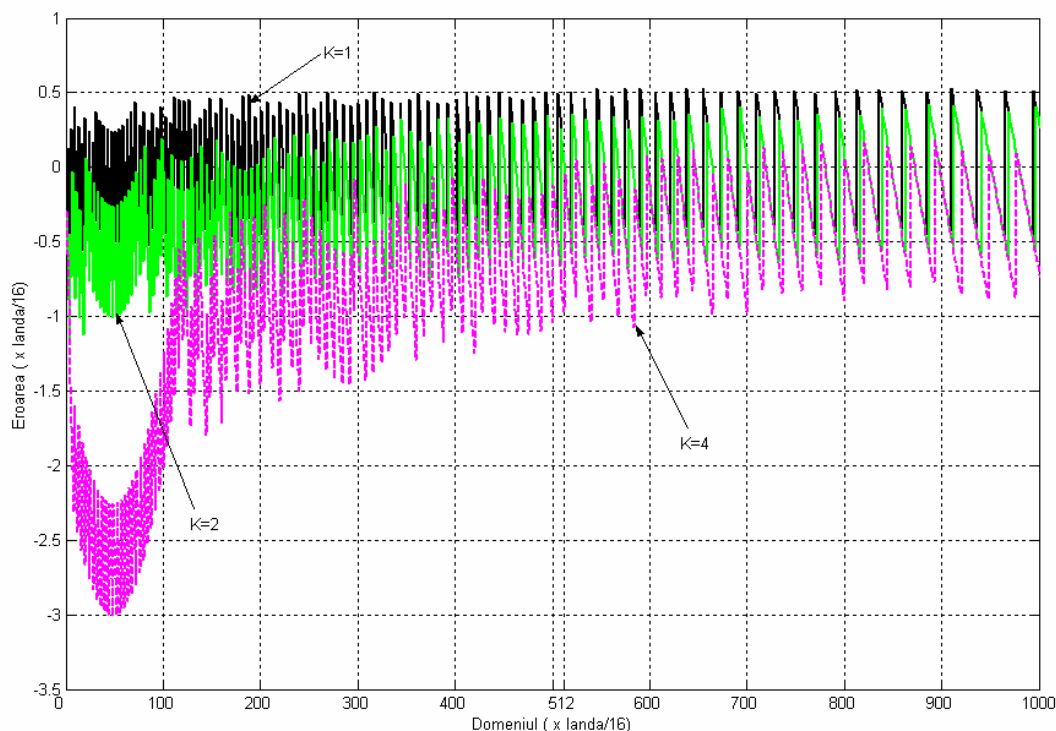


Fig. 5.9 Algoritmul punctului median – eroarea de aproximare (simulare MATLAB)

5.2.3 IMPLEMENTAREA GENERATORULUI

Entitățile VHDL folosite la descrierea structurii generatorului de tact pentru eşantionarea neuniformă sunt:

Const – blocul constantelor – conține valorile constantelor A și B precum și valoarea inițială a întârzierii i_0 . În structura finală blocul de constante se substituie cu un bloc de memorie din care se încarcă succesiv valorile inițiale A, B și i_0 , care definesc direcțiile de scanare

R_Counter - contorizează întârzierea inițială i_0 .

Sign_Calculator – actualizează valoarea funcției decizie și evaluează semnul acesteia

Scg_Comm - automat secvențial care comandă entitățile precedente pentru efectuarea operațiilor de încărcare, așteptare, activare și calcul.

Figura 5.10 arată modul de interconectare a entităților și semnalele aferente.

Verificarea practică a funcționării s-a realizat pentru un canal implementat într-un FPGA din familia FLEX10 – EPF10K20. S-a folosit pentru aceasta pachetul livrat de firma ALTERA în cadrul programului universitar pentru dezvoltarea aplicațiilor practice de laborator. Pachetul conține:

- Mediul software pentru dezvoltarea aplicațiilor MAX+PLUS II versiunea 10.1
- Placa de dezvoltare UP1 echipată cu un FPGA EPF10K20 și un CPLD EPM7128S

Structura unui canal a fost implementată în FPGA-ul de capacitate maximă din familia FLEX10K, circuitul EPF10K250ABC600. Utilizarea dispozitivului data în tabela 5.4 arată că acesta poate gazdui generatoarele tactului de esantionare pentru o arie cu 64 de elemente și prin urmare există bune șanse de a implementa într-un dispozitiv Altera de capacitate mai mare, cum sunt cele din familia STRATIX, secțiunea digitală a fasciculului receptor.

Verificarea soluției s-a făcut după o secvență de validare care poate fi aplicată pentru proiecte implementate în circuite programabile. Secvența de testare presupune parcurgerea următoarelor etape:

- modelarea Matlab a algoritmului și generarea unui fișier de date de ieșire

- generarea în Matlab a semnalelor de intrare pentru circuitul implementat
- simularea circuitului în Orcad – PLD folosind ca intrare semnalul generat
- transferul rezultatelor simulării în mediul MATLAB și folosirea capabilităților grafice ale acestuia pentru a compara rezultatele modelării cu cele ale simulării.
- achiziția pe un osciloscop digital a semnalului generat de PLD
- transferul datelor achiziționate în MATLAB și compararea cu rezultatele simulării.

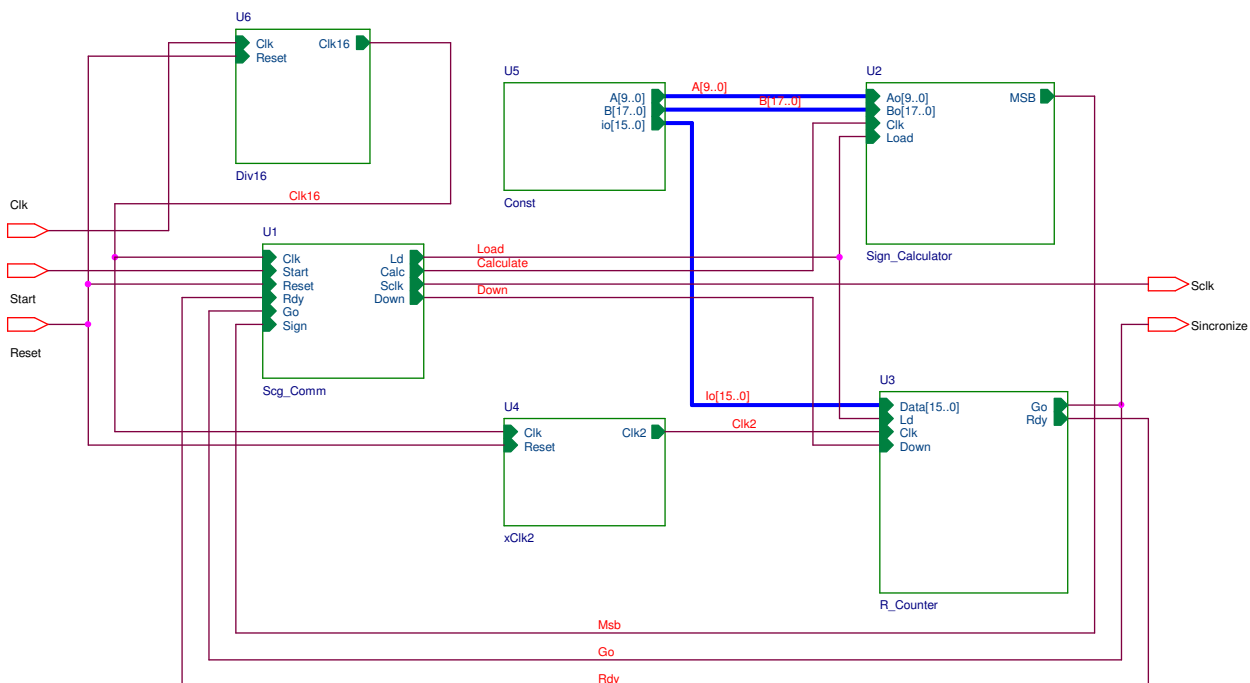


Figura 5.10 Interconectarea entităților generatorului de tact

Tabelul5.4: Utilizarea dispozitivului EPF10K250ABC600

Resource	Used	Available	Utilization
IOs	5	470	1.06%
LCs	149	12160	1.23%
DFFs	69	12160	0.57%
Memory Bits	0	40960	0.00%
CARRYs	11	12160	0.09%
CASCADEs	0	12160	0.00%

În figurile 5.17 și 5.18 sunt prezentate rezultatele simulării pentru generarea tactului de esantionare pentru un canal

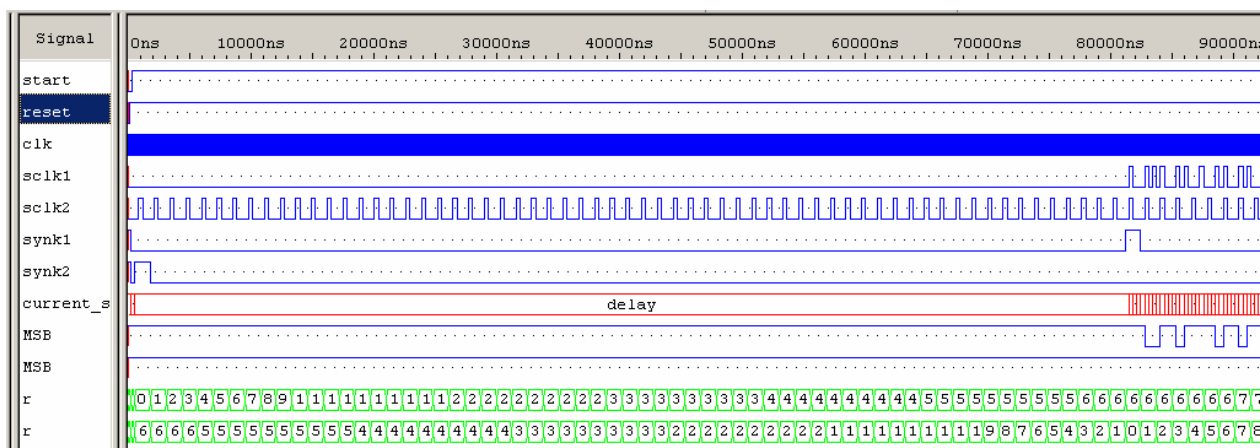


Figura 5.11 Simularea generării tactului de esantionare pe baza algoritmului punctului median

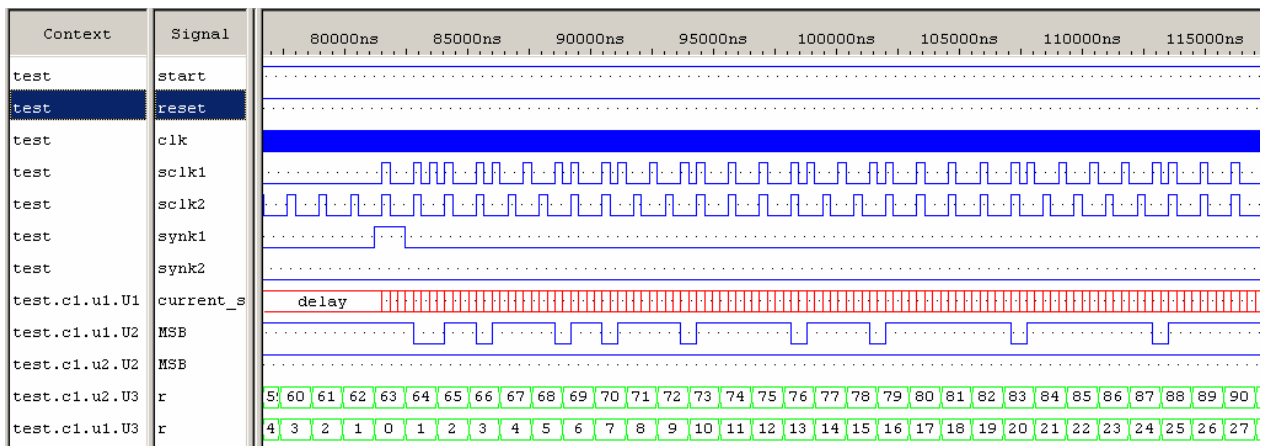


Figura 5.12 Simularea generării tactului de eşantionare pe baza algoritmului punctului median - detaliu

5.3. MEMORIA FIFO ȘI SUMATORUL

5.3.1 DIMENSIONAREA MEMORIEI FIFO

Informația de la ieșirea convertorului delta-sigma cuantificată pe un bit se stochează într-un bufer de memorie tip FIFO sau dual port care se încarcă cu tactul neuniform de eșantionare și se descarcă cu tactul uniform care comandă sumatorul. Lungimea unui buffer de memorie este dată de domeniul maxim investigat și de factorul de supraeșantionare. Astfel pentru o lungime a liniei de scanare $R=20\text{cm}$, o frecvență a traductorului $f_{tr} = 3,5\text{Mhz}$ și un $OSR = 16$ lungimea bufferului va fi :

$$L_B = \frac{2 \cdot R \cdot f_{tr} \cdot OSR}{c} \cong 15000$$

Având în vedere modularizarea matricii de memorie încorporată din dispozitivele Altera APEX și STRATIX în blocuri de 4 K este recomandabil de ales o lungime a buferului de 16K. Pentru o arie de 128 elemente rezultă un necesar de memorie de $128 \cdot 16K = 2048K$ biți.

5.3.2 SUMATORUL – OPTIMIZAREA STRUCTURII

Eșantioanele codate pe un bit, aliniate prin bufferele FIFO de pe fiecare canal sunt transmise simultan sumatorului. Acesta generează versiunea grosieră a fasciculului care conține suma ecourilor întârziate plus zgomotul de cuantificare. Sumatorul care primește la intrare, sincron, eșantioanele de la cele 128 canale și dă la ieșire suma acestora. Pentru a reduce întârzierea introdusă de sumator acesta a fost conceput într-o structură piramidală cu 6 nivele de însumare. Lungimea cuvintelor ce se insumează este incrementată cu o unitate, iar numărul celulelor de însumare se înjumătățește la trecerea pe nivelul de însumare superior

5.3.3 DESCRIERE VHDL

Buferul de memorie – a fost construit în structura unui bloc de memorie RAM cu două porturi pe baza modulului de librărie parametrizabil *lpm-ram-dp* disponibil în bibliotecile Altera Maxplus+II și Quartus. Structura scalară a acestor module permite implementarea oricărui tip de memorie RAM sau ROM.

Operarea buferului de memorie este asincronă la scriere și sincronă la citire. Tactul de scriere este generat simultan cu adresa de scriere în generatorul semnalului de eșantionare separat pentru fiecare canal. Tactul și adresa de citire se generează simultan pentru toate bufferele de către blocul de comandă al filtrului. Cu același tact se comandă și sumatorul care primește la intrare, sincron, eșantioanele de la cele 128 canale și dă la ieșire suma acestora. Pentru a reduce întârzierea introdusă de sumator acesta a fost conceput într-o structură piramidală cu 6 nivele de însumare.

Implementarea sumatorului se bazează pe arhitectura FLEX care pune la dispoziție 2 căi de date dedicate de mare viteză care conectează LE-uri adiacente nefolosind căile locale de interconectare: transportul rapid – “carry chain” și conectarea în cascadă – “cascade chain”. “Carry chain” pune la dispoziție o funcție de transport în avans foarte rapidă (până la 0,2 ns) între LE-uri. Semnalul de carry-in provenit de la un bit cu prioritate mai mică trece mai departe într-un bit cu prioritate mai mare prin lanțul carry și ajunge atât în LUT cât și în secțiunea următoare a lanțului carry. Această facilitate permite arhitecturii FLEX să implementeze sumatoare de lărgimi arbitrare, foarte rapide, într-un mod eficient.

Codul VHDL pentru memoria FIFO și pentru sumatorul cu arhitectură piramidală se găsește în anexa 2

5.4. FILTRUL DECIMATOR

Ieșirea sumatorului reprezintă suma semnalelor date de modulatorile delta-sigma și cuprinde pe lângă semnalul util componente de frecvențe înalte, zgomotul de cuantificare, zgomotul introdus de circuite și interferențele. Obținerea unui semnal corespunzător pentru procesările digitale ulterioare necesită atenuarea componentelor din afara benzii utile cu ajutorul unui filtru digital și reeșantionarea la rata Nyquist. Figura 5.13 prezintă diagrama bloc a unui filtru digital decimator. Mai întâi semnalul este introdus într-un filtru digital trece-jos care aproximează caracteristica ideală

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq 2\pi f_D T / 2 = \pi / D \\ 0, & \text{în rest} \end{cases} \quad (5.17)$$

unde f_D reprezintă rata de reeșantionare f/D . Reducerea ratei de eșantionare se poate realiza formând secvența de ieșire prin extragerea de eșantioane cu pasul D de la ieșirea filtrului. Dacă notăm răspunsul la impuls al filtrului trece jos cu $h[n]$ ieșirea filtrată $w[n]$ se poate

scrie: $w[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k]$ iar ieșirea decimatorului este $y[m] = w[Dm]$

Combinând relațiile de mai sus se obține $y[m] = \sum_{k=-\infty}^{\infty} h[k]x[mD-k]$. (5.18)

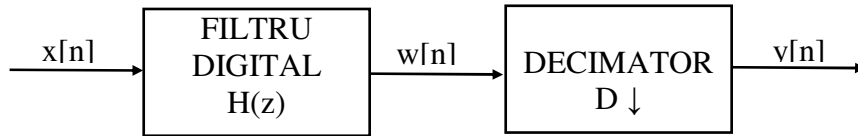


Fig. 5.13 Diagrama bloc a unui filtru digital decimator

Datorită prezenței în semnalul modulat a componentelor exterioare benzii utile este necesar un filtru trece jos cu pantă abruptă. Realizarea unor astfel de filtre pentru frecvențe ridicate necesită costuri semnificative și ca urmare implementarea lor trebuie atent analizată. Deoarece pentru aplicațiile de formare de fascicul este de importanță crucială conservarea informației de fază din semnalele recepționate, filtrul decimator trebuie să asigure o întârziere de grup constantă pentru toate frecvențele sau astfel spus o fază liniară. Datorită structurii nerecursive filtrele FIR sunt întotdeauna stabile și dacă au coeficienții simetrici asigură faza liniară.

5.4.1 FILTRU FIR CU FAZĂ LINIARĂ: STRUCTURĂ ȘI PROPRIETĂȚI

În general un filtru digital este complet caracterizat prin ecuația sa diferențială care descrie dependența intrare ieșire. Ecuația diferențială pentru un filtru digital este:

$$y[n] = a_0x[n] + a_1x[n-1] + a_2x[n-2] + \dots + a_Lx[n-L] - b_1y[n-1] - b_2y[n-2] - \dots - b_My[n-M] \quad (5.19)$$

Deoarece un filtru FIR este un sistem nerecursiv nu va conține termenii de reacție. El este caracterizat prin răspunsul la impuls $h[n]$. Ca atare ecuația diferențială pentru un filtru FIR de ordinul N este:

$$y[n] = h[0]x[n] + h[1]x[n-1] + h[2]x[n-2] + \dots + h[N]x[n-N] \quad (5.20)$$

și poate fi exprimată prin convoluția finită dintre semnalul de intrare $x[n]$ și răspunsul la impuls $h[n]$ $y[n] = \sum_{k=0}^N h[k]x[n-k]$. (5.21)

Funcția de transfer poate fi obținută din transformata z a răspunsului la impuls $h[n]$

$$H[z] = \sum_{n=0}^N h[n]z^{-n} \quad (5.22)$$

Constrângerea impusă pentru a obține un răspuns cu fază liniară este ca răspunsul finit la impuls să aibă simetrie conjugată pară sau impară față de punctul său median. Pentru a vedea dacă această constrângere asigură faza liniară se consideră filtrul FIR cu funcția de transfer dată de (4.6) pentru care $h[n] = \pm h^*[N-n] = |h[n]e^{j\phi_n}|$.

Dacă N este par, coeficientul $h[N/2]$ este real și corespunde centrului de simetrie al lui $h[n]$, iar pentru N impar nu avem coeficient central. Funcție de tipul de simetrie și de ordinul filtrului se deosebesc patru tipuri de filtre cu fază liniară.

Tip I	simetrie pară	ordin par
Tip II	simetrie pară	ordin impar
Tip III	simetrie impară	ordin par
Tip IV	simetrie impară	ordin impar

Pentru filtrele tip III și IV simetria impară implică ieșire nulă pentru orice intrare constantă – de curent continuu, $x[n]=a$

$$y[n] = a \sum_{k=0}^N h[k] = aH(1) = 0 \quad (5.23)$$

Ca atare pentru filtrele III și IV, $H(z)$ trebuie să aibă un zero la $z=1$. Pe de altă parte dacă se aplică la intrare o secvență Nyquist $x[n]=(-1)^n$, filtrele II și III dau zero la ieșiri deoarece

$$y[n] = \sum_{k=0}^N h[k](-1)^{n-k} = H(-1) = 0 \quad (5.24)$$

și prin urmare $H(z)$ are un zero la $z=-1$.

Poziționarea acestor zerouri la $z=-1$ și $z=1$ indică următoarele:

- numai filtrele de tipul I pot implementa caracteristica trece tot.
- filtrele de tipul II pot implementa caracteristica de tip trece jos
- filtrele tip III și IV nu pot fi folosite pentru implementarea de filtre trece jos.

Pentru a arăta că răspunsul are fază liniară se consideră un filtru de tipul I cu coeficienți reali, simetrie pară și N par. Relația (4.6) se poate scrie astfel:

$$H[z] = z^{-N/2} \sum_{n=0}^N h[n]z^{-n+N/2} = z^{-N/2} \left(h\left[\frac{N}{2}\right] + \sum_{n=0}^{\frac{N}{2}-1} \left(h[n]z^{-n+\frac{N}{2}} + h[N-n]z^{n-\frac{N}{2}} \right) \right) \quad (5.25)$$

Substituind $z=e^{j\omega}$ și ținând seama că $h[n]=h[N-n]$ se obține răspunsul în frecvență

$$H(\omega) = e^{-j\omega \frac{N}{2}} \left(h\left[\frac{N}{2}\right] + \sum_{n=0}^{\frac{N}{2}-1} 2h[n] \cos\left(\left(\frac{N}{2}-n\right)\omega\right) \right) = e^{-j\omega \frac{N}{2}} R(\omega), \quad (5.26)$$

unde $R(\omega)$ este real. Dacă $R(\omega)$ își păstrează semnul pentru orice ω , atunci $R(\omega) = \pm |H(\omega)|$ și se obține un răspuns cu fază liniară $\angle H(\omega) = -\omega \frac{N}{2} + C$, unde $C = 0$ sau π . Întârzierea de grup este constantă $D(\omega) = -\frac{d}{d\omega} \angle H(\omega) = \frac{N}{2}$. (5.27)

Proiectarea filtrelor FIR cu fază liniară folosind algoritmul Remez

În ultimii 30 de ani s-au dezvoltat două metode de proiectare a filtrelor FIR:

- metoda ferestrei (the window method)
- metoda ponderării Cebîșev (the Chebyshev method)

Metoda ferestrei folosește seriile Fourier în conjuncție cu o clasă de funcții cunoscute sub denumirea de funcții fereastră.

Avantajele metodei sunt:

- proiectarea presupune un volum redus de calcule
- este ușor de aplicat
- conduce la expresii compacte

Dezavantajele metodei:

- proiectarea este suboptimală
- satisfacerea cerințelor din specificație conduce la filtre de ordin mare
- un filtru de ordin ridicat presupune mai multe calcule per eșantion ceea ce face ca aceste filtre să fie mai lente și mai puțin eficiente în aplicațiile în timp real

Metoda ponderării Cebîșev este o metodă de optimizare multivariabilă iterativă care are la bază algoritmul de schimbare Remez.

Avantaje:

- proiectare optimală în sensul minimizării erorii maxime dintre răspunsul actual în frecvență și cel dorit.
- metoda este foarte flexibilă – poate fi folosită pentru a proiecta filtre, transformatoare Hilbert, diferențiatoare etc.
- Conduce la soluții cu ondulații simetrice în bandă (“equiripple filters”)
- Se obține filtrul de ordin minim care satisface specificația dată
- ordinul minim pentru filtru presupune un filtru rapid și eficient pentru aplicațiile în timp real

Dezavantaje

- procedura de proiectare necesită un volum mare de calcule
- nu este adecvată pentru aplicațiile care necesită proiectarea în timp real sau cvasi real (filtre adaptive sau programabile)

Deoarece pentru filtrele cu ondulații simetrice (“equiripple filters”) nu există o soluție analitică se aplică o metodă de proiectare iterativă. Algoritmul Remez este una din aceste metode.

Specificațiile pentru banda de trecere și respectiv pentru cea de atenuare sunt date prin expresiile:

$$\begin{aligned} 1 + \delta_1 &\geq |H(\omega)| \geq 1 - \delta_1, & 0 \leq \omega \leq \omega_c \\ |H(\omega)| &\leq \delta_2, & \omega_r \leq \omega \leq \pi \end{aligned} \quad (5.28)$$

Pentru exemplificare se presupune cazul unui filtru cu răspuns la impuls de tip I, simetric în jurul lui $n=0$. Ca urmare $h[n] = h[-n]$ și se obține următorul răspuns în frecvență:

$$H(\omega) = \sum_{n=-N/2}^{N/2} h[n] e^{-j\omega n} = h[0] + \sum_{n=1}^{N/2} 2h[n] \cos(n\omega) \quad (5.29)$$

$H(\omega)$ este o funcție reală pară exprimată ca o serie Fourier de cosinus finită cu coeficienții $h[n]$. Se exprimă $\cos(n\omega)$ ca un polinom trigonometric de ordinul n în $\cos(\omega)$ și se obține pentru $H(\omega)$ o sumă de polinoame care este echivalentă cu un polinom de ordinul n în $\cos(\omega)$:

$$H(\omega) = \sum_{n=0}^{N/2} a_n (\cos(\omega))^n \quad (5.30)$$

Un parametru important pentru algoritmul Remez este numărul total de puncte de extrem în intervalul Nyquist $[0, \pi]$. Pentru a afla acest număr se determină numărul de zerouri pentru derivata lui $H(\omega)$.

$$\frac{d}{d\omega} H(\omega) = 0 = -\sin(\omega) \sum_{n=0}^{\frac{N}{2}} n a_n (\cos(\omega))^{n-1} \quad (5.30a)$$

Aceasta este un polinom trigonometric de ordinul $N/2-1$ și prin urmare va avea $N/2-1$ rădăcini distincte. Factorul $\sin(\omega)$ implică soluțiile 0 și π și ca urmare sunt posibile $N/2+1$ extreme cu derivata nulă. Dacă se adaugă două extreme pentru limitele celor două benzi se obține în final numărul maxim de extreme posibile $N/2+3$. O teoremă din teoria aproximărilor [Oppenheim99] stabilește că proiectarea este optimizată pentru undulație minimă dacă și numai dacă sunt cel puțin $N/2+2$ extreme de amplitudini egale și semne alternate în benzile de trecere și de atenuare.

Mediul Matlab conține un pachet de programe pentru proiectarea asistată de calculator a filtrelor cu undulații simetrice ("equiripple filters") pe baza algoritmului Remez.

Deși se poate estima precis nu putem ști apriori valorile minime ale amplitudinilor undulațiilor care le asigură un filtru de ordinul N . Dacă se leagă valorile atenuării în banda de oprire și de trecere printr-un factor de proporționalitate K problema se reduce la o singură necunoscută $K\delta_2 = \delta_1 = \delta$ care se determină iterativ odată cu coeficienții filtrului. Dacă nu se poate respecta specificația ordinul trebuie crescut și algoritmul repetat.

Realizarea filtrelor FIR cu fază liniară

S-a arătat anterior că dependența intrare ieșire pentru un filtru FIR de ordinul N se poate exprima prin convoluția finită dintre semnalul de intrare $x[n]$ și răspunsul la impuls $h[n]$

$$y[n] = \sum_{k=0}^N h[k] x[n-k] \quad (5.31)$$

Structura rețelei care realizează convoluția este dată în figura 5.14. Această structură este numită forma directă deoarece este o implementare directă a relației 5.31.

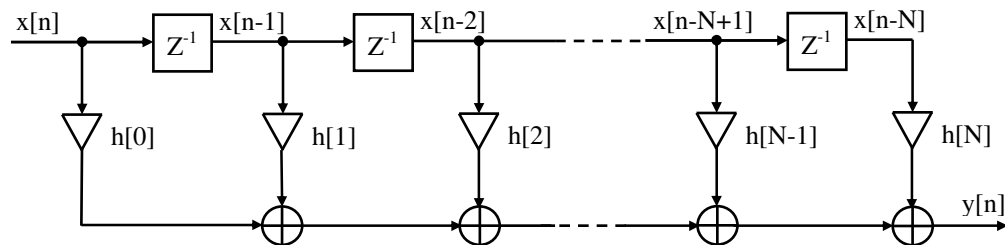


Figura 5.14 Structura rețelei pentru implementarea unui filtru FIR

Structurile în formă directă realizează în general funcția sistemului cu numărul minim posibil de întârzieri, însumări și multiplicări. Necesarul de astfel de componente este: N Blocuri de întârziere, N Sumatoare, $N+1$ Multiplicatoare și $N+1$ Coeficienți.

Prin sumatoare se înțeleg sumatoarele cu două intrări și prin urmare un nod de însumare cu S intrări necesită pentru implementare $S-1$ sumatoare.

Deoarece filtrele FIR cu fază liniară de ordinul N sunt definite de $[(N+1)/2]$ perechi de coeficienți egali sau complementari (plus coeficientul median în cazul N par), structura rețelei pentru forma directă poate fi implementată mai eficient. Pentru a exemplifica beneficiile simetriei coeficienților se consideră un filtru de tip II cu simetrie pară și N impar. Relația 5.31 se poate exprima ca suma a două componente astfel:

$$y[n] = \sum_{k=0}^{[(N-1)/2]} h[k]x[n-k] + \sum_{k=[(N+1)/2]}^N h[k]x[n-k] \quad (5.32)$$

Dacă se inversează ordinea de însumare și se decalează limitele însumării pentru termenul din dreapta se obține:

$$y[n] = \sum_{k=0}^{[(N-1)/2]} h[k]x[n-k] + \sum_{k=0}^{[(N-1)/2]} h[N-k]x[n-N+k] \quad (5.33)$$

Deoarece pentru filtrul cu fază liniară $h[k] = h[N-k]$ relația se poate simplifica în continuare și rezultă în final:

$$y[n] = \sum_{k=0}^{[(N-1)/2]} h[k](x[n-k] + x[n-N+k]) \quad (5.34)$$

Prin urmare dacă se însumează termenii $x[n-k]$ și $x[n-N+k]$ înainte de multiplicarea cu coeficienții corespunzători se vor efectua doar $[(N+1)/2]$ multiplicări și va fi necesar un nod de însumare cu numai $[(N+1)/2]$ pentru a genera semnalul de ieșire $y[n]$. În plus numărul coeficienților stocați se reduce la $[(N+1)/2]$. Structura rețelei modificate care implementează sistemul simplificat este dată în figura 5.15.

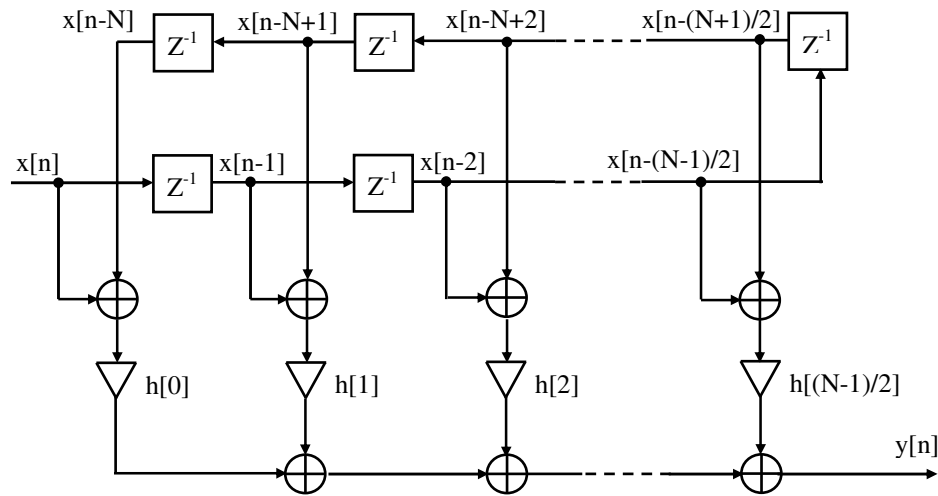


Figura 5.15 Structura rețelei pentru implementarea unui filtru FIR cu coeficienți simetrici

Numărul total de sumatoare cu două intrări se constituie din $[(N+1)/2]$ sumatoare cu care de obțin valorile de intrare pentru multiplicatoare plus $[(N-1)/2]$ sumatoare pentru nodul de însumare. Numărul componentelor necesare pentru implementarea filtrelor de tip I sau II este dat în tabelul de mai jos :

Componente	N impar	N par
Întâzieri	N	N
Sumatoare	N	N
Multiplicatoare	$(N+1)/2$	$N/2+1$
Coeficienți	$(N+1)/2$	$N/2+1$

5.4.2. EFECTUL CUANTIFICĂRII COEFICIENȚILOR

Ca în cazul tuturor aplicațiilor de procesare digitală a semnalelor, coeficienții $h[n]$ ai unui filtru FIR pot fi stocați cu o precizie limitată. Rotunjirea la pasul de cuantificare q conduce la apariția unei erori aditive $e[n]$ pentru fiecare coeficient. Astfel răspunsul la impuls al filtrului FIR cu coeficienți cuantificați se poate scrie:

$$\hat{h}[n] = h[n] + e[n] \quad (5.35)$$

unde $e[n]$ pentru orice n este un număr care satisface condiția $|e[n]| \leq q/2$. Dacă coeficienții $h[n]$ se cuantifică pe un număr B de biți pasul de cuantificare este $q = [\Delta/(2^B - 1)]$, unde Δ este domeniul de cuantificare maxim. În continuare se va considera $\Delta=1$ deoarece $|h[n]| \leq 1$ pentru orice filtru FIR cu câștig unitar. Prin urmare $q = [1/(2^B - 1)]$.

Considerând transformata z a relației 5.35 se obține:

$$\hat{H}(z) = \sum_{n=0}^N \hat{h}[n] z^{-n} = \sum_{n=0}^N h[n] z^{-n} + \sum_{n=0}^N e[n] z^{-n} = H(z) + E(z) \quad (5.36)$$

Prin urmare un filtru cu coeficienți cuantificați se poate reprezenta printr-o conexiune paralel între versiunea de precizie infinită a acestui filtru și un filtru cu funcția de transfer $E(z)$. Diagrama bloc a sistemului este dată în figura 5.16.

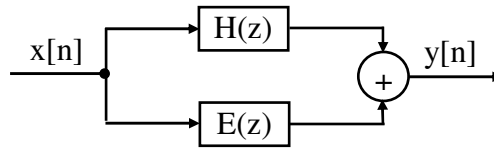


Figura 5.16 Diagrama bloc pentru un filtru FIR cu coeficienți cuantificați

Din considerentele de simetrie enunțate $\hat{h}[n]$ și $\hat{h}[N-n]$ corespund aceluiași coeficient. Ca urmare răspunsul la impuls al filtrului cu fază liniară și coeficienți cuantificați este simetric față de punctul median. Întrucât răspunsul ideal la impuls $h[n]$ este simetric $e[n]$ este de asemenea simetric și proprietatea de fază liniară nu este afectată de cuantificarea coeficienților.

Problema care se pune în continuare este modalitatea de a estima numărul minim de biți necesari pentru a asigura o atenuare minimă dorită pentru componentele din afara benzii utile.

Răspunsul în frecvență $\hat{H}[\omega]$ se obține substituind $z=e^{j\omega}$

$$\hat{H}(\omega) = \sum_{n=0}^N h[n] e^{-j\omega n} + \sum_{n=0}^N e[n] e^{-j\omega n} = H(\omega) + E(\omega) \quad (5.37)$$

Pentru a determina mărimea distorsiunii cauzate de $e[n]$ este necesar să analizăm eroarea de răspuns a filtrului $E(\omega)$. Întrucât valorile $e[n]$ sunt nepredictibile funcția $E(\omega)$ nu este deterministă și ca urmare se poate face doar o analiză calitativă. Deoarece $|e[n]| \leq q/2$ este posibil să se determine o valoare limită pentru $E(\omega)$, independentă de elementele $e[n]$, astfel:

$$|E(\omega)| = \left| \sum_{n=0}^N e[n] e^{-j\omega n} \right| \leq \sum_{n=0}^N |e[n]| |e^{-j\omega n}| \leq \frac{q}{2} (N+1) \quad (5.38)$$

Valoarea limită exprimată în dB se notează V_{limdB} și are expresia:

$$VlimdB = 20 \log_{10} \left(\frac{q}{2} (N+1) \right). \quad (5.39)$$

$VlimdB$ este o evaluare pesimistă deoarece a fost dedusă în ipoteza că toți termenii eroare au același semn și aceeași valoare $q/2$, situație a cărei probabilitate de realizare este foarte redusă. Datorită dificultății de a prezice natura erorilor de cuantificare o analiză statistică a efectului cuantificării coeficienților este mai potrivită. Modelul statistic utilizat presupune că erorile ce apar la cuantificarea coeficienților sunt independente și uniform distribuite în intervalul $-q/2, q/2$.

Funcția distribuție de probabilitate $f_e(e)$ are o valoare medie $\mu_e = 0$. Dispersia σ_e^2 pentru fiecare $e[n]$ se poate calcula astfel:

$$\sigma_e^2[n] = \int_{-\infty}^{\infty} e^2[n] f_e(e) de = \frac{1}{q} \int_{-q/2}^{q/2} e^2[n] de = \frac{q^2}{12} = \sigma_e^2$$

Pe baza raționamentului descris anterior obținem următoarea expresie pentru răspunsul filtrului la eroare $E(\omega)$:

$$E(\omega) = e^{-j\omega \frac{N}{2}} \left(e^{\left[\frac{N}{2} \right]} + \sum_{n=0}^{\frac{N-1}{2}} 2e[n] \cos \left(\left(\frac{N}{2} - n \right) \omega \right) \right) = e^{-j\omega \frac{N}{2}} 2R(\omega)$$

$E(\omega)$ conține un termen cu faza liniară și un termen real $R(\omega)$ care pentru orice ω este o combinație liniară de variabile aleatoare independente. Termenul cu fază liniară $e^{-j\omega N/2}$ reprezintă o întârziere cu un număr întreg de eșantioane care nu este afectată de cuantificare și ca urmare acest factor nu trebuie luat în considerare. Deoarece funcția densitate de probabilitate $f_e(e)$ pentru $e[n]$ este nulă cu excepția unui interval finit suma din factorul $R(\omega)$ satisface condiția Lindeberg a teoremei limitei centrale [Oppenheim99]. Astfel pentru N suficient de mare probabilitatea pentru $R(\omega)$ converge spre o distribuție Gaussiană. Datorită acestei tendințe a erorilor în domeniul frecvență media și dispersia dau o descriere bună a comportării statistice. Valoarea medie $\mu(\omega)$ și dispersia $\sigma(\omega)$ se calculează după cum urmează:

$$\begin{aligned} \mu(\omega) &= \left(\mu_e \left[\frac{N}{2} \right] + \sum_{n=0}^{\frac{N-1}{2}} 2\mu_e[n] \cos \left(\left(\frac{N}{2} - n \right) \omega \right) \right) = 0 \\ \sigma^2(\omega) &= \left(\sigma_e^2 \left[\frac{N}{2} \right] + \sum_{n=0}^{\frac{N-1}{2}} 2\sigma_e^2[n] \cos^2 \left(\left(\frac{N}{2} - n \right) \omega \right) \right) = \sigma_e^2 \left(N + \frac{\sin((N+1)\omega)}{\sin(\omega)} \right) \end{aligned} \quad (5.40)$$

Se introduce valoarea $\sigma_e^2 = \frac{q^2}{12}$, se normalizează expresia din paranteză și rezultă:

$$\sigma^2(\omega) = \frac{q^2}{12} (2N+1) \frac{1}{2N+1} \left(N + \frac{\sin((N+1)\omega)}{\sin(\omega)} \right) \quad (5.41)$$

Se obține următoarea expresie pentru deviația standard:

$$\sigma(\omega) = \frac{q}{2} \sqrt{\frac{(2N+1)}{3}} \sqrt{\frac{1}{2N+1} \left(N + \frac{\sin((N+1)\omega)}{\sin(\omega)} \right)} = \sigma_{max} \cdot W(\omega) \quad (5.42)$$

$W(\omega)$ este un termen pondere normalizat dependent doar de frecvență nu și de valoarea pasului de cuantificare. În domeniul $0 < \omega < \pi$, $\lim_{N \rightarrow \infty} W(\omega) = \frac{1}{\sqrt{2}}$, iar pentru $\omega = 0, \pm\pi, \pm2\pi, \dots$,

$W(\omega) = 1$, pentru orice N .

Pe baza acestor observații se poate scrie inegalitatea:

$$\sigma(\omega) \leq \sigma_{\max} = \frac{q}{2} \sqrt{\frac{(2N+1)}{3}} \quad (5.43)$$

care arată că $\sigma(\omega)$ se poate descrie prin valoarea $\sigma_{\max}$ care reprezintă limita minimă superioară.

Pe considerentul că probabilitatea ca o variabilă aleatoare Gaussiană să fie mai mică sau egală cu de două ori deviația sa standard este 0.954 iar răspunsul la eroare $E(\omega)$ are o distribuție Gaussiană se poate afirma cu probabilitate ridicată că acesta este limitat la de două ori valoarea deviației standard.

$$|E(\omega)| \leq 2\sigma_{\max} = q \sqrt{\frac{(2N+1)}{3}} \quad (5.44)$$

Exprimată în dB valoarea limită are expresia:

$$V \lim dB = 20 \log_{10} \left(q \sqrt{\frac{2N+1}{3}} \right). \quad (5.45)$$

Pentru a investiga distorsiunile răspunsului în frecvență introduse de cuantificarea coeficienților se consideră funcția ideală $L(\omega)$ ce urmează a fi aproximată de răspunsul în frecvență $H(\omega)$, al filtrului FIR cu faza liniară. Specificația de proiectare se definește printr-un set de benzi de frecvență $\Omega_k \subset [0, \pi]$, $k=1, \dots, M$ și setul corespunzător de valori limită $\delta_k > 0$ astfel încât pentru fiecare valoare a lui k se aproximează $L(\omega)$ prin $H(\omega)$ cu o eroare δ_k pentru toate frecvențele $\omega \in \Omega_k$.

$$\max_{\omega \in \Omega_k} |H(\omega) - L(\omega)| = \delta_k \quad k=1, \dots, M \quad (5.46)$$

Benzile de frecvență sunt separate de benzi de tranziție unde nu se impun constrângeri pentru răspunsul în frecvență. Cuantificarea coeficienților filtrului proiectat pentru a satisface această specificație poate conduce la creșterea erorii dincolo de valorile limită impuse. Prin urmare noua valoare limită superioară pentru orice ω este:

$$|\hat{H}(\omega) - L(\omega)| \leq |\hat{H}(\omega) - H(\omega)| + |H(\omega) - L(\omega)| \leq E(\omega) + |H(\omega) - L(\omega)| \quad (5.47)$$

Pentru setul de benzi de frecvență Ω_k care prezintă interes se obține:

$$|\hat{H}(\omega) - L(\omega)| \leq \max_{\omega \in \Omega_k} |E(\omega)| + \delta_k \leq q \cdot \sqrt{\frac{2N+1}{3}} + \delta_k \leq \frac{1}{2^B - 1} \cdot \sqrt{\frac{2N+1}{3}} + \delta_k = \varepsilon_k,$$

pentru $\omega \in \Omega_k$

Relația arată că în banda Ω_k se poate aproxima $L(\omega)$ prin răspunsul în frecvență al unui filtru FIR cu coeficienții cuantificați pe B biți, cu o eroare mai mică decât ε_k , dacă răspunsul în frecvență al filtrului ideal poate fi proiectat astfel ca pentru $\omega \in \Omega_k$

$$|H(\omega) - L(\omega)| \leq \delta_k \quad (5.48)$$

Dacă se definește atenuarea în bandă pentru filtrul ideal în decibeli

$$A_k = -20 \log_{10}(\delta_k) \quad (5.49)$$

atunci limita inferioară pentru atenuarea în bandă la filtrul cu coeficienți cuantificați $A_k \lim$ este:

$$A_k \lim \geq -20 \log_{10} \left(10^{-[(A_k / 20)]} + \frac{1}{2^B - 1} \sqrt{\frac{2N + 1}{3}} \right) \quad (5.50)$$

În procesul de proiectare al filtrului este mai util să se cunoască numărul minim de biți necesar pentru o deviație maximă de la atenuarea inițială în bandă, decât să se determine care este valoarea maximă a atenuării ce se poate obține pentru un număr dat de biți. Se definește Δ_k , deviația maximă a atenuării în bandă care se poate obține pentru un număr dat de biți, astfel: $A_k \lim \geq A_k - \Delta_k$, și înlocuind se obține:

$$B(N, A_k, \Delta_k) \geq \log_2 \left(1 + \frac{10^{[(A_k / 20)]}}{10^{[(\Delta_k / 20)]} - 1} \sqrt{\frac{2N + 1}{3}} \right) \quad (5.51)$$

Există situații când numărul maxim de biți utilizabil pentru cuantificarea coeficienților este limitat prin resursele hardware disponibile - procesoare de semnal în virgulă fixă cu magistrala de date cu lățimea fixată. În aceste situații este utilă o formulă care oferă posibilitatea de a investiga reducerea maximă a atenuării în bandă datorată cuantificării pe un număr de biți. Formula se obține din (5.50)

$$A_k - A_k \lim \leq 20 \log_{10} \left(1 + \frac{10^{[(A_k / 20)]}}{2^B - 1} \sqrt{\frac{2N + 1}{3}} \right) \quad (5.52)$$

Această relație permite proiectantului să estimeze influența implementării filtrului pe un suport hardware neideal și să ia măsurile necesare dacă filtru cuantificat nu se încadrează în specificația dată. Parametrii pentru relația (5.52) sunt ordinul filtrului N și atenuarea inițială în bandă A_k .

5.4.3. PROIECTAREA ÎN MATLAB A FILTRULUI DECIMATOR

Procedurile de proiectare a filtrelor digitale sunt algoritmice și se bazează pe folosirea de programe dedicate pentru rezolvarea unor seturi de ecuații liniare sau neliniare. Pachetul "Signal Processing Toolbox" din Matlab conține o colecție de programe pentru toate tipurile de aplicații de procesare digitală de semnal inclusiv pentru proiectarea filtrelor digitale decimatoare.

Pentru ca filtrul să poată fi proiectat trebuie cunoscute mai întâi specificațiile dorite pentru acesta. În mod normal specificațiile depind atât de blocurile de procesare care succed filtrul și de rata de decimare cât și de caracteristicile semnalului de intrare. În plus trebuie luate în considerare limitările hardware pentru implementare respectiv numărul de celule (prize) întrucât dimensiunea fizică a filtrului crește cu ordinul acestuia și de asemenea numărul maxim de biți disponibil pentru stocarea coeficienților în memorie.

Specificația filtrului

Specificația filtrului trece jos se definește prin banda de trecere și nivelul undulațiilor permise în banda de trecere, respectiv prin banda de blocare și atenuarea dorită pentru componentele spectrale din această bandă

Marginile benzii de trecere și de oprire sunt date de frecvența maximă a semnalului de intrare și de frecvența imagine a semnalului decimat. Pentru aria de traductoare considerată având frecvența centrală de 3.5Mhz și o bandă de 50% se obține frecvența maximă a semnalului de intrare de 4.375Mhz. În proiectare se va folosi valoarea de 4.5Mhz pentru frecvența maximă a semnalului de intrare și se fixează limitele benzii de tranziție la $f_c = 4.5\text{Mhz}$ și respectiv $f_r = 9\text{Mhz}$. Pentru a asigura amplitudini mai mici de -80 dB pentru toate componentele spectrale de frecvențe mai mari de 10Mhz nivelul atenuării în banda de blocare (oprire) se fixează la $\delta_p = -80\text{ dB}$. O ultimă constrângere în ceea ce privește răspunsul în frecvență al filtrului este eroarea maximă admisibilă în banda de trecere dată de nivelul undulațiilor în bandă. Pentru aplicațiile de formare de fascicul câștigul în banda de trecere nu trebuie să fluctueze cu mai mult de $\delta_s = \pm 0,1\text{dB}$. Pe seama cerințelor prezentate tabelul de mai jos prezintă o primă specificație a filtrului.

Frecvența de tăiere	$f_c = 4,5\text{Mhz}$
Frecvența de oprire	$f_r = 9\text{Mhz}$
Nivelul undulațiilor în banda de trecere	$\delta_p = \pm 0,1\text{dB}$.
Atenuarea în banda de oprire	$\delta_s = -80\text{ dB}$.

Proiectarea filtrului

Proiectarea unui filtru digital cu ajutorul unui program dedicat este în mod obișnuit un proces iterativ. Specificația furnizează setul de parametri pentru algoritmul de proiectare. Întrucât filtrul ideal poate fi doar aproximat trebuie să avem în vedere anumite compromisuri și ajustarea specificației pentru a se obține performanțele dorite.

Determinarea ordinului filtrului

Pentru a calcula coeficienții filtrului folosind algoritmul Remez este necesar să fie cunoscut ordinul filtrului N . Comanda $[N, F0, A0, W] = \text{remezord}(F, A, DEV, fs)$ din pachetul "Signal Processing Toolbox" calculează ordinul N , valorile normalizate pentru limitele benzilor $F0$, atenuările în banda de frecvență $A0$ și ponderile W care vor fi utilizate de funcția *remez*. Filtrul rezultat satisface cu aproximație specificațiile date prin parametri de intrare F, A, DEV și fs . F este vectorul marginilor benzilor de frecvență în ordine crescătoare între 0 și frecvența de eșantionare fs . A este un vector care specifică amplitudinilor dorite în benzile definite în F , iar vectorul DEV conține nivelul maxim admis pentru undulații în fiecare bandă.

$$F = \begin{bmatrix} f_c \\ f_s \end{bmatrix}, A = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, DEV = \begin{bmatrix} 10^{\delta_p/20} - 1 \\ 10^{\delta_s/20} \end{bmatrix},$$

Pentru valorile date în tabelul de mai sus funcția *remezord* returnează o valoare $N=110$. Așa cum se va vedea ulterior este benefic să se construiască filtre FIR decimatoare cu 2^n coeficienți unde n este un număr întreg. Următorul filtru de ordin mai mare pentru care este adevărată această cerință are 128 coeficienți și prin urmare ordinul $N = 127$. Evident în această situație cerințele din specificație vor fi cu atât mai mult îndeplinite în condițiile în care structura filtrului nu devine mult mai complexă. Alegerea unui ordin mai mare decât este necesar lasă posibilitatea de a îmbunătăți caracteristicile filtrului. Ordinul minim necesar pentru a realiza o caracteristică de filtrare oarecare este în general determinat de lățimea relativă a benzii de tranziție raportată la frecvența de eșantionare și de eroarea maximă admisibilă în benzile de frecvență. Pentru aplicația dată undulațiile definite anterior sunt suficiente dar prin reducerea lățimii benzii de tranziție se pot îmbunătăți semnificativ performanțele filtrului. Din spectrul semnalului de la ieșirea modulatorului delta-sigma se

observă că pentru frecvențele imediat superioare frecvenței de tăiere f_c puterea zgomotului de cuantificare crește semnificativ. Prin îngustarea benzii de tranziție se îmbunătățește rejectia zgomotului între f_c și f_s . Figura 5.17 redă dependența ordinului minim estimat de lățimea benzii de tranziție

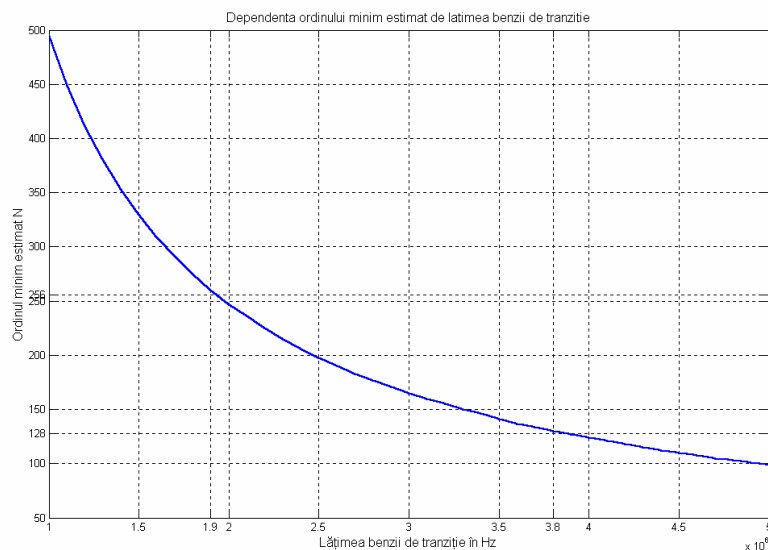


Figura 5.17 Dependența ordinului minim de lățimea benzii de tranziție

Se observă că pentru $N = 128$ banda de tranziție minimă este de aproximativ 3,8 Mhz, iar pentru $N = 256$ se obține o banda de tranziție minimă de aproximativ 1,9 Mhz. În concordanță cu aceste rezultate se lucrează cu $N = 255$ și o lățime a benzii de tranziție de 2Mhz. Rezultă noua specificație pentru filtru :

Frecvența de tăiere	$f_c = 4,5\text{Mhz}$
Frecvența de oprire	$f_r = 6,5\text{Mhz}$
Nivelul undulațiilor în banda de trecere	$\delta_p = \pm 0,1\text{dB}$
Atenuarea în banda de oprire	$\delta_s = -80\text{ dB.}$
Ordinul filtrului	$N = 255$

Ordinul minim estimat pentru această specificație este $N = 247$.

Calculul coeficienților filtrului

Comanda MATLAB: $h = \text{remez}(N, F0, A0, W)$ întoarce coeficienții unui filtru FIR cu fază liniară și de lungime $N+1$ care este cea mai bună aproximare a răspunsului dorit în frecvență în sensul minimizării erorii maxime

Răspunsul în frecvență se poate calcula prin transformarea Fourier a lui h . Figura 5.19 arată pe răspunsul în frecvență al filtrului definirea benzilor de trecere, tranziție și oprire și nivelul de atenuare în bandă.

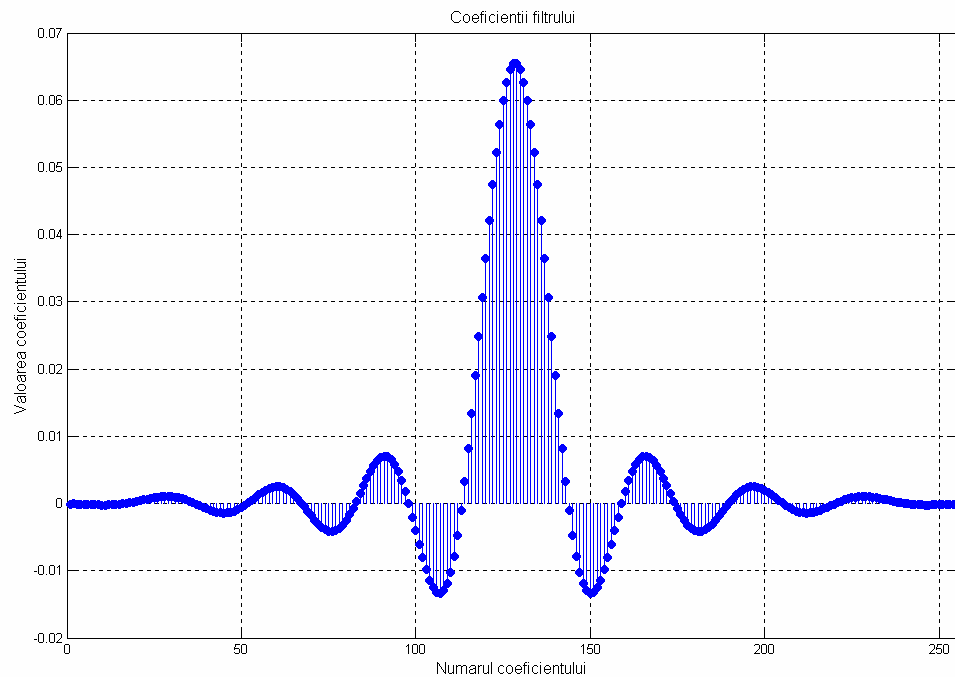


Figura 5.18 Coeficienții filtrului

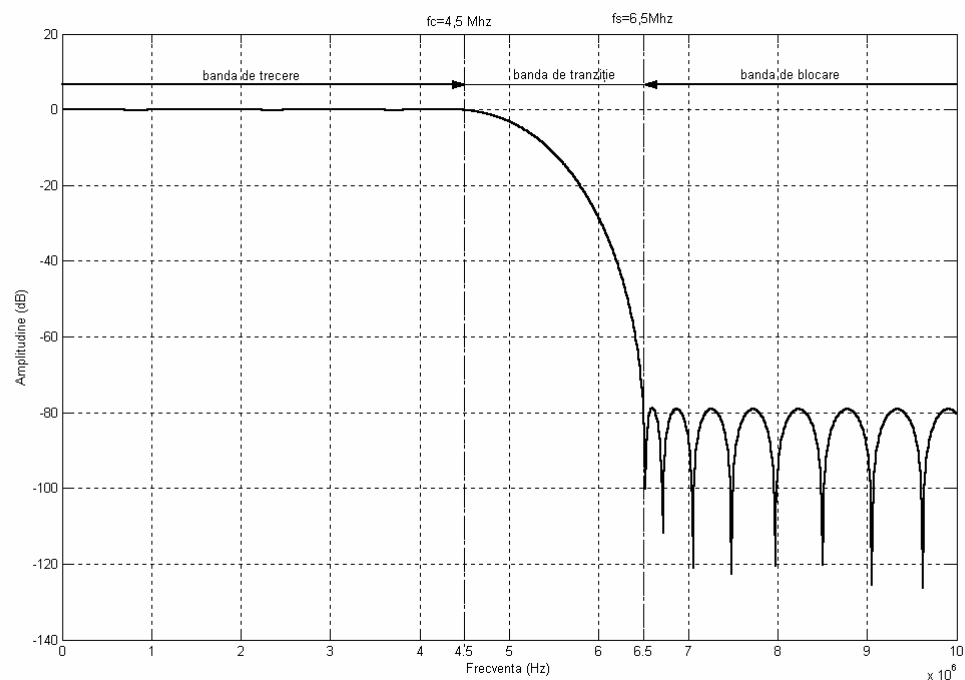


Figura 5.19 Filtru FIR cu $N=256$ – Răspunsul în frecvență

După ce coeficienții filtrului au fost determinați, performanța acestuia poate fi verificată prin simulare folosind comanda $y = \text{filter}(a, b, x)$, care filtrează datele din vectorul x cu filtrul descris de vectorii a și b . În cazul dat pentru filtrul FIR vectorul b care corespunde numitorului funcției de transfer este egal cu 1 iar vectorul a conține coeficienții determinați anterior. Evidențierea performanței rezultă din compararea spectrului semnalului de intrare cu a semnalului obținut la ieșirea filtrului. Figura 5.20 indică spectrele de intrare și ieșire pentru un semnal sinusoidal de 3,5Mhz.

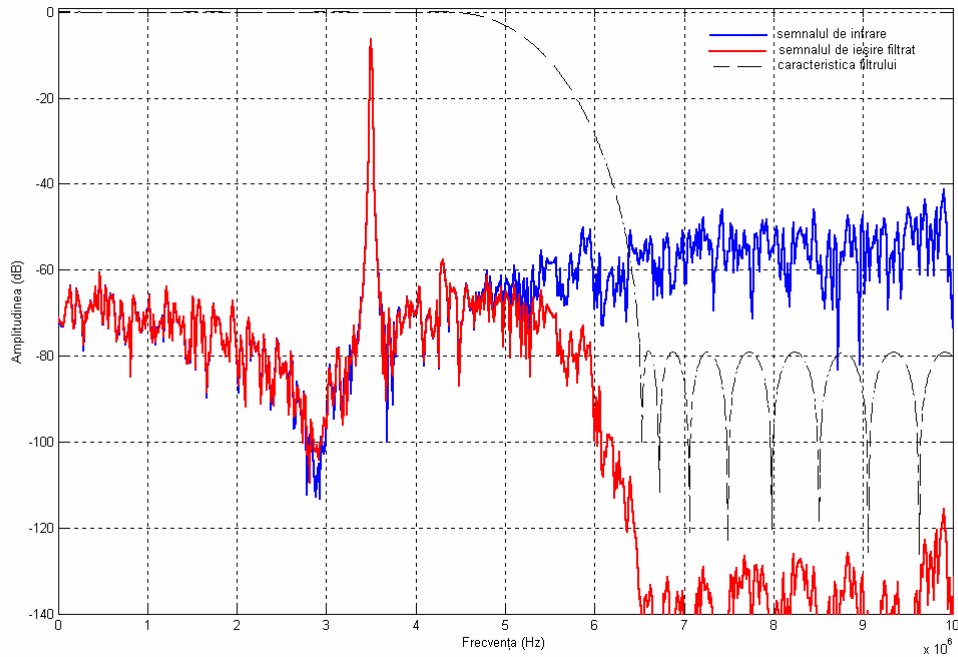


Figura 5.20 Comparație între semnalul de intrare și semnalul de ieșire filtrat

Cuantificarea coeficienților

Așa cum s-a arătat în paragraful 5.4.2 cuantificarea coeficienților filtrului determină apariția unei erori față de răspunsul filtrului necuantificat în special în banda de frecvență cu atenuări mari. În cazul dat pentru o atenuare în banda de blocare de 80dB și o variație maximă a atenuării de 3dB din relația (5.51) se obține numărul minim necesar de biți:

$$B(N, A_k, \Delta_k) = \log_2 \left(1 + \frac{10^{[(A_k/20)]}}{10^{[(\Delta_k/20)]} - 1} \sqrt{\frac{2N+1}{3}} \right) = 20.0287$$

Prin urmare coeficienții vor fi cuantificați pe 21 biți. Figura 5.21 arată influența rezoluției de cuantificare asupra atenuării în banda de blocare. Se remarcă alterarea valorii dorite pentru atenuare cu 10 dB la cuantificarea pe 15 biți și cu aproximativ 24 dB la cuantificarea pe 13 biți. Valoarea minimă a atenuării pentru $B = 21$ biți rezultă din relația 5.52 și este

$$A_k \text{ lim} = -20 \log_{10} \left(10^{-(A_k/20)} + \frac{1}{2^B - 1} \sqrt{\frac{2N+1}{3}} \right) = 79.7338 \text{ dB}$$

5.4.4. CONCEPTUL HARDWARE

Din punct de vedere practic proiectarea unui filtru digital presupune pe lângă calculul coeficienților și elaborarea unui concept hardware implementabil. În continuare se studiază modalitatea de implementare a operațiilor matematice ce descriu filtrul FIR într-o arhitectură Altera de tip FLEX (APEX).

Multiplicarea și adunarea pot fi optimizate prin folosirea tabelor de căutare (Look-Up Table - LUT) din structura dispozitivelor FLEX (APEX).

Ecuția 5.34 care descrie fitrul poate fi rescrisă sub forma următoare:

$$y[n] = \sum_{k=0}^{[(N-1)/2]} h[k](x[n-k] + x[n-N+k]) = \sum_{k=0}^{[(N-1)/2]} h[k] \cdot s[k], \text{ unde } s[k] = x[n-k] + x[n-N+k]$$

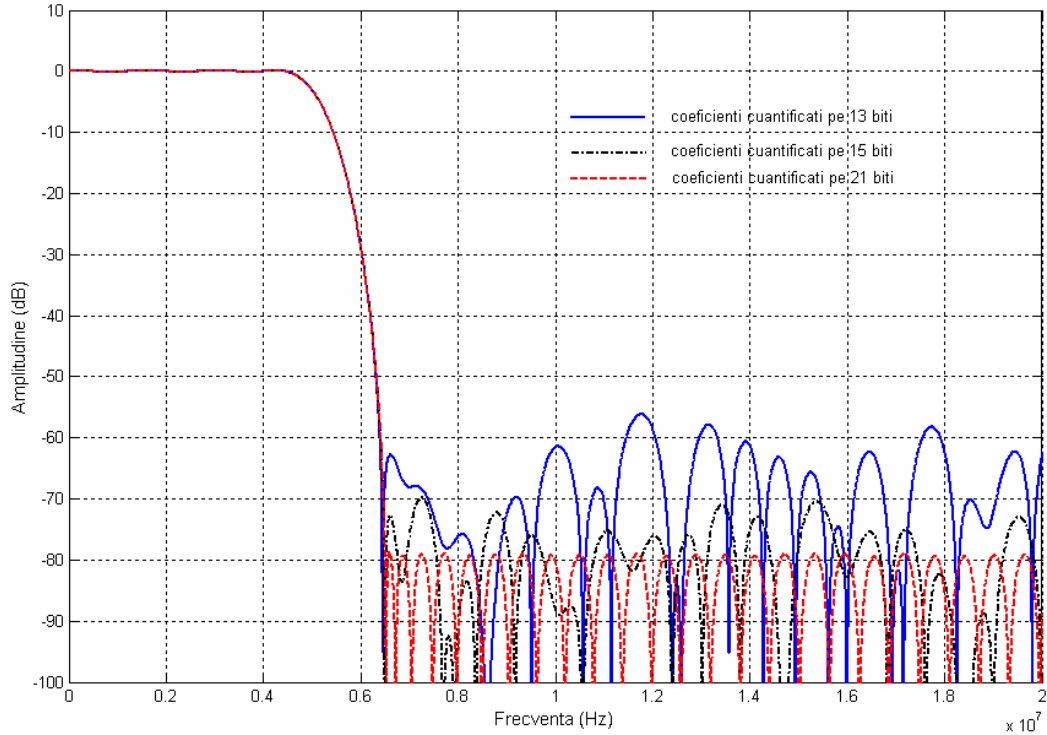


Figura 5.21 Influența numărului de biți asupra atenuării în banda de blocare

În primă instanță se observă că este necesar calculul în paralel al termenilor produs de forma $h[k]s[k]$ și însumarea rezultatelor.

Conform algoritmului de înmulțire termenii de forma $h[k]s[k]$ se calculează ca sumă de produse parțiale multiplicare cu puterile lui 2.

$$h[k]s[k] = 2^0 h[k]s[k]_0 + 2^1 h[k]s[k]_1 + 2^2 h[k]s[k]_2 + \dots + 2^{n_s-1} h[k]s[k]_{n_s-1}$$

n_s – este numărul de biți pe care se reprezintă semnalul de intrare

$s[k]_0$ – este bitul cel mai puțin semnificativ

$s[k]_{n_s-1}$ – este bitul cel mai semnificativ

Pe baza acestei observații ecuația de mai sus se scrie sub forma:

$$\begin{aligned} y[n] = & 2^0 \{h[0] \cdot s[0]_0 + h[1] \cdot s[1]_0 + h[2] \cdot s[2]_0 + h[3] \cdot s[3]_0 + \dots + h[(N-1)/2] \cdot s[(N-1)/2]_0\} + \\ & 2^1 \{h[0] \cdot s[0]_1 + h[1] \cdot s[1]_1 + h[2] \cdot s[2]_1 + h[3] \cdot s[3]_1 + \dots + h[(N-1)/2] \cdot s[(N-1)/2]_1\} + \\ & 2^2 \{h[0] \cdot s[0]_2 + h[1] \cdot s[1]_2 + h[2] \cdot s[2]_2 + h[3] \cdot s[3]_2 + \dots + h[(N-1)/2] \cdot s[(N-1)/2]_2\} + \\ & \dots + \\ & 2^{n_s-1} \{h[0] \cdot s[0]_{n_s-1} + h[1] \cdot s[1]_{n_s-1} + h[2] \cdot s[2]_{n_s-1} + h[3] \cdot s[3]_{n_s-1} + \dots + h[(N-1)/2] \cdot s[(N-1)/2]_{n_s-1}\} \\ = & 2^0 \cdot P_{p0} + 2^1 \cdot P_{p1} + 2^2 \cdot P_{p2} + 2^3 \cdot P_{p3} + \dots + 2^{n_s-1} \cdot P_{p_{n_s-1}} \end{aligned}$$

Întrucât coeficienții $h[k]$ au o valoare fixată rezultatul unui produs parțial P_{pk} se obține accesând un LUT înscris cu toate combinațiile posibile pentru esantioanele de la intrare. Spre exemplificare tabelul prezintă o cazul unui filtru de ordinul $N = 8$.

$s[0]_0$ $s[1]_0$ $s[2]_0$ $s[3]_0$	P_{p0}
0000	0
0001	$h[0]$
0010	$h[1]$
0011	$h[1] + h[0]$
0100	$h[2]$

0101	$h[2] + h[0]$
0110	$h[2] + h[1]$
0111	$h[2] + h[1] + h[0]$
1000	$h[3]$
1001	$h[3] + h[0]$
1010	$h[3] + h[1]$
1011	$h[3] + h[1] + h[0]$
1100	$h[3] + h[2]$
1101	$h[3] + h[2] + h[0]$
1110	$h[3] + h[2] + h[1]$
1111	$h[3] + h[2] + h[1] + h[0]$

Dimensiunea tabelii de căutare este dată de ordinul filtrului și de numărul de biți n_c pe care se reprezintă coeficienții filtrului. Dacă $N = 8$ și $n_c = 2$ pentru implementarea fiecărui produs parțial este necesar un LUT de dimensiune 16×4 care se construiește prin conectarea în paralel a generatoarelor de funcții (LUT-uri 16×1) din 4 elemente logice. Figura 5.22 arată structura multiplicatorului pentru un filtru simetric de ordinul 8 cu semnalul de intrare codificat pe 2 biți. Fiecare produs parțial se calculează în aceeași manieră iar multiplicarea cu 2^k revine la deplasarea rezultatului spre stânga cu k biți.

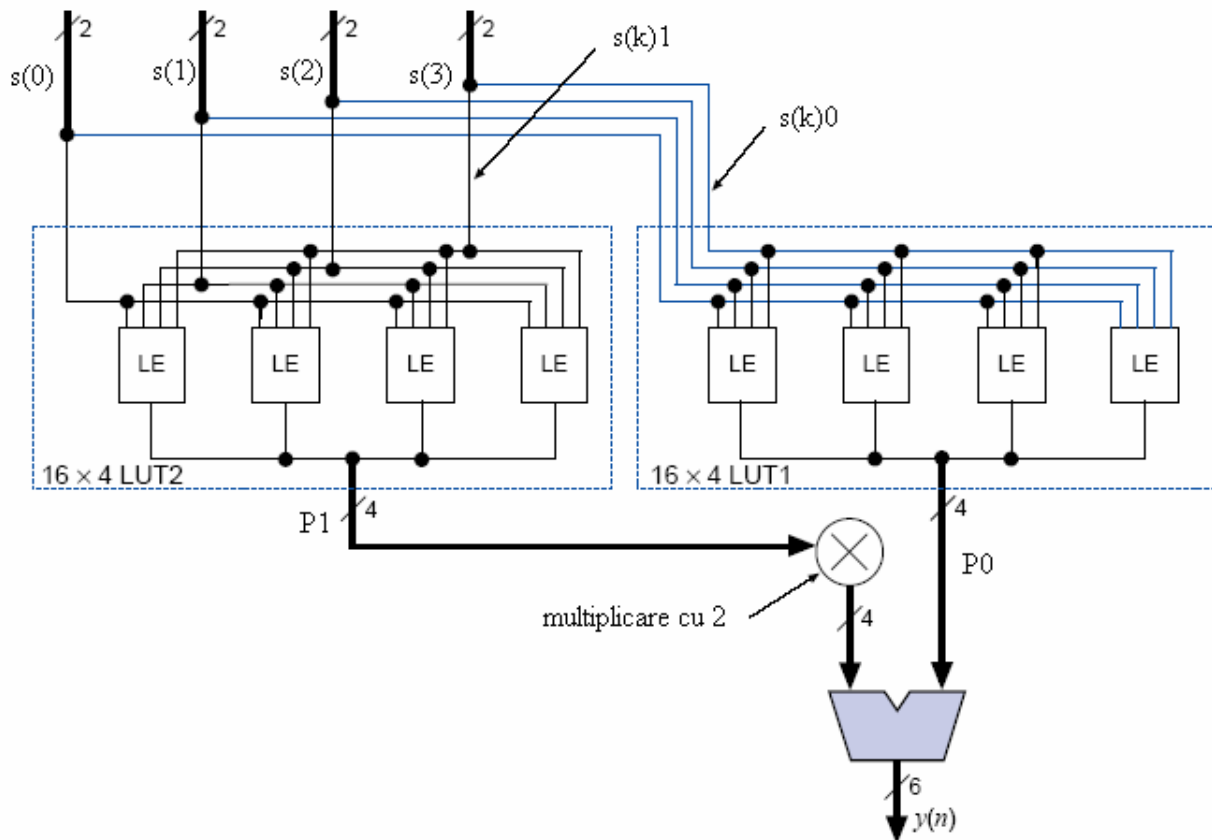


Figura 5.22 Exemplificarea implementării multiplicatorului pentru $N=8$ și $n_s = 2$

Generalizând pentru un filtru simetric de ordinul N cu coeficienții cuantificați pe n_c biți și un semnal de intrare cuantificat pe n_s biți un eșantion de la ieșire este rezultatul însumării a n_s produse parțiale și calculul fiecărui produs parțial revine la citirea unui LUT cu $N/2$ intrări și cuvinte de ieșire de $n_c + \log_2(N/2)$ biți.

Pentru filtre cu un N mare dimensiunile tabelii de căutare cresc foarte mult încât implementarea devine dificilă. În acest caz se apelează la arhitecturi scalare paralele optimizate pentru viteză în detrimentul resurselor utilizate. Astfel în cazul dat pentru un filtru

de ordinul $N = 128$, care operează cu semnale de intrare cuantificate pe $n_s = 8$ biți și are coeficienții cuantificați pe $n_c = 20$ biți se pot conecta în paralel 4 filtre de ordinul 16 sau 8 filtre de ordinul 8 cu coeficienți diferiți și se însumează ieșirile parțiale. (figura 5.23)

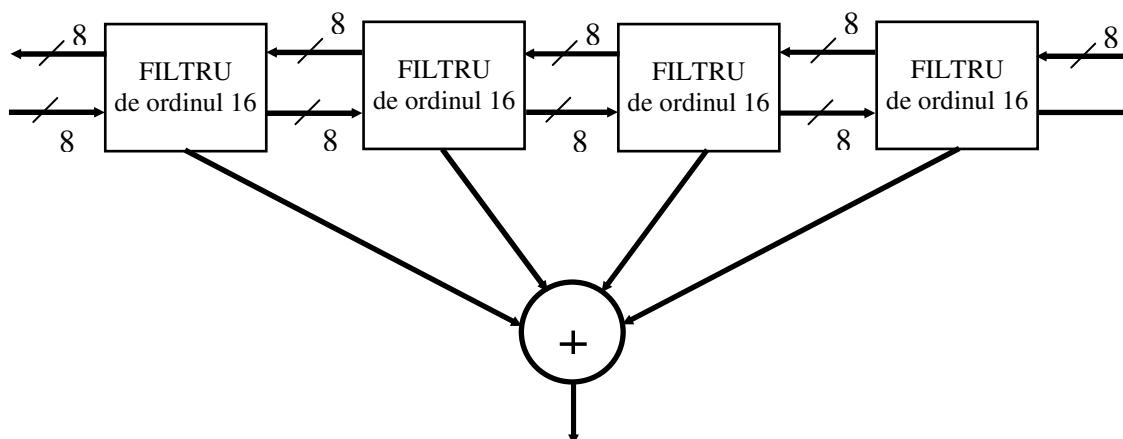


Figura 5.23 Exemplificarea implementării unui filtru simetric de ordinul 128

Un filtru decimator calculează fiecare cel de-al D eșantion unde D este factorul de decimare. Filtrul decimator poate fi realizat simplu prin eliminarea eșantioanelor ne semnificative de la ieșirea unui filtru obișnuit. Aceasta implementare nu este eficientă deoarece frecvența de procesare este dată de rata datelor de intrare iar în final cea mai mare parte a eșantioanelor calculate sunt aruncate. Într-o arhitectură optimizată frecvența de procesare este dată de rata datelor de ieșire fiind calculate doar eșantioanele necesare.

5.4.5 SIMULARE + SINTEZA FLEX10K

Pentru descrierea și verificarea prin simulare a filtrului decimator și obținerea în final a codului sintetizabil în dispozitivul FLEX 10K s-a folosit modulul de proiectare cu PLD-uri din ORCAD9.1 și mediul QUARTUS de la ALTERA.

Codul vhdL care descrie filtrul conține două entități: filtrul efectiv și un automat secvențial care comanda filtrul și realizează decimarea cu 16 a numărului de eșantioane. Figura 5.24 prezintă modalitatea de interconectare a celor două entități.

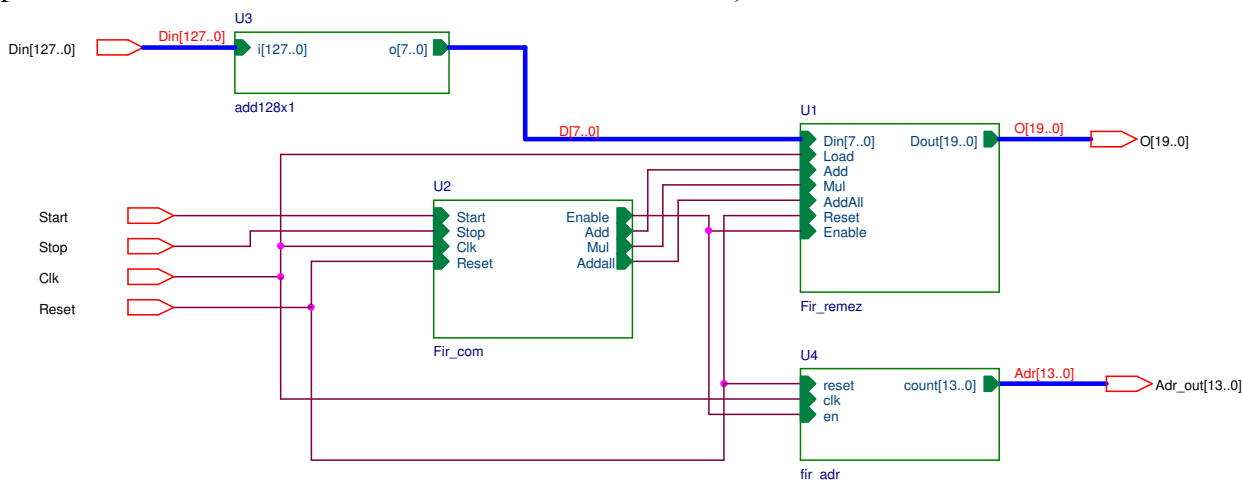


Figura 5.24 Interconectarea entităților filtrului decimator pentru implementare într-un dispozitiv FLEX10K

Numărul stărilor automatului secvențial este dat de factorul de decimare în cazul dat fiind 16. În succesiunea stărilor, între stările care activează comenzile pentru execuția operațiilor de adunare a eșantioanelor pentru coeficienții simetrici (Add), înmulțire (Mul) și adunare

produse parțiale (Addall) sunt intercalate stări de întârziere (del) cerute de operația de decimare fapt care relaxează cerințele privind viteza de execuție a operațiilor aritmetice. Semnalul de comandă pentru încărcarea datelor de intrare este activ în permanentă și asigură încărcarea registrului de deplasare cu frecvența semnalului de intrare

Metoda de testare a funcționării filtrului implementat în dispozitivul ALTERA:

- s-a folosit ca semnal de test semnalul recepționat de un element de arie pentru configurația descrisă în referatul 2.
- cu ajutorul unei rutine Matlab s-a realizat conversia sigma delta a semnalului de intrare.
- Semnalul convertit pe un bit s-a utilizat ca semnal de test pentru filtrul implementat FLEX10K. S-a folosit pentru aceasta modulul de proiectare PLD din Orcad9 și mediul Quartus.
- Ieșirea filtrului a fost preluată în Matlab pentru analiza comparativă în domeniul timp și frecvență a semnalului original cu cel filtrat.

Modificarea formatului datelor de la ieșirea unuia dintre programele folosite (Matlab sau Orcad) pentru a deveni compatibile cu intrarea celuilalt (Orcad sau Matlab) s-a realizat cu ajutorul unor programe de conversie scrise în limbajul C.

Figurile 5.25-26 prezintă spectrul de frecvență pentru semnalul original, codat pe 1 bit, de la ieșirea modulatorului sigma delta, pentru semnalul de la ieșirea filtrului fără decimare și respectiv pentru semnalul filtrat și decimat, ambele filtre fiind implementate în FPGA. Reprezentarea în domeniul timp pentru semnalele folosite la testare este data în figurile 5.27 și 5.28

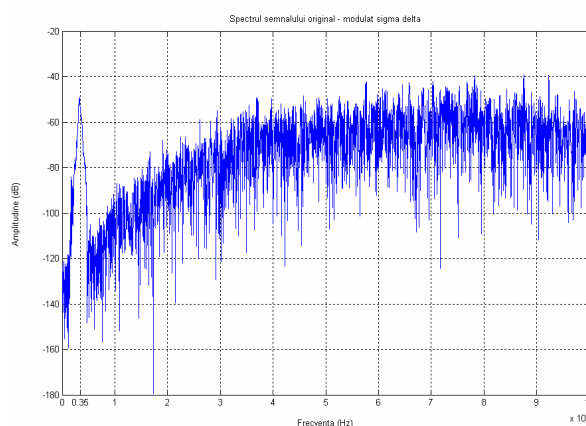


Figura 5.25 Spectrul semnalului original – modulat sigma delta

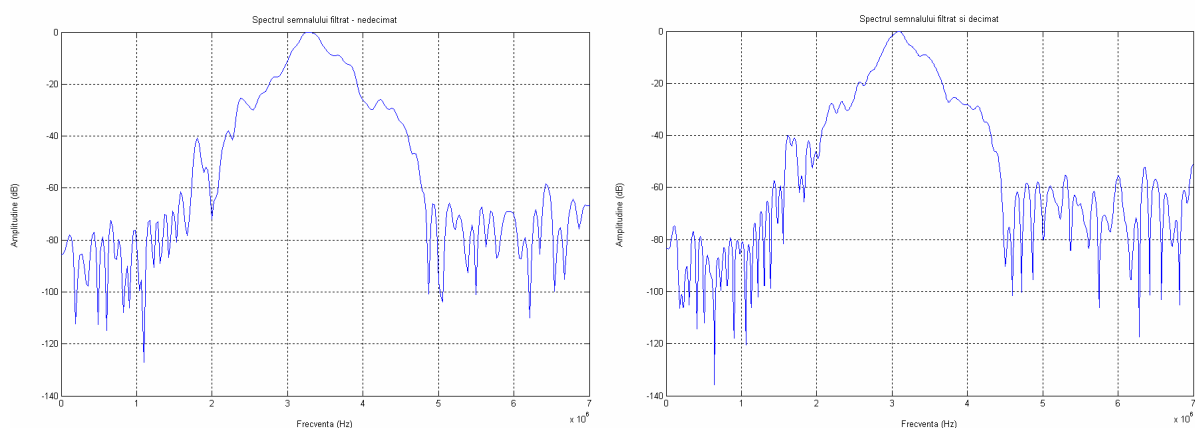


Figura 5.26 Spectrul semnalului filtrat nedecimat (stânga), Spectrul semnalului filtrat decimat (dreapta)

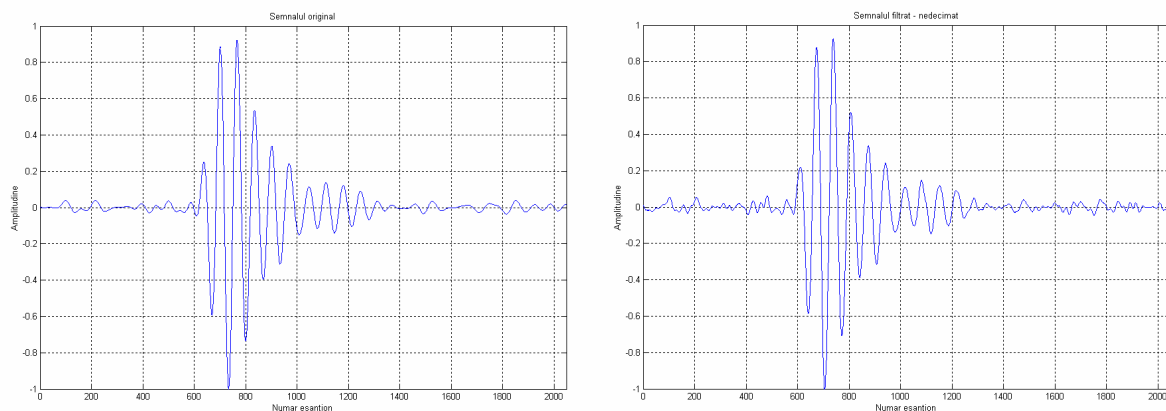


Figura 5.27 Semnalul de intrare interpolat (stânga), Semnalul filtrat nedecimat (dreapta)

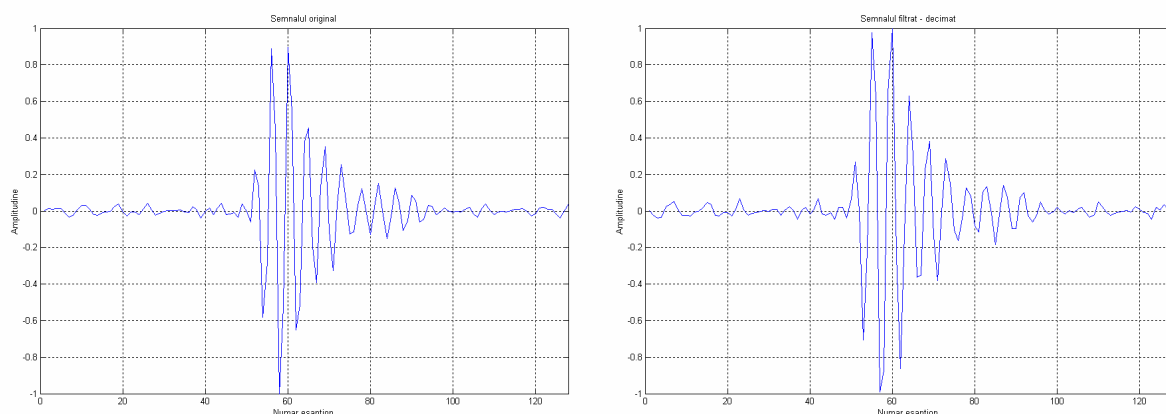


Figura 5.28 Semnalul de intrare neinterpolat (stânga), Semnalul filtrat decimat (dreapta)

5.5 IMPLEMENTAREA UNUI FORMATOR CU 128 CANALE

Pe baza rezultatelor obtinute la implementările parțiale în dispozitive din familia Flex10k s-a reusit implementarea completă a secțiunii digitale pentru un formator cu 128 de canale într-un dispozitiv din familia Stratix. Pentru a optimiza structura memoriei s-a ales un dispozitiv programabil care contine memoria RAM organizată pe blocuri de 4K. Arhitectura secțiunii digitale este dată în figura 5.29.

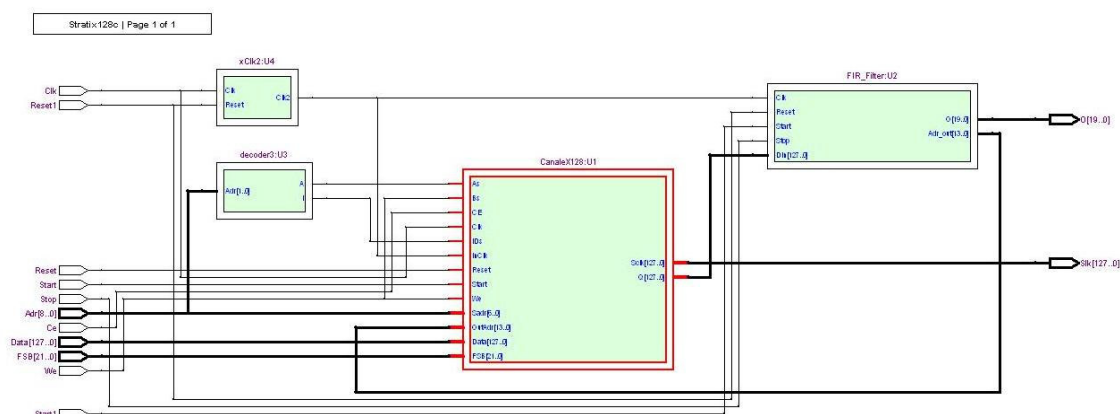


Figura 5.29 Secțiunea digitală pentru formatorul cu 128 canale

Utilizarea dispozitivului indicată în tabelul 5.5 sugerează că dispozitivul ales conține suficiente resurse pentru a extinde numărul canalelor procesabile la 256.

Tabelul 5.5: Utilizarea dispozitivului EP2S180F1508C4

Fitter Status	Successful - Wed Jul 20 2005
Quartus II Version	4.1 Build 181 06/29/2004 SJ
Top-level Entity Name	Stratix128c
Family	Stratix II
Device	EP2S180F1508C4
Total ALUTs	36,479 / 143,520 (25 %)
Total pins	315 / 1,171 (26 %)
Total memory bits	2,097,152 / 9,383,040 (22 %)

Anexa 2 contine codul sursa VHDL corespunzător arhitecturii formatorului cu 128 canale.

5.6 REZULTATE EXPERIMENTALE

Verificarea practică a funcționării structurii propuse s-a realizat pe un stant construit în jurul plăcii de dezvoltare pentru aplicații de laborator UP1.(figura 5.30). Standul mai cuprinde o sursa de alimentare de laborator I 4103, osciloscopul Hameg 407-2 si un PC pe care au fost instalate mediile de programare pentru configurarea FPGA-ului si pentru preluarea datelor de la osciloscop – figura 5.31

Comunicatia cu placa de dezvoltare se realizează prin intermediul unei interfețe JTAG comandată de modulul de programare din Mediul software pentru dezvoltarea aplicațiilor MAX+PLUS II. Pentru preluarea datelor achiziționate de osciloscop am folosit programul SP107 care autorizează controlul osciloscopului, captura de semnal si transferul în memoria PC printr-o interfață seriala RS232.

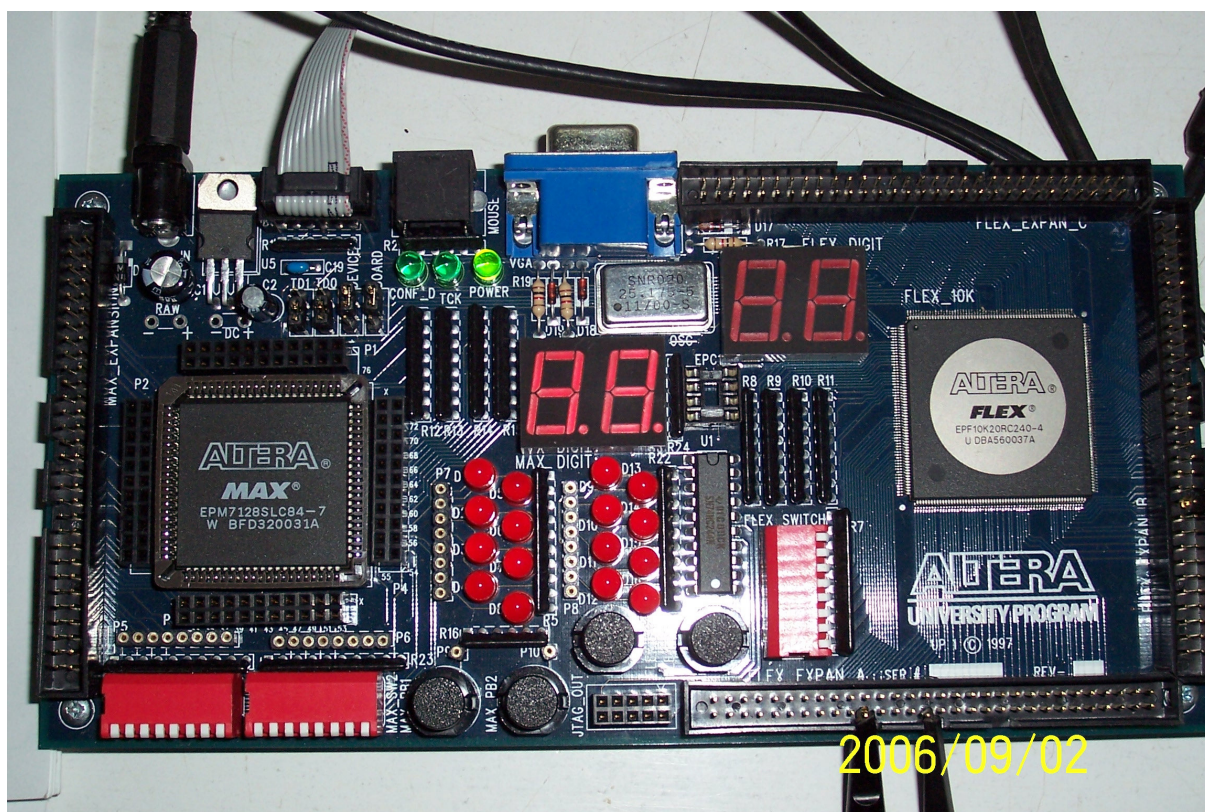


Fig. 5.30 Placa de dezvoltare UP1

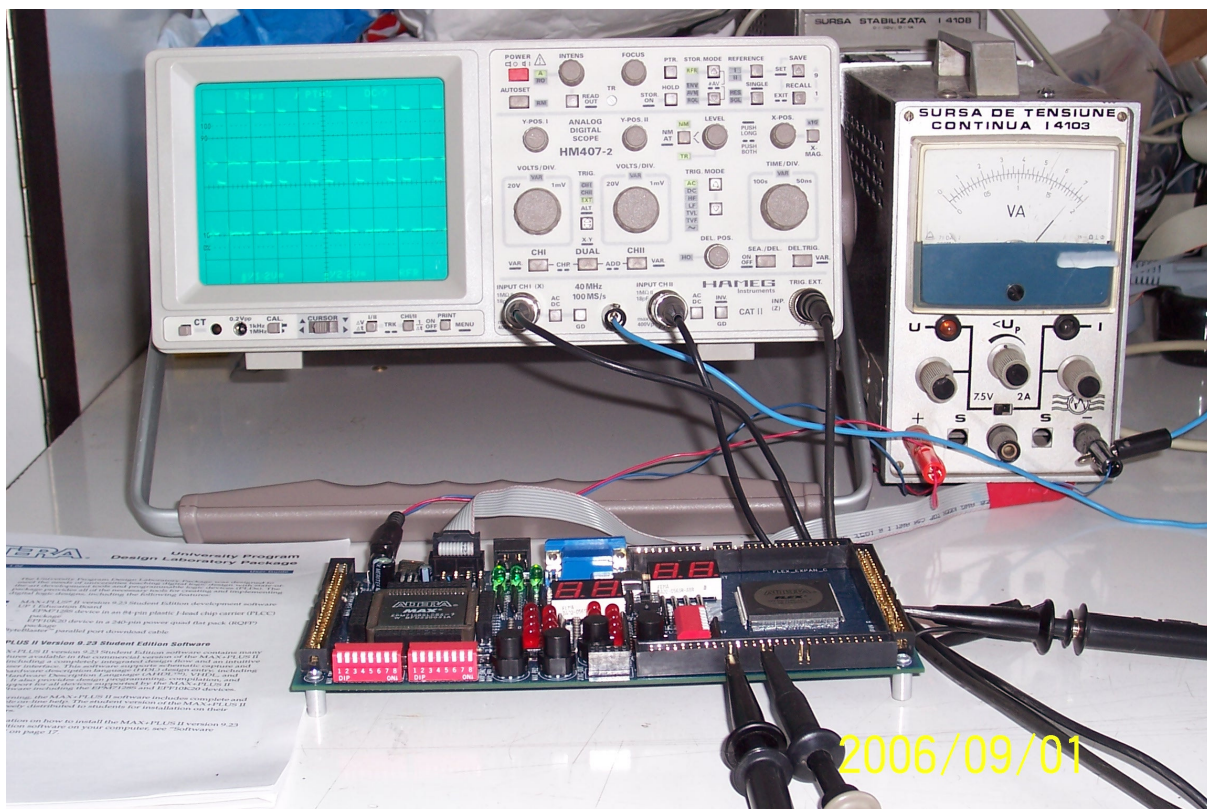


Fig. 5.31 Structura standului experimental

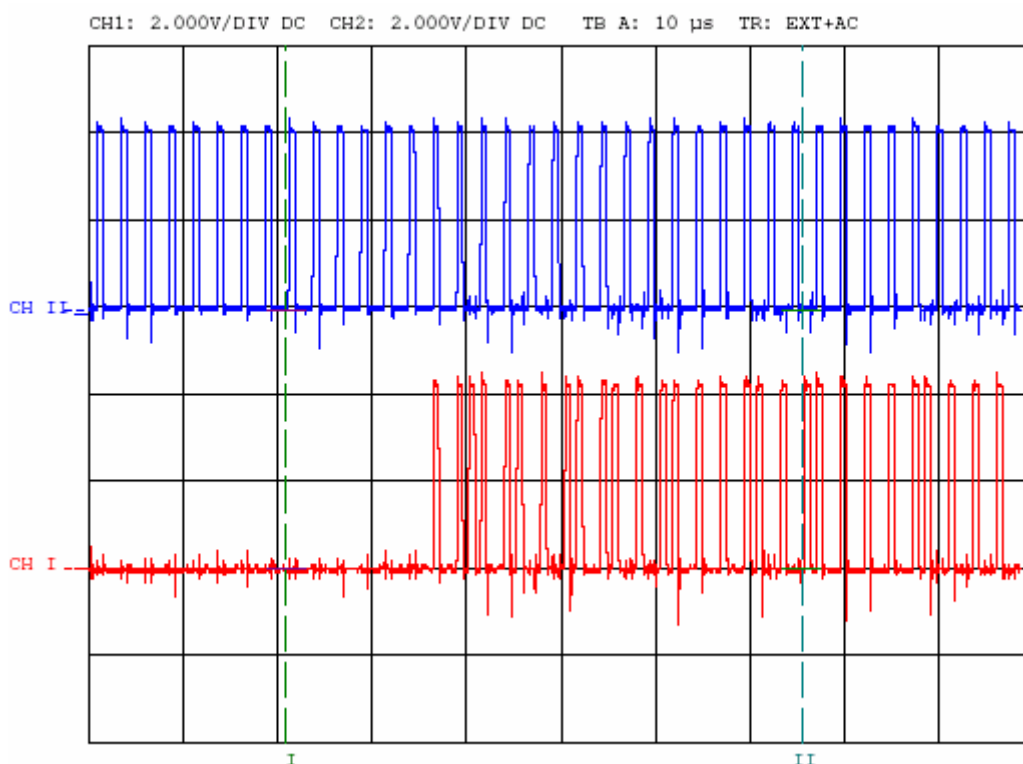


Fig. 5.32 Oscilograma generarii tactului de esantionare pentru 2 canale: CH 1 – referință; CH 2- canal lateral

Experimentul a constatat în încărcarea în FPGA a codului sintetizat pentru două canale și achiziția formelor de unda redată pe ecranul osciloscopului. Canalul de referință este

canalul central pentru care eşantionarea este uniformă şi faţă de care un canal lateral este eşantionat neuniform şi cu o întârziere iniţială.

În figurile 5.32 - 5.34 sunt prezentate formele de undă rezultate la ieşirile generatoarelor de tact ale celor două canale. Se remarcă identitatea formelor de undă cu secvenţele rezultate la simularea codului VHDL date în figurile 5.11 şi 5.12

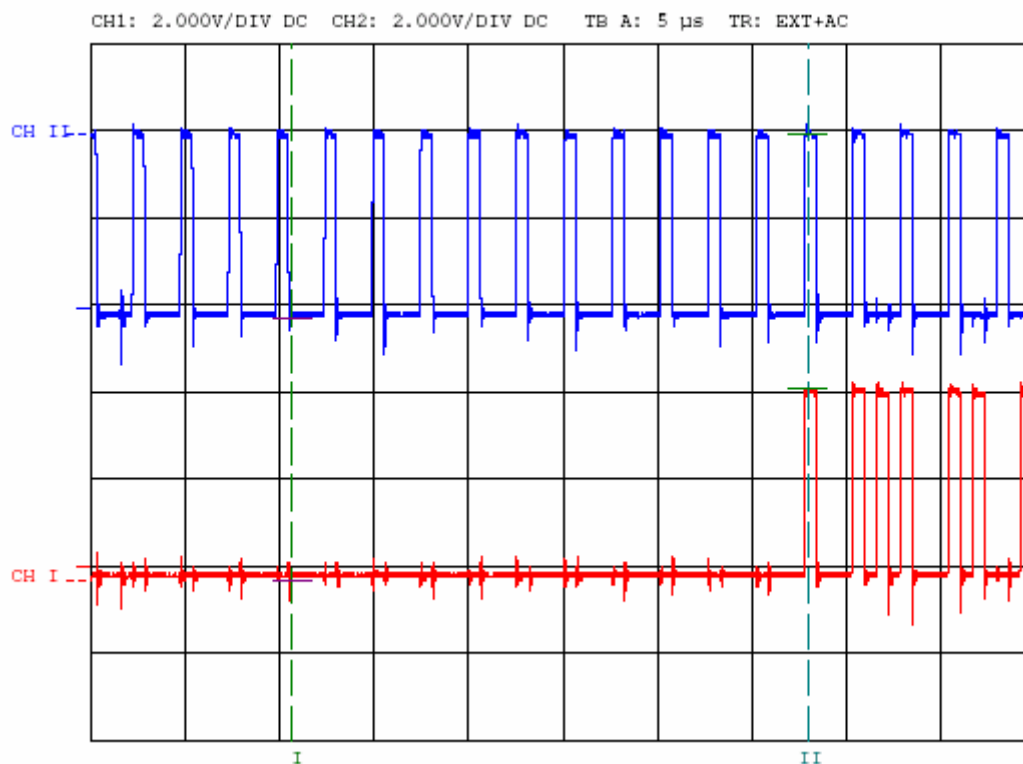


Fig. 5.33 Oscilograma generării tactului de eşantionare pentru 2 canale:- detaliu

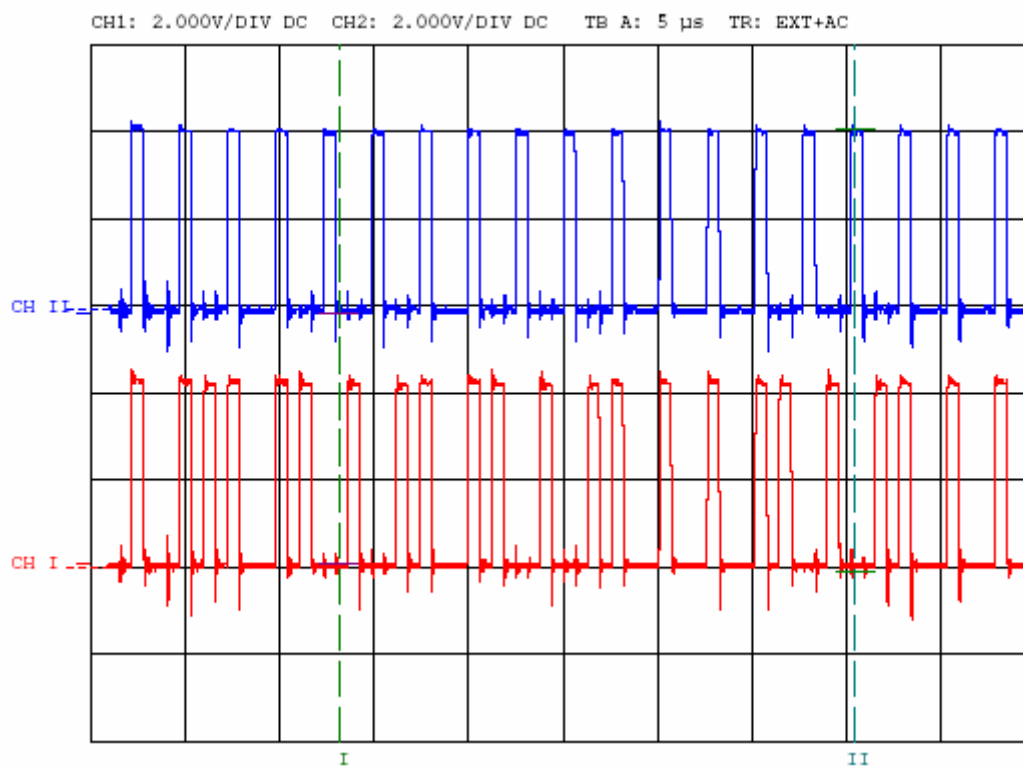


Fig. 5.34 Oscilograma generării tactului de eşantionare pentru 2 canale – detaliu

5.7. CONCLUZII

Capitolul abordează din punct de vedere structural secțiunea digitală a formatorului de fascicul cu modulație sigma delta neuniformă alcătuită din: controllerul digital, memoria FIFO, sumatorul, filtrul decimator

Entitatea centrală este controlerul digital care pe de o parte generează tactul de eșantionare neuniformă pentru comanda modulatorilor delta-sigma și încărcarea bufferelor FIFO iar pe de altă parte asigură semnalele de comandă pentru descărcarea FIFO, pentru sumator și pentru filtrul decimator.

Alegerea tehnologiei de implementare a secțiunii digitale necesită o analiză comparativă privind avantajele și dezavantajele tehnologiilor de implementare cu structuri logice programabile – FPGA, respectiv cu circuite integrate dedicate - ASIC.

FPGA-urile au două avantaje importante comparativ cu tehnologia ASIC:

- pot fi programate de mai multe ori (ASIC – ul se programează o singură dată)
- costul dispozitivelor este redus comparativ cu costul unui ASIC în special în etapa de realizare a prototipurilor.

Un argument suplimentar pentru folosirea tehnologiei de realizare cu circuite programabile, în situația dată, îl constituie existența pe piață a familiilor de dispozitive care conțin blocuri de memorie încorporate. Utilizarea acestor dispozitive crește șansele de realizare a unei implementări compacte unice a secțiunii digitale a formatorului. Familiile FLEX, APEX și STRATIX realizate de firma ALTERA pot fi opțiuni pentru faza de prototip.

O metodologie adecvată de proiectare presupune realizarea prototipului folosind un FPGA și utilizarea tehnologiei ASIC ca soluție de implementare finală pentru producția de serie.

Alegerea familiei Flex10k pentru implementarea structurii digitale a formatorului are la baza următoarele argumente:

- Dispozitivele Altera FLEX 10K, primele PLD-uri industriale cu blocuri de memorie încorporate, au o arhitectura bazată pe elemente reconfigurabile de tip RAM CMOS care încorporează toate trăsăturile necesare matricilor de porți în vederea implementării megafuncțiilor.

- Fiecare dispozitiv FLEX 10K conține o matrice încorporată și o matrice logică. Matricea de memorie încorporată este folosită pentru implementarea unei varietăți de funcții de memorie sau a unor funcții logice complexe. Matricea logică este folosită pentru implementarea de funcții logice generale, cum ar fi numărătoarele, sumatoarele, automatele de stare și multiplexoarele. Combinația celor 2 tipuri de matrici conduce la performanțe înalte și la posibilitatea implementării unui întreg sistem pe un singur dispozitiv.

- Dispozitivele FLEX 10K sunt suportate de mediile de dezvoltare Quartus și MAX+PLUS II; un pachet integrat unitar care oferă facilități de introducere a proiectelor în format grafic, text (incluzând AHDL) sau formă de undă, compilare și sinteză logică, simulare completă, analiză de timp pentru cele mai defavorabile cazuri și configurare de dispozitiv.

Determinarea întârzierii necesare fiecărui element traductor spre a asigura focalizarea dinamică pentru fiecare punct dintr-un cadru imagine necesită efectuarea cu rapiditate a unui mare număr de calcule complexe în timp real sau un volum de stocare considerabil pentru memorarea întârzierilor calculate “off-line”. O soluție pentru reducerea necesarului de memorie constă în compresia informației pentru focalizare. În acest sens pentru generarea tactului de eșantionare neuniformă s-a investigat posibilitatea calculării iterative în timp real, “în circuit”, a informației de întârziere pe baza algoritmului punctului median aplicat unei ecuații care derivă din geometria de focalizare.

Algoritmul punctului median estimează valoarea întârzierii l funcție de domeniul r din ecuația întârzierii folosind adunări și scăderi întregi. Euristic algoritmul punctului median alege valoarea cuantificată de pe grid, i, cea mai apropiată de curba reală. Deoarece valorile cuantificate sunt spațiate uniform valoarea cea mai apropiată i este una situată la mai puțin de jumătate de nivel de cuantificare de curbă.

Dependenta volumului calculelor de frecvența de esantionare a condus la studierea posibilității de a evalua funcția de decizie la un multiplu al unității de distanță. Alegerea valorii de multiplicare a incrementului de distanță s-a făcut funcție de eroarea de aproximare introdusă. Rezultatele simulărilor MATLAB confirmă raționamentul pentru alegerea valorii $K=2$ care asigură, pentru valori ale lui r mai mari de 32λ ($512\lambda/16$) încadrarea erorii de aproximare a curbei reale în intervalul $[-0,5 - 0,5]$.

În varianta finală structura funcțională care descrie algoritmul conține doar registre, multiplexoare și blocuri sumatoare cu două intrări fiind ușor implementabilă într-un dispozitiv programabil. Arhitectura descrisă în VHDL a fost sintetizată într-un FPGA din familia Flex folosind mediile ORCAD-PLD și MAX+PLUS2. Utilizarea dispozitivului arată că acesta poate găzdui generatoarele tactului de esantionare pentru o arie cu 64 de elemente și prin urmare există bune șanse de a implementa într-un dispozitiv Altera de capacitate mai mare, cum sunt cele din familia STRATIX, secțiunea digitală a fasciculului receptor.

Verificarea soluției a necesitat parcurgerea succesiunii modelare – simulare – sinteză în următoarele etape:

- modelarea Matlab a algoritmului și generarea unui fișier de date de ieșire
- generarea în Matlab a semnalelor de intrare pentru circuitul implementat
- simularea circuitului în Orcad – PLD folosind ca intrare semnalul generat
- transferul rezultatelor simulării în mediul MATLAB și folosirea capabilităților grafice ale acestuia pentru a compara rezultatele modelării cu cele ale simulării.
- achiziția pe un osciloscop digital a semnalului generat de PLD
- transferul datelor achiziționate în MATLAB și compararea cu rezultatele modelării / simulării.

Metoda are un anumit grad de generalitate pentru sisteme digitale dacă se asigură compatibilizarea formatelor de transfer a datelor între mediul de modelare (MATLAB), cel de simulare și sinteză (Orcad + Max+Plus2) și instrumentul de măsură (Osciloscop sau Analizor Logic).

Buferul de memorie – a fost construit în structura unui bloc de memorie RAM cu două porturi pe baza modului de bibliotecă parametrizabil *lpm-ram-dp* disponibil în bibliotecile Altera. Structura scalară a acestor module permite implementarea oricărui tip de memorie RAM sau ROM.

Operarea buferului de memorie este asincronă la scriere și sincronă la citire. Tactul de scriere este generat simultan cu adresa de scriere în generatorul semnalului de eşantionare separat pentru fiecare canal. Tactul și adresa de citire se generează simultan pentru toate buferele de către blocul de comandă al filtrului. Cu același tact se comandă și sumatorul care primește la intrare, sincron, eşantioanele de la cele 128 canale și dă la ieșire suma acestora. Pentru a reduce întârzierea introdusă de sumator acesta a fost conceput într-o structură piramidală cu 6 nivele de însumare.

Sinteza sumatorului se bazează pe facilitatea “*Carry chain*” a arhitecturii FLEX care pune la dispoziție o funcție de transport în avans foarte rapidă și permite implementarea, într-un mod eficient, a unor celule de însumare foarte rapide.

Ieșirea sumatorului cuprinde pe lângă semnalul util componente de frecvențe înalte, zgomotul de cuantificare, zgomotul introdus de circuite și interferențele. Datorită prezenței în

semnalul modulat a componentelor exterioare benzii utile este necesar un filtru trece jos cu pantă abruptă. Realizarea unor astfel de filtre pentru frecvențe ridicate necesită costuri semnificative și ca urmare implementarea lor trebuie atent analizată.

Deoarece pentru aplicațiile de formare de fascicul este de importanță crucială conservarea informației de fază din semnalele recepționate, filtrul decimator trebuie să asigure o întârziere de grup constantă pentru toate frecvențele sau astfel spus o fază liniară. Datorită structurii nerecursive filtrele FIR sunt întotdeauna stabile și dacă au coeficienții simetrici asigură faza liniară.

Proiectarea filtrului FIR cu fază liniară prin metoda ponderării Cebîșev folosind algoritmul de schimbare Remez este indicată pentru aplicația dată deoarece se realizează o proiectare optimală în sensul minimizării erorii maxime dintre răspunsul actual în frecvență și cel dorit și se obține filtrul de ordin minim care satisface specificația dată.

Evaluarea influenței cuantificării coeficienților a condus la stabilirea unor relații care permit studierea impactului cuantificării coeficienților asupra unui filtru proiectat prin metode conventionale și în final determinarea valorii coeficienților.

În practică proiectarea unui filtru digital presupune atât calculul coeficienților cât și dezvoltarea unui concept hardware implementabil în etapa următoare. În acest sens alegerea unei arhitecturi programabile de tipul FLEX furnizate de ALTERA asigură implementarea optimală a operațiilor de multiplicare și însumare prin folosirea tabelelor de căutare (Look-Up Tables).

Pentru verificarea funcționării filtrului implementat într-un dispozitiv ALTERA s-a definit o secvență de testare și rutinele aferente pentru conversia datelor între mediul MATLAB în care s-a făcut modelarea și analiza în domeniile timp/frecvență și mediile Orcad (modul PLD) și Quartus în care s-a realizat implementarea și verificarea funcționării prin simulare. Analiza comparativă în domeniul timp și respectiv în domeniul frecvență între semnalul original și cel filtrat confirmă corectitudinea soluției. În final, pe baza rezultatelor obținute la implementările parțiale în dispozitive din familia Flex10k, s-a reușit implementarea completă a secțiunii digitale pentru un formator cu 128 de canale într-un dispozitiv din familia Stratix. Utilizarea dispozitivului arată că dispozitivul ales conține suficiente resurse pentru a extinde numărul canalelor procesabile la 256.

CAPITOLUL 6

CONCLUZII GENERALE

Această teză, rezultat al activității de perfecționare prin doctorat, corelată cu preocupările profesionale ale autorului, reflectă contribuțiile personale și realizările concrete ale acestuia în domeniul sistemelor imagistice ultrasonice

Activitatea de cercetare și concepția inovativă a autorului în domeniul sintezei fasciculelor ultrasonice s-au manifestat prin conceperea de soluții formare a fasciculelor, prin modelarea, simularea și implementarea structurilor rezultate folosind limbaje dedicate MATLAB și VHDL și prin comunicarea rezultatelor cercetării în cadrul unor manifestări științifice interne și internaționale.

6.1 CONTRIBUȚII PERSONALE

Principalele contribuții ale autorului însoțite de referințele bibliografice la articolele publicate sunt prezentate sintetic în continuare

Capitolul 1

1.1 Abordarea cronologică a dezvoltării imagisticii ultrasonice în corelație cu disponibilitățile tehnologice pornind de la observația că progresele rapide în electronică, tehnica de calcul și tehnologia materialelor au modelat în fiecare etapă “viitorul” imagisticii ultrasonice.

Cronologia expusă evidențiază un decalaj între apariția tehnologiei și manifestarea efectelor sale la nivelul sistemelor de investigare. Existența acestui decalaj conduce la concluzia că ultimele dezvoltări tehnologice nu și-au manifestat în totalitate impactul asupra sistemelor imagistice. Prin urmare potențialul creșterii complexității și performanței sistemelor imagistice rezidă în combinarea dezvoltărilor recente în electronică cu o mai bună înțelegere a interacțiunii ultrasunetelor cu mediul.

Capitolul 2

2.1 Studiul amănunțit al bibliografiei în domeniu care a condus la următoarea concluzie:

Evoluția tehnicilor de formare a fasciculelor ultrasonice este un proces care rulează în paralel cu dezvoltarea și perfecționarea resurselor de procesare. Scopul final al procesului de formare a fost întotdeauna același, însumarea de semnale întârziate cât mai precis, dar modalitatea de implementare a fost subordonată stadiului atins în concepția circuitelor electronice și a algoritmilor de prelucrare. [Lie98], [Lie00a]

2.2 Definirea structurală a formatorului receptor pentru implementarea în două circuite integrate: [Lie04a],

- secțiunea analogică, care conține convertoarele $\Delta\Sigma$ (unul pentru fiecare canal), realizabilă sub forma unui ASIC
- secțiunea digitală, alcătuită din: liniile digitale de întârziere, sumator, decimator și blocul de comandă și control, implementabilă într-o structură logică programabilă (FPGA)

Capitolul 3

3.1 O analiză unitară a problematicii legate de cuantificarea semnalelor recepționate de o arie de traductoare ultrasonice, orientată pe două direcții principale:

- identificarea și evaluarea efectului erorilor de cuantificare a fazei asupra performanțelor formatorului digital de fascicul

Concluzia rezultată este că la investigarea ultrasonică medicală care folosește atât dirijarea cât și focalizarea, lobii laterali prezintă interes pentru sistemele cu un număr mare de elemente care funcționează la frecvențe ridicate (OSR redus). Folosirea unor frecvențe de eșantionare ridicate conduce la diminuarea influenței lobilor laterali. Relația (3.50) indică o valoare mai mare de 16 pentru OSR în cazul unei arii de 128 elemente cu ponderare uniformă. Acest rezultat constituie un criteriu de alegere a factorului de supraeșantionare și a structurii convertorului.

- determinarea dependenței complexitate performanță pentru convertoarele $\Delta\Sigma$

Această direcție de analiză a condus la următoarele concluzii:

- performanțele convertoarelor $\Delta\Sigma$ sunt direct corelate cu ordinul modulatorului
- asigurarea stabilității convertoarelor de ordin superior impune cerințe restrictive de proiectare care se regăsesc în final în creșterea complexității convertorului.

3.2 Identificarea criteriilor de alegere a arhitecturii potrivite pentru convertorul delta-sigma de mare viteză dedicat formatorului de fascicul ultrasonic:

- un factor de supraeșantionare mai mare sau egal cu 32 , pentru a asigura formarea cu acuratețe a fasciculului și diminuarea influenței lobilor laterali dați de eroarea de cuantificare pentru o arie cu 128 elemente.
- asigurarea unui domeniu dinamic de 80-100 dB pentru procesarea unei game largi de semnale inclusiv a reflexiilor provenite din mediul sangvin aflat în mișcare
- păstrarea complexității în limite rezonabile pentru a conserva șansa de integrare într-un singur circuit a convertoarelor pentru o arie de 64, 128 sau 256 elemente.

Capitolul 4

4.1 Elaborarea suportului software pentru testarea nivelului de procesare digitală. Blocurile care fac obiectul lucrării: formatorul de fascicul și convertorul delta-sigma au fost testate soft folosind setul de date în format numeric rezultate în urma conversiei analog numerice multi-bit, un program care implementează algoritmul de formare a fasciculului ultrasonic și o rutină care descrie software conversia de la coordonatele polare la cele carteziane. [Lie06a], [Lie06b]

4.2 Evaluarea comparativă a versiunilor unei imagini generate cu un formator convențional multibit și respectiv cu un formator Delta-sigma cu eșantionare uniformă și identificarea cauzei apariției unui zgomot suplimentar în imaginea generată cu formatorul Delta-sigma. Cauza apariției acestui zgomot a fost detectată a fi modificarea dinamică a întârzierilor în timpul procesului de formare de fascicul. [Lie06a]

4.3 Explicarea procesului de formare de fascicul ca proces de modulare - demodulare și poziționarea operației neliniare (întârziere) și a celei liniare (însurare) pentru

eliminarea zgomotului de reesantionare. Analiza justifică folosirea esantionării neuniforme în procesul de formare de fascicul. [Lie06a], [Lie06b]

4.4 Evaluarea cantitativă a celor trei variante de formare a fascicului folosind pentru cuantificarea performanțelor reprezentarea în domeniul timp și analiza spectrală la nivel de fascicul. Nivelul zgomotului de joasă frecvență mai mare cu 20dB la focalizarea dinamică confirmă ipoteza că acesta se datorează desincronizărilor modulator/demodulator la modificarea dinamică a întârzierilor.

Pentru formatorul cu modulație delta-sigma neuniformă cu FFTr/FDRec, care respectă condiția de evitare a repetiției eşantioanelor, spectrul semnalului de ieşire indică un nivel al zgomotului de joasă frecvență comparabil cu cel rezultat pentru FFTr/FFRec și doar cu 10 dB peste nivelul indicat de caracteristica de referință. Reprezentările în domeniul timp date pentru semnale de ieşire filtrate și decimate arată reducerea de 15dB a nivelului de zgomot pentru formatorul cu modulație neuniformă comparativ cu cel cu modulație uniformă necompensat datorită păstrării sincronismului modulator – demodulator și evitării repetiției eşantioanelor la formatorul cu esantionare neuniformă. [Lie06a], [Lie06b]

4.5 Alegerea soluției constructive pe baza criteriului „la performanțe similare decide complexitatea hardware”. Din acest punct de vedere soluția cu modulație neuniformă este mai simplă deoarece nu presupune modificarea modulatorului ci doar asigurarea raportului de frecvențe pentru a se evita repetiția. În ceea ce privește generarea și controlul întârzierilor suportul de memorie necesar este practic același în cele două situații. Reducerea în continuare a complexității structurii hardware are în vedere tocmai înlocuirea blocului de memorie folosit pentru stocarea întârzierilor cu un mecanism de generare „on-line” a secvențelor de întârziere. [Lie05e], [Lie06a], [Lie06b]

Capitolul 5

5.1 Alegerea tehnologiei de implementare a secțiunii digitale pe baza unei analize comparative privitoare la avantajele și dezavantajele tehnologiilor de implementare cu structuri logice programabile – FPGA, respectiv cu circuite integrate dedicate - ASIC. O metodologie adecvată de proiectare presupune realizarea prototipului folosind un FPGA și utilizarea tehnologiei ASIC ca soluție de implementare finală la producția de serie.

5.2 Identificarea unei soluții pentru compresia informației de focalizare pentru formatorul cu eşantionare neuniformă. Pentru generarea tactului de eşantionare neuniformă s-a investigat posibilitatea calculării iterative în timp real, “în circuit”, a informației de întârziere pe baza algoritmului “punctului median” aplicat unei ecuații care derivă din geometria de focalizare. Dependenta volumului calculelor de frecvența de esantionare a condus la studierea posibilității de a evalua funcția de decizie la un multiplu al unității de distanță. Alegerea valorii de multiplicare a incrementului de distanță s-a făcut funcție de eroarea de aproximare introdusă.

Rezultatele simulărilor MATLAB confirmă raționamentul pentru alegerea valorii $K = 2$ care asigură, pentru valori ale lui r mai mari de 32λ ($512 \lambda/16$) încadrarea erorii de aproximare a curbei reale în intervalul $[-0,5 - 0,5]$. [Lie04a], [Lie05a],

5.3 Conceperea unei structuri functionale care descrie algoritmul folosind doar registre, multiplexoare și blocuri sumatoare cu două intrări ușor implementabile într-un dispozitiv programabil. Arhitectura descrisă în VHDL a fost sintetizată într-un FPGA din familia Flex folosind mediile ORCAD-PLD și MAX+PLUS2. Utilizarea dispozitivului arată că acesta poate găzdui generatoarele tactului de esantionare pentru

o arie cu 64 de elemente și prin urmare exista bune sanse de a implementa într-un dispozitiv Altera de capacitate mai mare, cum sunt cele din familia STRATIX, secțiunea digitală a fasciculului receptor . [Lie04a], [Lie05a]

5.4 Definirea unei metodologii de verificare a soluției prin parcurgerea succesiunii modelare – simulare – sinteza în următoarele etape:

- modelarea Matlab a algoritmului și generarea unui fișier de date de ieșire
- generarea în Matlab a semnalelor de intrare pentru circuitul implementat
- simularea circuitului în Orcad – PLD folosind ca intrare semnalul generat
- transferul rezultatelor simulării în MATLAB și folosirea capacităților sale grafice pentru a compara rezultatele modelării cu cele ale simulării.
- achiziția pe un osciloscop digital a semnalului generat de PLD
- transferul datelor achiziționate în MATLAB și compararea cu rezultatele modelării / simulării.

Metoda are un anumit grad de generalitate pentru sisteme digitale dacă se asigură compatibilizarea formatelor de transfer a datelor între mediul de modelare (MATLAB), cel de simulare și sinteza (Orcad + Max+Plus2) și instrumentul de măsură. [Lie05a]5.4

5.5 Proiectarea filtrului FIR cu fază liniară prin metoda ponderării Cebîșev folosind algoritmul de schimbare Remez. Evaluarea influenței cuantificării coeficienților și stabilirea unor relații care permit studierea impactului cuantificării coeficienților asupra unui filtru proiectat prin metode convenționale și în final determinarea valorii coeficienților.[Lie05d]

5.6 Definirea secvenței de testare a filtrului implementat într-un dispozitiv programabil și a rutinelor aferente pentru conversia datelor între mediul MATLAB în care s-a făcut modelarea și analiza în domeniile timp/frecvență și mediile Orcad (modul PLD) și Quartus în care s-a realizat implementarea și verificarea funcționării prin simulare. Analiza comparativă în domeniul timp și respectiv în domeniul frecvență între semnalul original și cel filtrat a confirmat corectitudinea soluției. [Lie05d]

5.7 Sinteza într-un FPGA a secțiunii digitale pentru formatorul cu modulație delta-sigma neuniformă cu 128 canale [Lie05a].

6.2 PERSPECTIVE DE VIITOR

Subiectul abordat în cadrul lucrării se înscrie în tematica mai largă a miniaturizării sistemelor de investigare ultrasonică. Cercetările întreprinse de autor conturează două direcții de studiu și dezvoltare pe termen scurt în domeniul formatoarelor de fascicul ultrasonic.

- identificarea unor soluții optime de implementare pe un singur chip a unui număr cât mai mare de modulatori delta-sigma
- conceperea și evaluarea de soluții alternative pentru compresia informației de focalizare, care să conducă la structuri simple și eficiente din punct de vedere al consumului și vitezei de operare

Pe termen mediu o direcție promițătoare sugerată de cercetările din lucrare este aceea de implementare a modulatorilor delta-sigma pe același chip cu logică aferentă și realizarea de soluții uni-chip pentru interfațarea cu mediul analogic

Este de așteptat ca pe termen lung tendințele de miniaturizare sprijinite și pe direcțiile enumerate să se manifeste prin migrarea electronicii de interfață și nu numai în capul de scanare, prin creșterea densității la nivel analogic și digital exprimate în număr de funcții sau canale per chip și în final prin implementarea de formatoare software pe procesoare de mare viteză.

BIBLIOGRAFIE

- [1] [Amarandos95] M. Amarandos "Sistem Issues in Medical Imaging." Electronic Engineering, February 1995, pp. 55-57.
- [2] [Aarnio01] J. Aarnio, G. T. Clement, and K. Hynynen "Investigation of ultrasound phase shifts caused by the skull bone using low-frequency reflection data." IEEE Ultrasonics Symp. Proc., 2001, pag. 1503–1506.
- [3] [Aubry01] J. F. Aubry, M. Tanter, J. L. Thomas and M. Fink "Pulse echo imaging through a human skull: In vitro experiments." IEEE Ultrasonics Symp. Proc., 2001, pag. 1499–1502.
- [4] [Aziz96] P.M. Aziz, H.V. Sorensen, and J. Van Der Spiegel, "An Overview of Sigma-Delta Converters." IEEE Signal Processing Magazine, Vol. 13, No. 1, pp. 61-84, Jan. 1996.
- [5] [Bae00] M. H. Bae and M. K. Jeong. "A study of synthetic-aperture imaging with virtual source elements in B-mode ultrasound imaging systems." In IEEE Trans. Ultrason., Ferroelec., Freq. Contr., volume 47, pages 1510–1519, 2000.
- [6] [Berckhoff94] A. P. Berckhoff, H. J. Huisman, "Fast Scan Conversion Algorithms for Displaying Ultrasound Sector Images", Ultrasonic Imaging, No. 16, 1994
- [7] [Biquard72] P. Biquard "Paul Langevin." Ultrasonics 10, 1972, pag. 213–214.
- [8] [Black99] B. Black "Analog-to-Digital Converter Architectures and Choices for System Design." Electronic Analog Dialogue 33-8, 1999.
- [9] [Brenner01] A. E. Brenner „More on Moore's law." Physics Today 54, 2001, pag. 84.
- [10] [Bresenham65] J. E. Bresenham. "Algorithm for computer control of a digital plotter." IBM Systems Journal., 4:25–30, 1965.
- [11] [Brunner02] E. Brunner "Ultrasound System Considerations and their Impact on Front-End Components." Analog Devices, 2002.
- [12] [Candy92] J. C. Candy and G. C. Temes. "Oversampling Delta-Sigma Data Converters - Theory, Design and Simulation." IEEE Press, 1992.
- [13] [Chang93] S.H. Chang, S.B. Park, and G.H. Cho, "Phase-error free quadrature sampling technique in the ultrasonic B-scan imaging system and its application to the synthetic focusing system," IEEE Trans. Ultrasonics Ferro. and Freq. Control, vol 40, no 3, May 1993, pp 216-223.
- [14] [Chiang00] A. M. Chiang, P. P. Chang, and S. R. Broadstone. "PC-based ultrasound imaging system in a probe." Proc. IEEE Ultrason. Symp., 2:1255–1260, 2000.
- [15] [Cho96] W.H. Cho and Y.B. Ahn, "Multiorder Sampling for Digital Beamforming of Wide-band Signals." IEEE Trans. Ultrasonics Ferro. and Freq. Control, vol 43, no 3, May 1996, pp 495-499.
- [16] [Cote01] D. Cote, Personal Communication. (2001).
- [17] [Dean95] M. Dean "Digital Beamforming Array", U. S. Patent #5,461,389, Octombrie 24, 1995.
- [18] [Deventer00] J. van. Deventer, T. Lofquist and J. Delsing, "PSPICE Simulation of Ultrasonic Systems." IEEE Trans. Ultrasonics Ferro. and Freq. Control, vol 47, no 4, July 2000, pp 1014-1024.
- [19] [Edler91] I. Edler, „Early echocardiography." Ultrasound in Med. & Biol. 17, 1991, pag. 425–431.
- [20] [Engeler 92]. W.E. Engeler, M. O'Donnell, J.T. Pedicone and J.J. Bloomer "Dynamic Phase Focus for Coherent Imaging Beam Formation", U. S. Patent #5,111,695, May 12, 1992.

-
- [21] [Errico95] P. Errico and A. Hill “*Choosing High-Speed Signal Processing Components for Ultrasound Systems*” Analog Dialogue 29-2, Februarie 1995.
 - [22] [Erikson74] K. R. Erikson, F. J. Fry and J. P. Jones „*Ultrasound in medicine: A review.*” IEEE Trans. Sonics Ultrasonics SU-21, 1974, pag. 144–170.
 - [23] [Eyer81] M. K. Eyer, M. A. Brandestini, D. J. Philips and D. W. Baker „*Color digital echo/Doppler presentation.*” Ultrasound in Med. & Biol. 7, 1981 21.
 - [24] [Feldkamper00] H. T. Feldkamper, R. Schwann, V. Gierenz and T. G. Noll “*Low Power Delay Calculation for Digital Beamforming in Handheld Ultrasound Systems*”, IEEE Ultrasonics Symposium Proceedings, 2000.
 - [25] [Fenster96] A. Fenster, D. B. Downey “*3D Ultrasound Imaging: A Review.*” IEEE Engineering in Medicine and Biology, November / December 1996
 - [26] [Flax88] S. W. Flax and M. O'Donnell. “*Phase-aberration correction using signals from point reflectors and diffuse scatterers: basic principles.*” IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 35:758–767, 1988.
 - [27] [Freeman 95a] S. Freeman and M. O'Donnell. “*A complex arithmetic digital signal processor using CORDIC rotators.*” International Conf. Acous., Speech, Signal Processing, Vol. 5 May, 1995.
 - [28] [Freeman 95b] S. Freeman, Pai-Chi Li, and M. O'Donnell. “*Retrospective dynamic transmit focusing.*” Ultrasonic Imaging, 17 173–196, 1995.
 - [29] [Freeman99a] S. R. Freeman, M. K. Quick, M. A. Morin, R. C. Anderson, C. S. Desilets, T. E. Linnenbrink, and M. O'Donnell. “*Delta-sigma oversampled ultrasound beamformer with dynamic delays.*” IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 46:320–332, 1999.
 - [30] [Freeman99b] S. R. Freeman, M. K. Quick, M. A. Morin, R. C. Anderson, C. S. Desilets, T. E. Linnenbrink, and M. O'Donnell. “*Heterodyning technique to improve performance of delta sigma based beamformers.*” IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 46:771–790, 1999.
 - [31] [Gierenz01] V. Gierenz, R. Schwann, and T. G. Noll “*A Low Power Digital Beamformer for Handheld Ultrasound Systems*”, 2001
 - [32] [Goldberg88] B. B. Goldberg and B. A. Kimmelman „*Medical Diagnostic Ultrasound: A Retrospective on its 40th Anniversary.*” Eastman Kodak Company, New York. 1988.
 - [33] Gray85] D. A. Gray, *Effect of time-delay errors on the beam pattern of a linear array*, IEEE Journal Ocean. Eng., vol. OE-10, July, 1985.
 - [34] [Holm92] S. Holm and K. Kristoffersen, *Analysis of worst-case phase quantization sidelobes in focused beamforming*, IEEE Trans. Ultrasonics, Ferro and Freq. Contr., pages 593–599, 1992.
 - [35] [Holm94] S. Holm, *Digital beamforming in ultrasound imaging*. In Nordic Signal Processing Symposium, 1994.
 - [36] [Holm97] S. Holm, B. Elgetun, and G. Dahl. “*Properties of the beampattern of weight- and layout-optimized sparse arrays.*” IEEE Tran. on Ultrasonics, Ferro. and Freq. Control, vol. 44 no. 5 pp. 983–991, 1997.
 - [37] [Holmes80] J. H. Holmes „*Diagnostic ultrasound during the early years of the AIUM.*” J. Clin. Ultrasound 8, 1980, pag. 299–308.
 - [38] [Hunt83] J. W. Hunt, M. Arditi, and F. S. Foster “*Ultrasound Transducers for Pulse-Echo Medical Imaging*” IEEE Transactions on Biomedical Engineering, Vol. BME-30, No. 8, pp. 453–481, August 1983.
 - [39] [Inerfield02] M. Inerfield, G. R. Lockwood and S. L. Garverick, “*A Sigma-Delta-Based Sparse Synthetic Aperture Beamformer for Real-Time 3-D Ultrasound*” IEEE Trans. on Ultrasonics, Ferro, and Freq. Control, Vol. 49, No. 2, pp. 243-254, February. 2002.
 - [40] [Jarman95] D. Jarman, “*A Brief Introduction to Sigma Delta Conversion*”, Application Note AN9504, Intersil, May 1995.

-
- [41] [Jensen93] J. A. Jensen, D. Gandhi, and W. D. O'Brien. "*Ultrasound fields in an attenuating medium.*" Proc. of IEEE Ultrason. Symp., pages 943–946, 1993.
 - [42] [Jensen96a] J. A. Jensen. "*Estimation of Blood Velocities Using Ultrasound: A Signal Processing Approach.*" Cambridge University Press, New York, 1996.
 - [43] [Jensen96b] J. A. Jensen. "*Field: A program for simulating ultrasound systems.*" , 10th Nordic-Baltic Conference on Biomedical Imaging, Vol. 4, Supplement 1, Part 1:351–353, 1996.
 - [44] [Jensen99] J. A. Jensen, O. Holm, L. J. Jensen, H. Bendsen, H. M. Pedersen, K. Salomonsen, J. Hansen, and S. Nikolov. Experimental ultrasound system for real-time synthetic imaging. In *Proc. IEEE Ultrason. Symp.*, volume 2, pages 1595–1599, 1999.
 - [45] [Jeon94] K. Jeon, M. H. Bae, S. B. Park, and S. D. Kim. "*An efficient real time focusing delay calculation in ultrasonic imaging systems.*" In *Ultrasonic Imaging*, volume 16, pages 231–248, 1994.
 - [46] [Johnson93] D. H. Johnson and D. E. Dudgeon. *Array signal processing. Concepts and techniques.* Prentice-Hall., Englewood Cliffs, New Jersey, 1993.
 - [47] [Karaman91] M. Karaman, A. Atalar and H. Koymen, "*Adaptive Digital Beamforming for Phased Array Ultrasound Imaging*" Proc. IEEE Ultrasonics Symposium, 1991, pp. 1207-1210.
 - [48] [Karaman93a] M. Karaman and H. Koymen. "*A phase aberration correction method for ultrasound imaging.*" IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 40:275–282, 1993.
 - [49] [Karaman93b] M. Karaman, A. Atalar, and H. Koymen. "*VLSI circuits for adaptive digital beamforming in ultrasound imaging.*" IEEE Trans. Med. Imag., 12:711–720, 1993.
 - [50] [Karaman95] M. Karaman, Pai-Chi Li, and M. O'Donnell. *Synthetic aperture imaging for small scale systems.* IEEE Trans. Ultrasonics, Ferro. and Freq. Contr., 42 429–442, 1995.
 - [51] [Kim95] J. Kim, L. X. Yao, Z. Banjanin, H. Fukukita, H. Hagiwara and M. Kawabuchi "*Partial Beamforming*", U. S. Patent #5,388,079, Februarie 7, 1995.
 - [52] [Kino87] G. S. Kino, *Acoustic Waves, Devices, Imaging and Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
 - [53] [Klein81] H. G. Klein „*Are B-scanners' days numbered in abdominal diagnosis?*" Diagnostic Imaging 3, 1981, pag. 10–11.
 - [54] [Kossloff74] G. Kossoff „*Display techniques in ultrasound pulse echo investigations: A review.*" J. Clin. Ultrasound 2, 1974, pag. 61–72.
 - [55] [Kozak99] M. Kozak and M. Karaman, "*Digital Beamforming Using Non-Uniform Oversampling Delta-Sigma Conversion*" Proc. IEEE Ultrasonics Symposium, 1999, pp. 1279-1282.
 - [56] [Kozak01] M. Kozak and M. Karaman, "*Digital Phased Array Beamforming Using Single-Bit Delta-Sigma Conversion with Non-Uniform Oversampling*" IEEE Trans. on Ultrasonics, Ferro. and Freq. Control, Vol. 48, No. 4, pp. 922-930, July. 2001.
 - [57] [Laakso96] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine. "*Splitting the unit delay.*" IEEE Sig. Proc. Mag., pages 30–60, 1996.
 - [58] [Larsen80] H.G. Larsen and S.C. Leavitt. *An image display algorithm for use in real-time sector scanners with digital scan converters.* IEEE Ultrasonic Symposium Proceeding , pages 763–765, 1980.
 - [59] [Leach94] W. M. Leach, "*Controlled-source Analogous Circuits and SPICE Models for Piezoelectric Transducers.*" IEEE Trans. Ultrasonics Ferro. and Freq. Control, vol 41, no 1, January 1994, pp. 60-66.
 - [60] [Li00] P. C. Li, J. J. Huang, H. L. Liu, and M. O'Donnell. "*A Dynamic focusing Technique for Delta-Sigma-Based Beamformers.*" Ultrasonic Imaging 22, pp. 197–205, 2000.

-
- [61] [Lie94] I. Lie, S. Ionel, C. Gavrilescu, "An Autoregressive Spectrum Analyzer for Ultrasound Signals.", Proceedings of the 4th International Conference on Optimisation of Electric and Electronic Equipments, "Transilvania" University of Brasov, pag. 163-168. 1994
 - [62] [Lie95a] I. Lie, C. Gavrilescu, "Interfață pentru Studiul Semnalelor Ultrasonice Doppler cu DSP", A 3-a Conferință Internațională de Ingineria Sistemelor Moderne in Electrotehnica, Oradea, 1995, pag. 98-101.
 - [63] [Lie95b] I. Lie, L. Jurca, "Considerații Asupra unei Metode de Investigare Ultrasonica." Sesiunea de Comunicări Științifice "Implicarea Cercetării Științifice in Dezvoltarea si Modernizarea Proceselor de Fabricație", Sibiu, 1995.
 - [64] [Lie96a] I. Lie, L. Jurca, *Ultrasonic Generator for Nondestructive Testing*. Proceedings of the 5th International Conference on Optimisation of Electric and Electronic Equipments, "Transilvania" University of Brasov, 1996, pag. 291-296.
 - [65] [Lie96b] I. Lie, S. Nanu, *Ultrasonic Generator with Frequency Control*, International Symposium on Systems Theory, Robotics, Computers and Process Informatics, SINTES 8, 6-7 June 1996, Craiova, pag. 139-144.
 - [66] [Lie96c] I. Lie, M. A. Herbei, "About the Signals Used to Pulsing the Ultrasonic transducers in NDT", Buletinul științific al UPT, Tom 41 (55) 1996, Fascicula 1. Proceedings of the International Symposium on Electronics and Telecommunications ETC96, Timisoara, September. 1996, pag 119-124.
 - [67] [Lie98] I. Lie, *A new implementation of dynamic focusing technique for ultrasonic systems*, Buletinul științific al UPT Tomul 43 (57) 1998 Fascicula 2 pp 121-126
 - [68] [Lie99a] I. Lie, *The calculation of ultrasonic transducer field*, Analele Universitatii din Oradea 1999, Volumul 2, pp 256-261
 - [69] [Lie99b] I. Lie, *An efficient pulse generator for driving ultrasonic arrays*, Proceedings of the 5th international workshop Bases of Electronics SBE'99, 10-11 june, 1999 Cluj-Napoca, Romania, pag.195-200.
 - [70] [Lie00a] I. Lie, "The digital ultrasound beamforming", Buletinul științific al UPT Tomul 45 (59) 2000 Fascicula 1, pp 269-274.
 - [71] [Lie00b] I. Lie "The compensation of ultrasound attenuation through time-gain correction", Buletinul științific al UPT Tomul 45 (59) 2000 Fascicula 1, pp 275-278.
 - [72] [Lie02] I. Lie, "About optimizing ultrasonic investigation using synthetic aperture method", Buletinul științific al UPT Tomul 47 (61) 2002 Fascicula 1, 2, pp. 43-48.
 - [73] [Lie03] M.E. Tănase, I. Lie, B. Marinca, *About Real Aperture Beamforming* Proceedings of the 9th International Symposium for Design And Technology of Electronic Package "SIITME 2003", pag. 225-229, Timișoara 2003.
 - [74] [Lie04a] I. Lie, M. E. Tanase, *The nonuniform sampling rate generation in ultrasonic beamformers*, Proceedings of International Conference on Signals and Electronic Systems ICSSES '04, in co-operation with IEEE Circuits and Systems Society, pp. 207-210, 13-15 september 2004, Poznan, Poland, ISBN 83-906074-7-6
 - [75] [Lie04b] I. Lie, M. E. Tanase, *CPLD based pulse – echo processor*, Proceedings of International Conference on Signals and Electronic Systems ICSSES '04, in co-operation with IEEE Circuits and Systems Society, pp. 337-340, 13-15 september 2004, Poznan, Poland, ISBN 83-906074-7-6
 - [76] [Lie05a] I. Lie, M. E. Tanase, *About the Possibility to Implement a Nonuniform Oversampling Receive Beamformer in a FPGA*, Proceedings of the 2005 IEEE Ultrasonics Symposium, Roterдам, The Netherlands, September 18-21, 2005, CD-ROM: 05CH37716C, IEEE Catalog Number: 05CH37716C, ISBN 0-7803-9383-X, ISSN 1051-0117, pp. 1404-1407
 - [77] [Lie05b] I. Lie, M. E. Tanase, *A Compact FPGA Beamformer Architecture*, Proceedings of 2005 WSEAS Int. Conf. on DYNAMICAL SYSTEMS and CONTROL, Venice, Italy, November 2-4, 2005 pp. 463-466, ISBN 960-8457-37-8

-
- [78] [Lie05c] I. Lie, M. E. Tanase, A *Sing-Around Ultrasonic Low Power Flowmeter*, Proceedings of 2005 WSEAS Int. Conf. on DYNAMICAL SYSTEMS and CONTROL, Venice, Italy, November 2-4, 2005, pp. 174-176, ISBN 960-8457-37-8
 - [79] [Lie05d] I. Lie, M.E. Tănase, H. Cârstea, A. Avram, *FPGA Implemented Decimating Filter*, Proceedings of 28th International Spring Seminar on Electronics Technology, ISSE 2005, pp. 215-218, may 19-22 2005, Wiener Neustadt, Austria.
 - [80] [Lie05e] I. Lie, M. E. Tănase, *Optimizing the cost to quality ratio in ultrasonic imaging*, Proceedings of the 4th International Conference on Microelectronics and Computer Science ICMCS – 2005, 15 -17 september 2005, pp. 393-396, Chisinau, Republic of Moldova.
 - [81] [Lie06a] I. Lie, M. E. Tanase, D. Lascu, M. Lascu, *Ultrasonic Beamforming with Delta-Sigma Modulators*, Proceedings the 10th WSEAS CSCC Multiconference, Athena, Grece, July, 2006
 - [82] [Lie06b] I. Lie, M. E. Tanase, D. Lascu, M. Lascu, *Compact Ultrasonic Beamformer Based on Delta-Sigma Modulation*, WSEAS Transactions on Circuits and Systems, Issue 8, Volume 5, August 2006, pp 1315-1322, ISSN 1109-2734.
 - [83] [Lie04ref] I. Lie, “*Studiul Comparativ al Tehnicilor de Formare a Fasciculului Ultrasonic.*” Referat nr. 1, Decembrie, 2004.
 - [84] [Lie05ref2] I. Lie, “*Concepții Actuale Privind Structura Hardware a Formatoarelor Digitale de Fascicul*” Referat nr. 2, Februarie, 2005.
 - [85] [Lie05ref3] I. Lie, “*Posibilități de Optimizare a Structurilor Hard și Soft a Echipamentelor Electronice de Investigare Ultrasonică.*” Referat nr. 3, Iulie, 2005.
 - [86] [Ligtvoet78] C. Ligtvoet, H. Rijsterborgh, L. Kappen and N. Bom „*Real time ultrasonic imaging with a hand-held scanner: Part I, Technical description.*” *Ultrasound in Med. & Biol.* 4, 1978, pag. 91–92.
 - [87] [Lipschutz94] D. Lipschutz “*Delay Interpolator for Digital Phased Array Ultrasound Beamformers*”, U. S. Patent #5,345,426, September 6, 1994.
 - [88] [Loockwood96] Geoffrey R. Lockwood, Pai-Chi Li, Matthew O'Donnell, and F. Stuart Foster. *Optimizing the radiation pattern of sparse periodic linear arrays.* IEEE Tran. Ultrasonics, Ferro. and Freq. Cont., 43(1) 4–12, 1996.
 - [89] [Lu94] Jian-Yu Lu, Hehong Zou, and James F. Greenleaf. “*Biomedical ultrasound beam forming*” *Ultrasound in Med. and Biol.*, Vol. 20.pp. 403–428, 1994.
 - [90] [Lutloff89] R.M. Lutloff, A. Vieli and S. Basler, “*Ultrasonic Phased Array Scanner with Digital Echo Synthesis for Doppler Echocardiography.*” IEEE Trans. Ultrasonics Ferro. and Freq. Control, vol 36, 1989, pp 494-506.
 - [91] [Macovski75] A. Macovski, “*Ultrasonic Array for Reflection Imaging*”, U. S. Patent #3,918,024, November 4, 1975.
 - [92] [Magnin81] P. A. Magnin, O. T. Von Ramm, and F. L. Thurstone. “*Delay quantization error in phased array images.*” IEEE Trans. Son. Ultrason., Vol. SU-28 No. 5, September 1981, pp. 305–310.
 - [93] [Maslak79] S.H. Maslak, “*Acoustic Imaging Apparatus*”, U. S. Patent #4,140,022, Februarie 20, 1979.
 - [94] [Maslak96] S.H. Maslak, C.R. Cole and J.G. Petrofsky “*Method and Apparatus for Doppler Receive Beamformer System*”, U. S. Patent #5,555,534, Septembrie. 10, 1996.
 - [95] [Na98] B. Y. Na, S. J. Kwon and Y. B. Ahn, “*Experimental Considerations of Multi-Order Sampling for Digital Beamforming*” Proc. IEEE Ultrasonics Symposium, 1998, pp. 1721-1724.
 - [96] [Nikolov00] S. I. Nikolov. “*Users guide for the beamformation toolbox.*” Technical report, Department of Information Technology, DTU, 2000.
 - [97] [Nikolov02] S. I. Nikolov and J. A. Jensen. “*Comparison between different encoding schemes for synthetic aperture imaging.*” In Proc. SPIE - Progress in biomedical optics and imaging, volume 3, pages 1–12, 2002.

-
- [98] [Nikolov01] S. I. Nikolov and J. A. Jensen. "Velocity estimation using synthetic aperture imaging." In Proc. IEEE Ultrason. Symp., pages 1409–1412, 2001.
 - [99] [Nock92] L. F. Nock and G. E. Trahey. "Synthetic receive aperture imaging with phase correction for motion and for tissue inhomogeneities - part I: basic principles." IEEE Trans. Ultrason., Ferroelec., Freq. Contr., 39:489–495, 1992.
 - [100] [Norman96] O. Norman, "A Band-Pass Delta-Sigma Modulator for Ultrasound Imaging at 160 MHz Clock Rate," IEEE Journal of Solid-State Circuits, Vol. 31, No. 12, pp. 2036–2041, Dec. 1996.
 - [101] [Noujaim93]. S.E. Noujaim S. L. Graverick and M. O'Donnell, "Phased Array Ultrasonic Beam Forming Using Oversampled A/D Converters", U. S. Patent #5,203,335, Aprilie 20, 1993
 - [102] [O'Brien98] W. O'Brien „Assessing the risks for modern diagnostic ultrasound imaging." Japanese J. of Applied Physics 37, 1998, pag. 2781–2788.
 - [103] [Odom99] B. Odom "Ultrasound Analog Electronics Primer" Analog Dialogue 33-5, May 1999.
 - [104] [O'Donnell88] M. O'Donnell, "Applications of VLSI circuits to medical imaging," Proceedings of the IEEE, Vol 76, pp. 1106–1114, September 1988.
 - [105] [O'Donnell89a]. M. O'Donnell and M.G. Magrane "Method and Apparatus for Fully Digital Beam Formation in a Phased Array Coherent Imaging System", U. S. Patent #4,809,184, Februarie 28, 1989.
 - [106] [O'Donnell89b]. M. O'Donnell, W.E. Engeler, T.L. Vogeslong, S. G. Karr and S.E. Noujaim "Method and Apparatus for High Speed Digital Phased Array Coherent Imaging System", U. S. Patent #4,839,652, Iunie 13, 1989.
 - [107] [O'Donnell90] M. O'Donnell, W.E. Engeler, J.T. Pedicone, A.M. Itani, S.E. Noujaim, R.J. Dunki-Jacobs, W.M. Leue, C.L. Chalek, L.S. Smith, J.E. Piel, R.L. Harns, K.B. Welles and W.L. Hinrichs, "Real-time phased array imaging using digital beam forming and autonomous channel control," IEEE Ultrasonics Symposium, 1990, pp. 1499-1502.
 - [108] [O'Donnell91]. M. O'Donnell, W.E. Engeler, J.J. Bloomer and J.T. Pedicone "Method and Apparatus for Digital Phased Array Imaging", U. S. Patent #4,983,970, Ianuarie 8, 1991.
 - [109] [O'Donnell92] M. O'Donnell and L. J. Thomas. "Efficient synthetic aperture imaging from a circular aperture with possible application to catheter-based imaging" IEEE Trans. on Ultrasonics. Ferro. and Freq. Control, Vol 39 No. 3 pp. 366–380, May 1992.
 - [110] [Oppenheim99] A. V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, 1999.
 - [111] [Ozaki 88] Y. Ozaki, H. Sumitami, T. Tomoda and M. Tanaka, "A New System for Real-Time Synthetic Aperture Ultrasonic Imaging." IEEE Trans. Ultrasonics Ferro. and Freq. Control, vol 35, no 6, November 1988, pp. 828-838.
 - [112] [Peterson84] D. K. Peterson and G. S. Kino. "Real-Time Digital Image Reconstruction: A Description of Imaging Hardware and an Analysis of Quantization Errors." IEEE Transactions on Sonics and Ultrasonics, vol. 31 no. 4 pp. 337–351, 1984.
 - [113] [Pridham79] R. G. Pridham and R. A. Mucci "Digital Interpolation Beamforming for Low-Pass and Bandpass Signals", Proceedings of the IEEE, Vol. 67, No. 6 pages 904–919, June, 1979.
 - [114] [Proakis92] J. G. Proakis, D. G. Manolakis, "Digital Signal Processing.", Maxwell MacMillan International Editions 1992
 - [115] [Quistgaard97] J. U. Quistgaard "Signal Acquisition and Processing in Medical Diagnostic Ultrasound." IEEE Signal Processing January 1997, pp 67-74.
 - [116] [Richard94a] W. D. Richard, R. M Arthur, "Real-Time Ultrasonic Scan Conversion via Linear Interpolation on Oversampled Vectors", Ultrasonic Imaging, No. 16, 1994

-
- [117] [Richard94b] W. D. Richard, D. M. Zar, “A CMOS VLSI Linear Interpolation Chip for Ultrasonic Imaging”, 1994
 - [118] [Roelandt78] J. Roelandt, J. W. Wladimiroff and A. M. Bars „Ultrasonic real time imaging with a hand-held scanner: Part II, Initial clinical experience.” *Ultrasound in Med. & Biol.* 4, 1978, pag. 93–97.
 - [119] [Routh96] H. F. Routh “*Doppler Ultrasound: The Ability to Measure and Image Blood Flow.*” *IEEE Engineering in Medicine and Biology*, November / December 1996
 - [120] [Schafer84] M. Schafer and P. Lewin. “*The Influence of Front-End Hardware on Digital Ultrasonic Imaging.*” *IEEE Transactions on Sonics and Ultrasonics*, vol. 31 no. 4 pp. 295–306, July 1984.
 - [121] [Shoup88] T. A. Shoup and J. Hart “*Ultrasonic Imaging Systems*”, *IEEE Ultrasonics Symposium Proceedings*, pages 863–871, 1988.
 - [122] [Shung96] K. K. Shung, and M. Zipparo “*Ultrasound Transducers and Arrays.*” *IEEE Engineering in Medicine and Biology*, November / December 1996
 - [123] [Smith91] S. W. Smith „*High speed ultrasound volumetric imaging system I: Transducer design and beam steering.*” *IEEE Trans. UFFC* 37, 1991, pag. 100–108.
 - [124] [Somer68] J. C. Somer „*Electronic sector scanning for ultrasonic diagnosis.*” *Ultrasonics*, 1968, pag. 153–159.
 - [125] [Song94] T.K. Song and J.F. Greenleaf, “*Ultrasonic Dynamic Focusing Using an Analog FIFO and Asynchronous Sampling,*” *IEEE Trans. on Ultrasonics, Ferro., and Freq. Cont.*, Vol. 41, No. 3, pp. 326–332, May 1994.
 - [126] [Stefanelli00] B. Stefanelli, I. O'Connor, L. Quiquerez, A. Kaiser, and D. Billet. “*An analog beamforming circuit for ultrasound imaging using switched-current delay lines.*” *IEEE Journal of Solid-State Circuits*, 35:202–211, 2000.
 - [127] [Steinberg92] B.D. Steinberg, “*Digital Beamforming in Ultrasound,*” *IEEE Trans. On Ultrasonics, Ferro., and Freq. Cont.*, Vol. 39, No. 6, pp. 716–721, Nov. 1992.
 - [128] [Stewart98] R.W. Stewart and E. Pfann. “*Oversampling and sigma-delta strategies for data conversion.*” *Electronics & Communication Engineering Journal*, 10:37–47, Feb. 1998.
 - [129] [Tanase03] M.E. Tănase, I. Lie, B. Marinca, *Digital Controller for Ultrasonic Beamforming*, *Kurzfassungen der Vorträge*, pg. 24–25. „Kolloquium de Automatisierungstechnik“, 07–08 November 2003, Salzhausen, Germany
 - [130] [Thomenius96] K.E. Thomenius, “*Evolution of Ultrasound Beamformers*” *Proc. IEEE Ultrasonics Symposium*, *IEEE Cat. No. 96CH35993*, 1996, pp. 1615–1622.
 - [131] [Tomov01] B.G. Tomov and J. A. Jensen, “*A new architecture for a single-chip multi-channel beamformer based on a standard FPGA*” *Proceedings of the IEEE International Ultrasonics Symposium*, Atlanta, USA, pages 1529–1533, 2001.
 - [132] [Trahey92a] Gregg E. Trahey and Levin F. Nock. „*Synthetic receive aperture imaging with phase correction for motion and for tissue inhomogeneities — par I Basic principles.*” *IEEE Tran. Ultrasonics, Ferro. and Freq. Cont.*, 39 (4) 489–495, 1992.
 - [133] [Trahey92b] Gregg E. Trahey and Levin F. Nock. „*Synthetic receive aperture imaging with phase correction for motion and for tissue inhomogeneities — par II Effects of and correction for motion.*” *IEEE Tran. Ultrasonics, Ferro. and Freq. Cont.*, 39 (4) 496–501, 1992.
 - [134] [Tutwiler98] R. L. Tutwiler, “*Ultrasonic Beamforming Architectures*” *SPIE Medical Imaging*, 1998.
 - [135] [VanAken84] J.R. Van Aken, “*An efficient ellipse-drawing algorithm*” *IEEE Journal of Computer Graphics and Application*, Vol. 4. No. 9, 1984, pp. 24–35.
 - [136] [vonRamm91] O. T. von Ramm “*High speed ultrasound volumetric imaging system II: Parallel processing and image display.*” *IEEE Trans. UFFC* 38, 1991, pag. 109–115.

-
- [137] [vonRamm75] O. T. von Ramm and F. L. Thurstone „*Thaumascan: Design considerations and performance characteristics.*” *Ultrasound in Med* 1, 1975, pag. 373–378.
- [138] [vonRamm83] O.T. von Ramm and S.W. Smith, “*Beam Steering with Linear Arrays,*” *IEEE Trans. On Biomedical Engineering*, Vol. BME-30, No. 8, August 1983.
- [139] [Wells69a] P. N. T. Wells „*Physical Principles of Ultrasonic Diagnosis.*” Academic Press, London. 1969.
- [140] [Wells69b] P. N. T. Wells „*A range-gated ultrasonic Doppler system.*” *Med. Biol. Eng.* 7, 1969, pag. 641–652.
- [141] [Wells77] P. N. T. Wells „*Biomedical Ultrasonics.* Academic Press, London. 1977.
- [142] [Williams91] L.A. Williams and B.A. Wooley, “*Third-Order Cascaded Sigma-Delta Modulators.*” *IEEE Trans. On Circuits and Systems*, Vol. 38, No. 5, pp. 489-497, May 1991.
- [143] [Zar94] D. M Zar, W. D. Richard “*A Scan Conversion Engine for Standard B-Mode*” *Ultrasonic Imaging*, 1994

ANEXA 1

MODELAREA MATLAB A FORMATOARELOR DE FASCICUL

- A 1.1 FFTrFDReConvIMG - FORMATOR CONVENTIONAL DE FASCICUL CU FOCALIZARE FIXĂ LA TRANSMISIE ȘI FOCALIZARE DINAMICĂ LA RECEPȚIE**
- A 1.2 FFTrFDRecDSIMG1 - FORMATOR DE FASCICUL CU MODULAȚIE DELTA-SIGMA UNIFORMĂ, FOCALIZARE FIXĂ LA TRANSMISIE SI FOCALIZARE DINAMICĂ LA RECEPȚIE**
- A 1.3 FFTrFDRecDSneunifIMG1 - FORMATOR DE FASCICUL CU MODULAȚIE DELTA-SIGMA NEUNIFORMĂ, FOCALIZARE FIXĂ LA TRANSMISIE SI FOCALIZARE DINAMICĂ LA RECEPȚIE**

A 1.1 FFTrFDRConvIMG - FORMATOR CONVENTIONAL DE FASCICUL CU FOCALIZARE FIXA LA TRANSMISIE ȘI FOCALIZARE DINAMICĂ LA RECEPȚIE

```
% calculul intarzierilor + focalizare dinamica - varianta cu efectul
FNumber!!
% Testat functional in 05.01.2003 - pentru fasciculul 93 (contine maximul
pentru firul 3 la esantionul 606 _ FD conventionala )
% focalizare dinamica conventional
% 27.01.2003 - modificat pentru FF la emisie si FD la receptie imagine
clear all

TOff=0.000029448; % offsetul 29,448us
%TOff=0.0; % offsetul 29,448us
NS=2048; % numarul de esantioane dintr-o inregistrare
interpft=1; % factroul de interpolare
no_samp=2048*interpft; % numarul de esantioane dintr-o inregistrare
c=1500; % viteza us
NR=128; % numarul elementelor ariei
FTus= 3500000; % frecventa traductorului
lambda=c/FTus; % lungimea de unda
ID=lambda/2 ; % spatiul interelement
SF = 13889000; % frecventa de esantionare
TS = 1/13889000; % perioada de esantionare
RD = 1536*2; % numarul punctelor focale (512 in Yao)
NBeam = 201; % numarul fasciculelor (201 in Yao)
sinPI4=sin(pi/4);
fcsz = TS/(2*interpft);
sbeam=2*sinPI4/(NBeam-1)
IDsec = ID/c % distanta interelement in unitati de timp

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pentru focalizare fixa
Depth = 60e-3; % adancimea de focalizare in m
fdepth = Depth/c; % adancimea de focalizare in sec
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% time gain compensation
tgc = 0; % 1 - da

% datele de intrare
DateIN = 1

fff='elem1'; % 'elem2';
[fid,message] = fopen(fff,'r','n')
B = zeros (1,1*2048*128);
Y = zeros (1,1*2048*128);
[A,count] = fread(fid,[4096*128*1], 'uint8');
for i=1:2:4096*128*1
    if A(i)>=128
        B((i+1)/2)=-1*(256*(255-A(i))+(255-A(i+1)));
    else
        B((i+1)/2)=A(i)*256+A(i+1);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Filtrare date intrare - filtru Kaiser

b = fir1(128,[0.23 0.67], kaiser(129,7));
C = filter(b,1,B);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
img_in = zeros (1, 128, 2048);
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:1
    for i=1:128 % elementul curent 1...N
        for j=1:2048 % esantionul curent 1...M
            img_in1(k,i,j)=C((i-1)*2048+j);% transfer vectorul linie cu
128*2048 valori intr-o matrice cu
                                % 128 linii (nr elementelor ariei) si 2048
coloane (nr esantioane) *8 (pentru 8 elemente)
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculul fasciculului
calculfascicul = 1

elem_start=1; % elementele active
elem_stop=128; % elementele active
a0=0.53836; % hamminng window
a1=0.46164; % hamminng window
FNTr = 2 % FNumber pentru emisie FN = R/D R=distanța focală, D=diametrul
aperturii
FNRe = 2 % FNumber pentru receptie

for rd=1:1:RD*interpft
    %rdafisat=rd
    r=fsize*rd;
    RRDD=rd % afisare functionare
    %Trnr_start = fix((elem_start+elem_stop-1)/2);
    %Trnr_stop = Trnr_start+1;

    %Renr_start = fix(NR/2);
    %Renr_stop = Renr_start+1;

    %elem = fix(r/(2*FNTr*IDsec));
    %Trnr_start = Trnr_start-elem;
    %Trnr_stop = Trnr_stop+elem;

    %if (Trnr_start<elem_start)
    %    Trnr_start=elem_start;
    %end

    %if (Trnr_stop>elem_stop)
    %    Trnr_stop=elem_stop;
    %end

    %elem = fix(r/(2*FNRe*IDsec));
    %Renr_start = Renr_start-elem;
    %Renr_stop = Renr_stop+elem;

    %if (Renr_start<1)
    %    Renr_start=elem_start;
    %end

    %if (Renr_stop>NR)
    %    Renr_stop=elem_stop;
    %end

    %N= Renr_stop-Renr_start;
    %for i=1:1:N
    %    hamming(i)=a0-a1*cos(2*pi*i/N);
    %end

    Trnr_start = 1;

```

```

Trnr_stop = 1;

% Apodizare fnumber la receptie

Renr_start = fix((elem_start+elem_stop-1)/2);
Renr_stop = Renr_start+1;

elem = fix(r/(2*FNRe*IDsec));
Renr_start = Renr_start-elem;
Renr_stop = Renr_stop+elem;

if (Renr_start<elem_start)
    Renr_start=elem_start;
end

if (Renr_stop>elem_stop)
    Renr_stop=elem_stop;
end

for beam =1:1:NBeam %beam = 1:1:NBeam
%for beam = 93:93 %beam = 1:1:NBeam
    sum1=0;
%    beam = 93;

    for elemTr = Trnr_start:1:Trnr_stop

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      adaugat 21.01      - calcul intarziere pentru
elementul emitator - focalizare fixa
        tmp1 = fdepth*fdepth;
        tmp2 = 2*fdepth;
        stheta = (beam-1)*sbeam-sinPI4;
        tmp3 = stheta*tmp2;
        tmp4T = (-NR/2+elemTr-0.5)*IDsec;
        b1T=(tmp1+tmp4T*(tmp4T-tmp3))^0.5;
        FdeltaTr=fdepth-b1T;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end adaugat 07.12
        for elemRe = Renr_start:1:Renr_stop
            if ((elemTr~=elemRe)&(elemRe~=1)&(elemRe~=2))
%                if ((elemRe~=1)&(elemRe~=2))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%      adaugat 07.12      - calcul intarziere pentru
elementul receptor tmp1, tmp2 si tmp3 sunt identice la TR si REC
                tmp1R = r*r;
                tmp2R = 2*r;
                stheta = (beam-1)*sbeam-sinPI4;
                tmp3R = stheta*tmp2R;
                tmp4R = (-NR/2+elemRe-0.5)*IDsec;
                b1R=(tmp1R+tmp4R*(tmp4R-tmp3R))^0.5;
                DdeltaRe=r-b1R;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end adaugat 07.12
                sdelay = (2*r-TOff-FdeltaTr-DdeltaRe)*SF*interpft;
                s_low=floor(sdelay);
                s_high=s_low+1;
                s_weight=sdelay-s_low;
                %end
                if ( (sdelay > 1) & (sdelay < no_samp-1))
%                    sum=sum+((img_in(elemTr,elemRe,s_high)-
img_in(elemTr,elemRe,s_low))*s_weight+(img_in(elemTr,elemRe,s_low))*hamming
(elemRe-Renr_start+1));
%                    sum=sum+((img_in1(elemTr,elemRe,s_high)-
img_in1(elemTr,elemRe,s_low))*s_weight+(img_in1(elemTr,elemRe,s_low))*1);
                    sum1=sum1+img_in1(elemTr,elemRe,s_low);
                end
            end
        end
    end
end

```

```

        end
    end
end
% polar(rd, beam)=sum;
% polar1(rd, beam)=sum1;

% polar(rd)=sum;
polar2(rd,beam)=abs(sum1);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% scanconverter
scanconverter=1
fcsz = TS/2
size = 512;
img_out = zeros (size, size);
img_lengd = RD*TS/2;
size = 512;
pix_size = img_lengd / size;
q=pix_size / fcsz;
tmp1=0.5*TOff;
tmp3=0.5*NS/SF + tmp1;
if (tmp3>img_lengd)
    tmp3 = img_lengd
end
for y=1:1:size
    for x=1:1:size
        avstx = -size/2 -0.5 + x;
        avsty = y-1;
        avstr = (avstx*avstx+avsty*avsty)^0.5;
        tmp2= avstr*pix_size;
        if ((tmp2 >= tmp1) &(tmp2 <= tmp3))
            stheta = avstx/avstr;
            r = avstr*q;
            b = (stheta + sinPI4)/sbeam +1;
            %end
            if ((r > 1) & (r < (RD-1)) & (b > 1) & (b < (NBeam-1)) )
                r_low = floor(r) ;
                r_high = r_low + 1 ;
                r_weight = r-r_low ;
                b_low = floor(b) ;
                b_high = b_low + 1 ;
                b_weight = b-b_low ;

                % interpolare biliniara
                p1 = abs (polar2(r_low, b_high));
                p2 = abs (polar2(r_low, b_low));
                p3 = abs (polar2(r_high, b_high));
                p4 = abs (polar2(r_high, b_low));

                p_low= (p1-p2)*b_weight+p2;
                p_high= (p3-p4)*b_weight+p4;
                %if (tgc = 1) % time gain compensation
                %
                img_out (y,x) = ((p_high -
p_low)*r_weight+p_low)*TGC(tmp2);
                %else
                img_out (y,x) = (p_high - p_low)*r_weight+p_low;
                %end
            end
        end
    end
end
end
figure

```

```

mesh (img_out)

mddinamic=1

% modificare domeniul dinamic
%size = 400;
maxval = max(max(img_out));
range = 70;
img = zeros (size, size);
for y=1:1:size
    for x=1:1:size
        if(img_out(y,x)==0)
            tmp = 20*log10(1/maxval);
        else
            tmp = 20*log10((abs(img_out (y,x)))/maxval);
        end

        if(tmp > -range)
            img(y,x) = tmp;
        else
            img(y,x) = - range;
        end
    end
end
figure
mesh (img)

[fid1,message] = fopen('polarFFTrFDRecConv','w','n')
count = fwrite(fid1, polar2, 'int32');
status = fclose(fid1)

```


A 1.2 FFTrFDRecDSIMG1 - FORMATOR DE FASCICUL CU MODULAȚIE DELTA-SIGMA UNIFORMĂ, FOCALIZARE FIXA LA TRANSMISIE SI FOCALIZARE DINAMICĂ LA RECEPȚIE

```
% calculul intarzierilor + focalizare dinamica - varianta cu efectul
FNumber!!
% 26 01 2003 - imagine pentru un element - FFTrFDRecDS Esantionare
Uniforma - corectat
clear all

TOff=0.000029448; % offsetul 29,448us
NS=2048; % numarul de esantioane dintr-o inregistrare
interpft=8; % factroul de interpolare
no_samp=2048*interpft; % numarul de esantioane dintr-o inregistrare
c=1500; % viteza us
NR=128; % numarul elementelor ariei
FTus= 3500000; % frecventa traductorului
lambda=c/FTus; % lungimea de unda
ID=lambda/2 ;% spatiul interelement
SF = 13889000; % frecventa de esantionare
TS = 1/13889000; % perioada de esantionare
RD = 1536*2; % numarul punctelor focale (512 in Yao)
NBeam = 200; % numarul fasciculelor (201 in Yao)
sinPI4=sin(pi/4);
fcsize = TS/(2*interpft);
sbeam=2*sinPI4/(NBeam-1)
IDsec = ID/c % distanta interelement in unitati de timp
% time gain compensation
tgc = 0; % 1 - da

FNTr = 2 % FNumber pentru emisie FN = R/D R=distanta focala, D=diametrul
aperturii
FNRe = 2 % FNumber pentru receptie
elem_start=1; % elementele active
elem_stop=128; % elementele active

% parametrii convertorului sigma delta
OSR = 16; % OSR - factorul de supraesantionare
ORDIN = 2; % - ordinul convertorului sigma delta

% parametrii pentru focalizare fixa pe un arc de cerc de raza "Depth"

Depth = 60e-3; % adancimea de focalizare in m
fdepth = Depth/c; % adancimea de focalizare in sec

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculul intarzierilor
%calculDelay = 1

%Ddelta = zeros (RD*interpft, NBeam, NR);

%for rd=1:1:RD*interpft
%   r=fcsize*rd;
%   tmp1 = r*r;
%   tmp2 = 2*r;
%   for beam = 1:1:NBeam %beam = 1:1:NBeam
%       stheta = (beam-1)*sbeam-sinPI4;
%       tmp3 = stheta*tmp2;
%       for elem =1:1:NR
%           tmp4 = (-NR/2+elem-0.5)*IDsec;
%           b1=(tmp1+tmp4*(tmp4-tmp3))^0.5;
%           Ddelta(rd,beam,elem)=r-b1;
%       end
%   end
```

```

%      end
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% datele de intrare
DateIN = 1

fff='elem1';          % 'elem2';
[fid,message] = fopen(fff,'r','n')
B = zeros (1,1*2048*128);
A = zeros (1,1*4096*128);
[A,count] = fread(fid,[4096*128*1], 'uint8');
for i=1:2:4096*128*1
    if A(i)>=128
        B((i+1)/2)=-1*(256*(255-A(i))+(255-A(i+1)));
    else
        B((i+1)/2)=A(i)*256+A(i+1);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Filtrare date intrare - filtru Kaiser

b = fir1(128,[0.23 0.67], kaiser(129,7));
C = filter(b,1,B);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:1
    for i=1:128 % elementul curent 1...N
        for j=1:2048 % esantionul curent 1...M
            img_in(k,i,j)=C((i-1)*2048+j);% transfer vectorul linie cu
128*2048 valori intr-o matrice cu
                                % 128 linii (nr elementelor ariei) si 2048
coloane (nr esantioane) *8 (pentru 8 elemente)
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%A. Interpolare
interpolare=1
BBB = zeros (1,128,2048*interpft);
deltastep = 1;
k=1;
for i = 1 :128
    for j = 1:2047
        val1 = img_in(k,i,j);
        val2 = img_in(k,i,j+1);
        step = (val2 - val1)/(interpft);
        BBB(k,i,interpft*(j-1)+1) = val1;
        for n = 1:interpft-1,
            %      inval = (round((val1+step*n)/deltastep)) * deltastep;
            inval = val1+step*n;
            BBB(k,i,interpft*(j-1)+n+1) = inval;
        end
    end
end
BBB (k,i,interpft*j+1) = val2;

% B. Normalizare Date
k=1;
for i = 1 :128
    BBBN(1,i, :) = 0.5*(BBB(1,i, :)/max(abs(BBB(1,i, :)))));

```

```

end
% C. Conversie Delta Sigma
ConvSigmaDelta=1
H = synthesizenTF(ORDIN,OSR,1);
k=1;
for i = 1 :128
img_in1(1,i, :) = simulateDSM(BBBN(1,i, :),H);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear BBB ;
clear BBBN ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% afisare pentru verificare
%for i = 1 :2048*interpft
%vv1(i) = 1000*v(1,4, i);
%end
%stairs (vv1, 'r')
%hold on

%for i = 1 :2048*interpft
%vv2(i) = BBB(1,4,i);
%end
%plot (vv2, 'g')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculul fasciculului
calculfascicul = 1

elem_start=1; % elementele active
elem_stop=128; % elementele active
a0=0.53836; % hamminng window
a1=0.46164; % hamminng window
FNTr = 2 % FNumber pentru emisie FN = R/D R=distanța focală, D=diametrul
aperturii
FNRe = 2 % FNumber pentru receptie

for rd=1:1:RD*interpft
    %rdafisat=rd
    r=fcsiz*rd;
    RRDD=rd % afisare functionare
    %Trnr_start = fix((elem_start+elem_stop-1)/2);
    %Trnr_stop = Trnr_start+1;

    %Renr_start = fix(NR/2);
    %Renr_stop = Renr_start+1;

    %elem = fix(r/(2*FNTr*IDsec));
    %Trnr_start = Trnr_start-elem;
    %Trnr_stop = Trnr_stop+elem;

    %if (Trnr_start<elem_start)
    %    Trnr_start=elem_start;
    %end

    %if (Trnr_stop>elem_stop)
    %    Trnr_stop=elem_stop;
    %end

    %elem = fix(r/(2*FNRe*IDsec));
    %Renr_start = Renr_start-elem;
    %Renr_stop = Renr_stop+elem;

    %if (Renr_start<1)
    %    Renr_start=elem_start;

```

```

%end

%if (Renr_stop>NR)
%   Renr_stop=elem_stop;
%end

%N= Renr_stop-Renr_start;
%for i=1:1:N
%   hamming(i)=a0-a1*cos(2*pi*i/N);
%end
Trnr_start = 1;
Trnr_stop = 1;
% Apodizare fnumber la receptie
Renr_start = fix((elem_start+elem_stop-1)/2);
Renr_stop = Renr_start+1;
elem = fix(r/(2*FNRe*IDsec));
Renr_start = Renr_start-elem;
Renr_stop = Renr_stop+elem;
if (Renr_start<elem_start)
    Renr_start=elem_start;
end
if (Renr_stop>elem_stop)
    Renr_stop=elem_stop;
end
for beam =1:1:NBeam %beam = 1:1:NBeam
    sum1=0;
    for elemTr = Trnr_start:1:Trnr_stop
    %%%%%%%%%%%%%% focalizare fixa la emisie
        tmp1T = fdepth*fdepth;
        tmp2T = 2*fdepth;
        stheta = (beam-1)*sbeam-sinPI4;
        tmp3T = stheta*tmp2T;
        tmp4T = (-NR/2+elemTr-0.5)*IDsec;
        b1T=(tmp1T+tmp4T*(tmp4T-tmp3T))^0.5;
        FdeltaTr=fdepth-b1T;
    %%%%%%%%%%%%%%
        for elemRe = Renr_start:1:Renr_stop
            if ((elemTr~=elemRe)&(elemRe~=1)&(elemRe~=2))
%
                if ((elemRe~=1)&(elemRe~=2))

    %%%%%%%%%%%%%%focalizare dinamica la receptie
                    tmp1R = r*r;
                    tmp2R = 2*r;
                    stheta = (beam-1)*sbeam-sinPI4;
                    tmp3R = stheta*tmp2R;
                    tmp4R = (-NR/2+elemRe-0.5)*IDsec;
                    b1R=(tmp1R+tmp4R*(tmp4R-tmp3R))^0.5;
                    DdeltaRe=r-b1R;
    %%%%%%%%%%%%%%
                    sdelay = (2*r-TOff-FdeltaTr-DdeltaRe)*SF*interpft;
                    s_low=floor(sdelay);
                    s_high=s_low+1;
                    s_weight=sdelay-s_low;
                    %end
                    if ( (sdelay > 1) & (sdelay < no_samp-1))
                        sum1=sum1+img_in1(elemTr,elemRe,s_low);
                    end
                    end
                end
            end
        end
        polar(rd, beam)=sum1;
    end
end
end

```

```

figure
for i = 1 :RD*interpft
pp1(i) = polar(i,93);
end
plot (pp1, 'r')
hold on

% filtrare + decimare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
filtraredecimare=1
bb = fir1(128, 0.1, kaiser(129,5));
polar1 = filter(bb,1,polar); % filtrare
%Y3=Y2(64:length(Y2)); % elimina intarzierea de grup

for j = 1:NBeam
    for i = 1 :RD
        polar2(i,j) = polar1(i*interpft, j) ;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% afisare pentru verificare
figure
for i = 1 :RD
pp1(i) = polar2(i,93);
end
plot (pp1, 'r')
hold on

for i = 1 :RD
pp2(i) = polar2(i,109);
end
plot (pp2, 'g')

%pause
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% introdus in 28.01.2003
polar2A = abs (polar2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% scanconverter
scanconverter=1
fcsz = TS/2
size = 512;
img_out = zeros (size, size);
img_lengd = RD*TS/2;
size = 512;
pix_size = img_lengd / size;
q=pix_size / fcsz;
tmp1=0.5*TOff;
tmp3=0.5*NS/SF + tmp1;
if (tmp3>img_lengd)
    tmp3 = img_lengd
end
for y=1:1:size
    for x=1:1:size
        avstx = -size/2 -0.5 + x;
        avsty = y-1;
        avstr = (avstx*avstx+avsty*avsty)^0.5;
        tmp2= avstr*pix_size;
        if ((tmp2 >= tmp1) &(tmp2 <= tmp3))
            stheta = avstx/avstr;
            r = avstr*q;
            b = (stheta + sinPI4)/sbeam +1;
            %end
            if ((r > 1) & (r < (RD-1)) & (b > 1) & (b < (NBeam-1)) )

```

```

        r_low = floor(r) ;
        r_high = r_low + 1 ;
        r_weight = r-r_low ;
        b_low = floor(b) ;
        b_high = b_low + 1 ;
        b_weight = b-b_low ;

        % interpolare biliniara
        p1 = abs (polar2A(r_low, b_high));
        p2 = abs (polar2A(r_low, b_low));
        p3 = abs (polar2A(r_high, b_high));
        p4 = abs (polar2A(r_high, b_low));

        p_low= (p1-p2)*b_weight+p2;
        p_high= (p3-p4)*b_weight+p4;
        %if (tgc = 1) % time gain compensation
        %img_out (y,x) = ((p_high -
p_low)*r_weight+p_low)*TGC(tmp2);
        %else
        img_out (y,x) = (p_high - p_low)*r_weight+p_low;
        %end
    end
end
end
figure
mesh (img_out)
title ('FFTrFDRcdS, filtrare cu 0.1')

mddinamic=1

% modificare domeniul dinamic
%size = 400;
maxval = max(max(img_out));
range = 70;
img = zeros (size, size);
for y=1:1:size
    for x=1:1:size
        if(img_out(y,x)==0)
            tmp = 20*log10(1/maxval);
        else
            tmp = 20*log10((abs(img_out (y,x)))/maxval);
        end

        if(tmp > -range)
            img(y,x) = tmp;
        else
            img(y,x) = - range;
        end
    end
end
figure
mesh (img)
title ('FFTrFDRcdS, 70dB DD, filtrare cu 0.1')

[fid1,message] = fopen('polarFFTrFDRcdS1','w','n')
count = fwrite(fid1, polar, 'int16');
status = fclose(fid1)

%fff='polarFFTrFDRcdS1'; % 'elem2';
%[fid,message] = fopen(fff,'r','n')
%[polar,count] = fread(fid,[24576,200], 'int16');

```

A 1.3 FFTrFDRecDSneunifIMG1 - FORMATOR DE FASCICUL CU MODULAȚIE DELTA-SIGMA NEUNIFORMĂ, FOCALIZARE FIXA LA TRANSMISIE SI FOCALIZARE DINAMICĂ LA RECEPȚIE

```
% calculul intarzierilor + focalizare dinamica - varianta cu efectul
FNumber!!
% Testat functional in 05.01.2003 - pentru fasciculul 93 (contine maximul
pentru firul 3 la esantionul 606 _ FD conventionala )
% focalizare dinamica conventional
% 24.01.2003 - esantionare neuniforma (delta sigma)
clear all

TOff=0.000029448; % offsetul 29,448us
%TOff=0.0; % offsetul 29,448us
NS=2048; % numarul de esantioane dintr-o inregistrare
interpft=8; % factroul de interpolare
no_samp=2048*interpft; % numarul de eșantioane dintr-o inregistrare
c=1500; % viteza us
NR=128; % numarul elementelor ariei
FTus= 3500000; % frecventa traductorului
lambda=c/FTus; % lungimea de unda
ID=lambda/2 ;% spatiul interelement
SF = 13889000; % frecventa de esantionare
TS = 1/13889000; % perioada de esantionare
RD = 1536*2; % numarul punctelor focale (512 in Yao)
NBeam = 201; % numarul fasciculelor (201 in Yao)
sinPI4=sin(pi/4);
fcsiz = 1.00*TS/(2*interpft) ;
sbeam=2*sinPI4/(NBeam-1)
IDsec = ID/c % distanta interelement in unitati de timp

% time gain compensation
tgc = 0; % 1 - da

FNTr = 2 % FNumber pentru emisie FN = R/D R=distanta focala, D=diametrul
aperturii
FNRe = 2 % FNumber pentru receptie
elem_start=1; % elementele active
elem_stop=128; % elementele active

% parametrii convertorului sigma delta
OSR = 16; % OSR - factorul de supraesantionare
ORDIN = 2; % - ordinul convertorului sigma delta

% parametrii pentru focalizare fixa pe un arc de cerc de raza "Depth"

Depth = 60e-3; % adancimea de focalizare in m
fdepth = Depth/c; % adancimea de focalizare in sec

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% datele de intrare
DateIN = 1

fff='elem1'; % 'elem2';
[fid,message] = fopen(fff,'r','n')
B = zeros (1,1*2048*128);
Y = zeros (1,1*2048*128);
[A,count] = fread(fid,[4096*128*1], 'uint8');
for i=1:2:4096*128*1
    if A(i)>=128
        B((i+1)/2)=-1*(256*(255-A(i))+(255-A(i+1)));
    else
```

```

        B((i+1)/2)=A(i)*256+A(i+1);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Filtrare date intrare - filtru Kaiser

b = fir1(128,[0.23 0.67], kaiser(129,7));
C = filter(b,1,B);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
img_in = zeros (1, 128, 2048);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:1
    for i=1:128 % elementul curent 1...N
        for j=1:2048 % esantionul curent 1...M
            img_in(k,i,j)=C((i-1)*2048+j);
            % transfer vectorul linie cu 128*2048 valori intr-o matrice cu
            % 128 linii (nr elementelor ariei) si 2048 coloane (nr
            % esantioane) *8 (pentru 8 elemente)
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%A. Interpolare
interpft=1
BBB = zeros (1,128,2048*interpft);

deltastep = 1;
k=1;
for i = 1 :128
    for j = 1:2047
        val1 = img_in(k,i,j);
        val2 = img_in(k,i,j+1);
        step = (val2 - val1)/(interpft);
        BBB(k,i,interpft*(j-1)+1) = val1;
        for n = 1:interpft-1,
            %      inval = (round((val1+step*n)/deltastep)) * deltastep;
            inval = val1+step*n;
            BBB(k,i,interpft*(j-1)+n+1) = inval;
        end
    end
end
BBB (k,i,interpft*j+1) = val2;

% B. Normalizare Date
k=1;
for i = 1 :128
    BBBN(1,i, :) = 0.5*(BBB(1,i, :)/max(abs(BBB(1,i, :)))));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculul fasciculului
calculfascicul = 1

elem_start=1; % elementele active
elem_stop=128; % elementele active
a0=0.53836; % hamminng window
a1=0.46164; % hamminng window
FNTr = 2 % FNumber pentru emisie FN = R/D R=distanța focală, D=diametrul
aperturii
FNRe = 2 % FNumber pentru receptie
NUS = zeros (RD*interpft, NR );

for rd=1:1:RD*interpft
    %rdafisat=rd

```

```

r=fcsiz*rd;
RRDD=rd % afisare functionare
%Trnr_start = fix((elem_start+elem_stop-1)/2);
%Trnr_stop = Trnr_start+1;

%Renr_start = fix(NR/2);
%Renr_stop = Renr_start+1;

%elem = fix(r/(2*FNTr*IDsec));
%Trnr_start = Trnr_start-elem;
%Trnr_stop = Trnr_stop+elem;

%if (Trnr_start<elem_start)
%    Trnr_start=elem_start;
%end

%if (Trnr_stop>elem_stop)
%    Trnr_stop=elem_stop;
%end

%elem = fix(r/(2*FNRe*IDsec));
%Renr_start = Renr_start-elem;
%Renr_stop = Renr_stop+elem;

%if (Renr_start<1)
%    Renr_start=elem_start;
%end

%if (Renr_stop>NR)
%    Renr_stop=elem_stop;
%end

%N= Renr_stop-Renr_start;
%for i=1:1:N
%    hamming(i)=a0-a1*cos(2*pi*i/N);
%end

Trnr_start = 1;
Trnr_stop = 1;

% Apodizare fnumber la receptie

Renr_start = fix((elem_start+elem_stop-1)/2);
Renr_stop = Renr_start+1;

elem = fix(r/(2*FNRe*IDsec));
Renr_start = Renr_start-elem;
Renr_stop = Renr_stop+elem;

if (Renr_start<elem_start)
    Renr_start=elem_start;
end

if (Renr_stop>elem_stop)
    Renr_stop=elem_stop;
end

for beam =1:1:NBeam %beam = 1:1:NBeam
%for beam = 93:93 %beam = 1:1:NBeam
%    sum=0;
%    sum1=0;
%beam = 93;

for elemTr = Trnr_start:1:Trnr_stop

```

```

%focalizare fixa la emisie
    tmp1T = fdepth*fdepth;
    tmp2T = 2*fdepth;
    stheta = (beam-1)*sbeam-sinPI4;
    tmp3T = stheta*tmp2T;
    tmp4T = (-NR/2+elemTr-0.5)*IDsec;
    b1T=(tmp1T+tmp4T*(tmp4T-tmp3T))^0.5;
    Fdelta=fdepth-b1T;

%
    %tmp1 = r*r;
    %tmp2 = 2*r;
    %stheta = (beam-1)*sbeam-sinPI4;
    %tmp3 = stheta*tmp2;
    %tmp4T = (-NR/2+elemTr-0.5)*IDsec;
    %b1T=(tmp1+tmp4T*(tmp4T-tmp3))^0.5;
    %DdeltaTr=r-b1T;

%
    for elemRe = Renr_start:1:Renr_stop
        if ((elemTr~=elemRe)&(elemRe~=1)&(elemRe~=2))
%
            if ((elemRe~=1)&(elemRe~=2))

% focalizare dinamica la receptie
                tmp1 = r*r;
                tmp2 = 2*r;
                stheta = (beam-1)*sbeam-sinPI4;
                tmp3 = stheta*tmp2;
                tmp4R = (-NR/2+elemRe-0.5)*IDsec;
                b1R=(tmp1+tmp4R*(tmp4R-tmp3))^0.5;
                DdeltaRe=r-b1R;

%
                sdelay = (2*r-TOff-Fdelta-DdeltaRe)*SF*interpft;
                s_low=floor(sdelay);
                s_high=s_low+1;
                s_weight=sdelay-s_low;
                %end
                if ( (sdelay > 1) & (sdelay < no_samp-1))
%
                    sum=sum+((img_in1(elemTr,elemRe,s_high)-
img_in1(elemTr,elemRe,s_low))*s_weight+(img_in1(elemTr,elemRe,s_low))*1);
%
                    sum1=sum1+img_in1(elemTr,elemRe,s_low);
                    NUS(rd, elemRe)= BBBN(1, elemRe, s_low );
                end
            end
        end
    end
    polar(rd, beam)=sum;
    polar1(rd, beam)=sum1;

%
    polar(rd)=sum;
    polar1(rd)=sum1;
end
end

%polar = sum(NUS');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% C. Conversie Delta Sigma
ConvSigmaDelta=1
H = synthesizNTF(ORDIN,OSR,1);
for i = 1 :NR
    NUSDS(:, i) = simulateDSM(NUS(:, i),H);
end

% afisare pentru verificare
% spectrul semnalului convertit sigma-delta

```

```

for i = 1 :2048*interpft
vv1(i) = NUSDS(i, 64);
end

%figure
%Fs = 13890000*interpft;
%[Pxx,f] = pwelch(vv1,257,256,[],Fs,'onesided');
%plot (f,20*log10(Pxx/max(Pxx)))

%stairs (vv1, 'r')
%hold on

%bb = fir1(128, 0.1, kaiser(129,5));
%p21 = filter(bb,1,vv1); % filtrare
%plot (p21)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sumare fascicul
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

polar = sum(NUSDS');
figure
plot (polar)

% spectrul demnalului la iesirea formatorului (inainte de filtrare)
figure

Fs = 13890000*interpft;
[Pxx,f] = pwelch(polar,513,512,[],Fs,'onesided');
plot (f,20*log10(Pxx/max(Pxx)))
title('FFTrFDRec Sigma-Delta Neuniform Ordin2 OSR16')
xlabel('Frecventa [Hz]')
ylabel('Amplitudinea [dB]')

% filtrare + decimare
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

filtraredecimare=1
bb = fir1(128, 0.2, kaiser(129,5));
polar21 = filter(bb,1,polar); % filtrare

%for j = 1:NBeam
    for i = 1 :RD
        %polar4(i,j) = polar21(i*interpft, j) ;
        %polar5(i,j) = polar31(i*interpft, j) ;
        polar4(i) = polar21(i*interpft) ;
    end
%end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% afisare pentru verificare

%for ii=1:1:RD
%    BBBDS(ii)=polar4(ii,2);
%end
figure

plot(polar4)
title('FFTrFDRec Sigma-Delta Neuniform Ordin2 OSR16 - Semnalul de iesire')
xlabel('Numar esantion')
ylabel('Amplitudine')
% afisare pentru verificare
%for ii=1:1:RD
%    BBBDS1(ii)=polar5(ii,2);

```

```

%end

%b = fir1(128,[0.2 0.75], kaiser(129,7));
%BBBF = filter(b,1,BBBDs);
%BBBF1 = filter(b,1,BBBDs1);

figure
Fs = 13890000
[Pxx,f] = pwelch(polar4,513,512,[],Fs,'onesided');
plot (f,20*log10(Pxx/max(Pxx)))
title('FFTrFDRec Sigma-Delta Neuniform Ordin2 OSR16 - Spectrul semnalului
de iesire ')
xlabel('Frecventa [Hz]')
ylabel('Amplitudinea [dB]')

figure
plot(20*log10(abs(polar4)/max(abs(polar4))))
title('FFTrFDRec Sigma-Delta Neuniform Ordin2 OSR16 - Semnalul de iesire')
xlabel('Frecventa [Hz]')
ylabel('Amplitudine [dB]')

```

A 1.4 DIMENSIONAREA FILTRULUI ȘI CUANTIFICAREA COEFICIENȚILOR

```
% caracteristica modulatorului - varianta pentru dimensionarea filtrului
% influenta numarului de biti asupra atenuarii in banda de blocare
R=16; % factorul de supraesantionare (OSR)
H = synthesizNTF(2,R,1); % functia de transfer a zgomotului
fr=5000000
pas=1/(2*fr*R)
t = 0:pas:10000*pas;
x = 0.5*sin(2*pi*3500000*t);
plot(x(1:100))
title('Signal 3.5Mhz')
xlabel('time (seconds)')

v = simulateDSM(x,H); % simulare modulator delta-sigma
echo off;

figure(1); %clf;
t = 0:100;
stairs(t, x(t+1), 'r');
hold on;
stairs(t, v(t+1), 'g');
axis([0 100 -1.2 1.2]);
xlabel('Sample Number');
ylabel('u, v');
title('Modulator Input & Output');

N = 2*fr*R;
N = 1001;
Y = fft(v,N);
Pyy = Y.* conj(Y) / N;
figure
plot(linspace(0,0.5*160e6,N/2), dbv(Pyy(1:N/2)))
axis([0 0.5*160e6 -120 0]);
%text(0.25, -90, sprintf('BW=%7.5f',1.5/N));
grid on;
xlabel('Frequency')
ylabel('dBFS')
title('Output SpectrumBB');

%f = 100000000*(0:256)/512;
%plot(f,Pyy(1:257))
%title('Frequency content of y')
%xlabel('frequency (Hz)')

N = 2*fr*R;
N = 10001;
echo on
spec = fft(v.*hann(N))/(N/4);
echo off;
figure
figure(2); clf;
plot(linspace(0,0.5*160e6,N/2), dbv(spec(1:N/2)))
axis([0 0.5*160e6 -120 0]);
text(0.25, -90, sprintf('BW=%7.5f',1.5/N));
grid on;
xlabel('Frequency')
ylabel('dBFS')
title('Output SpectrumAA');

F=[4.5e6;6.5e6]
A=[1;0]
dp=0.1
ds=-80
```

```

Dev=[ (10^(dp/20))-1;10^(ds/20)]
fs=160e6
[N,Fo,Ao,W]=remezord(F,A,Dev,fs)
Na=255
h=remez(Na,Fo,Ao,W);
H=fft(h,127);

B=13;
hq1=(round(2^B*h))/(2^B);
B=15;
hq2=(round(2^B*h))/(2^B);
B=21;
hq3=(round(2^B*h))/(2^B);

[h1, f] = freqz(hq1,1,10001,fs);% coeficienti cuantificati pe 13 biti
[h2, f] = freqz(hq2,1,10001,fs);% coeficienti cuantificati pe 15 biti
[h3, f] = freqz(hq3,1,10001,fs);% coeficienti cuantificati pe 21 biti
hold on
s.plot = 'mag'; % Plot the magnitude only.
s.xunits = 'hz'; % Label the frequency units correctly.
s.yunits = 'dB'; % Plot the magnitude dB.
[h, f] = freqz(h,1,10001,fs);% coeficienti necuantificati
hh = [h1 h2 h3];
freqzplot(hh,f,s); % plot frequency characteristic

% spectrul semnalului filtrat
N=10001
hold on
zz = filter(hq, 1, v);
spec = fft(zz.*hann(N))/(N/4);
echo off;
%figure(2); clf;
plot(linspace(0,0.5*160e6,N/2), dbv(spec(1:N/2)), 'r')
axis([0 0.5*160e6 -120 0]);
%text(0.25, -90, sprintf('BW=%7.5f',1.5/N));
grid on;

Ak=-ds
deltak=1
B = log2((1+(10^(Ak/20))/(10^(deltak/20)-1))*(sqrt((2*Na+1)/3)))

As = -20*log10(10^(-Ak/20)+(1/(2^22-1))*sqrt((2*Na+1)/3))

```

ANEXA 2

FORMATOR DE FASCICUL CU 128 CANALE
Secțiunea digitală – descriere VHDL

A2.1 SECTIUNEA DIGITALA A FORMATORULUI

Structura si conexiunile exterioare

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY Stratix128c IS PORT (
    O : OUT std_logic_vector(19 DOWNTO 0);
    Data : IN std_logic_vector(127 DOWNTO 0);
    Adr : IN std_logic_vector(8 DOWNTO 0);
    Slk : OUT std_logic_vector(127 DOWNTO 0);
    FSB : IN std_logic_vector(21 DOWNTO 0);
    Start1 : IN std_logic;
    We : IN std_logic;
    Ce : IN std_logic;
    Reset1 : IN std_logic;
    Stop : IN std_logic;
    Reset : IN std_logic;
    Clk : IN std_logic;
    Start : IN std_logic;
    Sel: IN std_logic
); END Stratix128c;

ARCHITECTURE STRUCTURE OF Stratix128c IS

-- COMPONENTS

Component CanaleX128 IS PORT (
    Sclk : OUT std_logic_vector(127 DOWNTO 0);
    Sadr : IN std_logic_vector(6 DOWNTO 0);
    OutAdr : IN std_logic_vector(13 DOWNTO 0);
    Data : IN std_logic_vector(127 DOWNTO 0);
    Q : OUT std_logic_vector(127 DOWNTO 0);
    FSB : IN std_logic_vector(21 DOWNTO 0);
    As : IN std_logic;
    We : IN std_logic;
    Bs : IN std_logic;
    CE : IN std_logic;
    Reset : IN std_logic;
    InClk : IN std_logic;
    IDs : IN std_logic;
    Clk : IN std_logic;
    Start : IN std_logic;
    Sel: IN std_logic
); END Component;

COMPONENT fir_filter
PORT (
    Adr_out : OUT std_logic_vector(13 DOWNTO 0);
    Clk : IN std_logic;
    Din : IN std_logic_vector(127 DOWNTO 0);
    O : OUT std_logic_vector(19 DOWNTO 0);
    Reset : IN std_logic;
    Start : IN std_logic;
    Stop : IN std_logic
); END COMPONENT;

COMPONENT decoder3
PORT (
    Adr : IN std_logic_vector(1 DOWNTO 0);
    A : OUT std_logic;
    B : OUT std_logic;
    I : OUT std_logic
); END COMPONENT;
```

```

COMPONENT xClk2
    PORT (
        Clk : IN std_logic;
        Reset : IN std_logic;
        Clk2 : OUT std_logic
    ); END COMPONENT;

-- SIGNALS

SIGNAL As : std_logic;
SIGNAL Bs : std_logic;
SIGNAL IDs : std_logic;
SIGNAL Clk2 : std_logic;
SIGNAL D : std_logic_vector(127 DOWNTO 0);
SIGNAL OUTADR : std_logic_vector(13 DOWNTO 0);

-- GATE INSTANCES

BEGIN

U1 : canalex128    PORT MAP (
    Sclk => Slk,
    Sadr => Adr(6 downto 0),
    OutAdr => OUTADR,
    Data => Data,
    Q => D,
    FSB => FSB,
    As => As,
    We => We,
    Bs => We,
    CE => Ce,
    Reset => Reset,
    InClk => Clk2,
    IDs => IDs,
    Clk => Clk,
    Start => Start,
    Sel => Sel
);

U2 : fir_filter    PORT MAP (
    Adr_out => OUTADR,
    Clk => Clk2,
    Din => D,
    O => O,
    Reset => RESET1,
    Start => START1,
    Stop => STOP
);

U3 : decoder3      PORT MAP (
    Adr(1) => ADR(8),
    Adr(0) => ADR(7),
    A => As,
    B => Bs,
    I => IDs
);

U4 : xClk2    PORT MAP (
    Clk => CLK,
    Reset => RESET1,
    Clk2 => Clk2
);
END STRUCTURE;

```

A2.2 STRUCTURA SI INTERCONEXIUNILE CANALELOR

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
use ieee.numeric_std.all;

ENTITY CanaleX128 IS PORT (
    Sclk : OUT std_logic_vector(127 DOWNTO 0);
    Sadr : IN std_logic_vector(6 DOWNTO 0);
    OutAdr : IN std_logic_vector(13 DOWNTO 0);
    Data : IN std_logic_vector(127 DOWNTO 0);
    Q : OUT std_logic_vector(127 DOWNTO 0);
    FSB : IN std_logic_vector(21 DOWNTO 0);
    As : IN std_logic;
    We : IN std_logic;
    Bs : IN std_logic;
    CE : IN std_logic;
    Reset : IN std_logic;
    InClk : IN std_logic;
    IDs : IN std_logic;
    Clk : IN std_logic;
    Start : IN std_logic;
    Sel: IN std_logic
); END CanaleX128;

ARCHITECTURE STRUCTURE OF CanaleX128 IS

-- COMPONENTS

Component Canal Port (
    OutAdr : IN std_logic_vector(13 DOWNTO 0);
    Ain : IN std_logic_vector(13 DOWNTO 0);
    IDin : IN std_logic_vector(13 DOWNTO 0);
    Bin : IN std_logic_vector(21 DOWNTO 0);
    Data : IN std_logic;
    Q : OUT std_logic;
    As : IN std_logic;
    We : IN std_logic;
    Bs : IN std_logic;
    Cs : IN std_logic;
    Reset : IN std_logic;
    InClk : IN std_logic;
    IDs : IN std_logic;
    Clk : IN std_logic;
    Sclk : OUT std_logic;
    Start : IN std_logic;
    Sel: IN std_logic
); END component;

component decoder
port(
    SAdr :in unsigned (6 downto 0);
    CE:in std_logic;
    CS : out unsigned (127 downto 0)
);
end component;
-- SIGNALS

SIGNAL CS : std_logic_vector(127 DOWNTO 0);

-- GATE INSTANCES

BEGIN
```

```

U128 : decoder PORT MAP (
    SAdr => unsigned(SADR),
    CE => CE,
    std_logic_vector(CS) => CS
);

U0 : Canal PORT MAP (
    OutAdr => OUTADR,
    Ain => FSB(13 downto 0),
    IDin => FSB(13 downto 0),
    Bin => FSB,
    Data => DATA(0),
    Q => Q(0),
    As => AS,
    We => WE,
    Bs => BS,
    Cs => CS(0),
    Reset => RESET,
    InClk => INCLK,
    IDs => IDS,
    Clk => CLK,
    Sclk => SCLK(0),
    Start => START,
    Sel => Sel
);

U1 : Canal PORT MAP (
    OutAdr => OUTADR,
    Ain => FSB(13 downto 0),
    IDin => FSB(13 downto 0),
    Bin => FSB,
    Data => DATA(1),
    Q => Q(1),
    As => AS,
    We => WE,
    Bs => BS,
    Cs => CS(1),
    Reset => RESET,
    InClk => INCLK,
    IDs => IDS,
    Clk => CLK,
    Sclk => SCLK(1),
    Start => START,
    Sel => Sel
);

U2 : Canal PORT MAP (
    OutAdr => OUTADR,
    Ain => FSB(13 downto 0),
    IDin => FSB(13 downto 0),
    Bin => FSB,
    Data => DATA(2),
    Q => Q(2),
    As => AS,
    We => WE,
    Bs => BS,
    Cs => CS(2),
    Reset => RESET,
    InClk => INCLK,
    IDs => IDS,
    Clk => CLK,
    Sclk => SCLK(2),
    Start => START,
    Sel => Sel
);

```

```

.....
.....
.....

U63 : Canal PORT MAP (
  OutAdr => OUTADR,
  Ain => FSB(13 downto 0),
  IDin => FSB(13 downto 0),
  Bin => FSB,
  Data => DATA(63),
  Q => Q(63),
  As => AS,
  We => WE,
  Bs => BS,
  Cs => CS(63),
  Reset => RESET,
  InClk => INCLK,
  IDs => IDS,
  Clk => CLK,
  Sclk => SCLK(63),
  Start => START,
  Sel => Sel
);

process(clk)
begin
  for i in natural range 64 to 127 loop
    SCLK(i) <= '0';
  end loop;
end process;

END STRUCTURE;

```

A2.3 STRUCTURA UNUI CANAL

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY Canal IS PORT (
    OutAdr : IN std_logic_vector(13 DOWNTO 0);
    Ain : IN std_logic_vector(13 DOWNTO 0);
    IDin : IN std_logic_vector(13 DOWNTO 0);
    Bin : IN std_logic_vector(21 DOWNTO 0);
    Data : IN std_logic;
    Q : OUT std_logic;
    As : IN std_logic;
    We : IN std_logic;
    Bs : IN std_logic;
    Cs : IN std_logic;
    Reset : IN std_logic;
    InClk : IN std_logic;
    IDs : IN std_logic;
    Clk : IN std_logic;
    Sclk : OUT std_logic;
    Start : IN std_logic;
    Sel: IN std_logic
); END canal;

ARCHITECTURE STRUCTURE OF canal IS

-- COMPONENTS

COMPONENT Scg_Comm
    PORT (
        Clk : IN std_logic;
        Start : IN std_logic;
        Reset : IN std_logic;
        Rdy : IN std_logic;
        Go : IN std_logic;
        Sign : IN std_logic;
        Ld : OUT std_logic;
        Calc : OUT std_logic;
        Sclk : OUT std_logic;
        Down : OUT std_logic
    ); END COMPONENT;

COMPONENT Sign_calculator
    PORT (
        Ao : IN std_logic_vector(13 DOWNTO 0);
        Bo : IN std_logic_vector(21 DOWNTO 0);
        Clk : IN std_logic;
        Load : IN std_logic;
        MSB : OUT std_logic
    ); END COMPONENT;

COMPONENT R_counter
    PORT (
        Data : IN std_logic_vector(13 DOWNTO 0);
        Ld : IN std_logic;
        Clk : IN std_logic;
        Down : IN std_logic;
        Clkscg : IN std_logic;
        Go : OUT std_logic;
        Rdy : OUT std_logic;
        We : OUT std_logic;
        Adr_out : OUT std_logic_vector(13 DOWNTO 0)
    ); END COMPONENT;
```

```

COMPONENT xClk2
  PORT (
    Clk : IN std_logic;
    Reset : IN std_logic;
    Clk2 : OUT std_logic
  ); END COMPONENT;

COMPONENT Reg
  PORT (
    Ain : IN std_logic_vector(13 DOWNTO 0);
    Bin : IN std_logic_vector(21 DOWNTO 0);
    IDin : IN std_logic_vector(13 DOWNTO 0);
    As : IN std_logic;
    Bs : IN std_logic;
    IDs : IN std_logic;
    We : IN std_logic;
    Cs : IN std_logic;
    Aout : OUT std_logic_vector(13 DOWNTO 0);
    Bout : OUT std_logic_vector(21 DOWNTO 0);
    IDout : OUT std_logic_vector(13 DOWNTO 0)
  ); END COMPONENT;

COMPONENT Ram
  PORT (
    DATA : IN std_logic;
    INADDR : IN std_logic_vector(13 DOWNTO 0);
    OUTADDR : IN std_logic_vector(13 DOWNTO 0);
    WE : IN std_logic;
    INCLOCK : IN std_logic;
    OUTCLOCK : IN std_logic;
    Q : OUT std_logic
  ); END COMPONENT;

-- SIGNALS

SIGNAL we_con : std_logic;
SIGNAL sclk_con : std_logic;
SIGNAL LOAD : std_logic;
SIGNAL CALCULATE : std_logic;
SIGNAL DOWN : std_logic;
SIGNAL MSB : std_logic;
SIGNAL GO : std_logic;
SIGNAL RDY : std_logic;
SIGNAL CLK2 : std_logic;
SIGNAL ADR : std_logic_vector(13 DOWNTO 0);
SIGNAL A : std_logic_vector(13 DOWNTO 0);
SIGNAL B : std_logic_vector(21 DOWNTO 0);
SIGNAL ID : std_logic_vector(13 DOWNTO 0);
SIGNAL inclk1, inclk2, outclk1, outclk2, q1, q2 : std_logic;

-- GATE INSTANCES

BEGIN

inclk1 <= sclk_con and Sel;
inclk2 <= sclk_con and (not Sel);
outclk1 <= inclk and (not Sel);
outclk2 <= inclk and Sel;
Q <= q1 or q2;
SCLK <= sclk_con;

U1 : Scg_Comm PORT MAP (
  Clk => CLK,
  Start => START,

```

```

        Reset => RESET,
        Rdy => RDY,
        Go => GO,
        Sign => MSB,
        Ld => LOAD,
        Calc => CALCULATE,
        Sclk => sclk_con,
        Down => DOWN
    );
U2 : Sign_calculator    PORT MAP (
    Ao => A,
    Bo => B,
    Clk => CALCULATE,
    Load => LOAD,
    MSB => MSB
);
U3 : R_counter    PORT MAP (
    Data => ID,
    Ld => LOAD,
    Clk => CLK2,
    Down => DOWN,
    Clkscg => sclk_con,
    Go => GO,
    Rdy => RDY,
    We => we_con,
    Adr_out => ADR
);
U4 : xClk2    PORT MAP (
    Clk => CLK,
    Reset => RESET,
    Clk2 => CLK2
);
U5 : Reg    PORT MAP (
    Ain => AIN,
    Bin => BIN,
    IDin => IDIN,
    As => AS,
    Bs => BS,
    IDs => IDS,
    We => WE,
    Cs => CS,
    Aout => A,
    Bout => B,
    IDout => ID
);
U6 : Ram    PORT MAP (
    DATA => DATA,
    INADR => ADR,
    OUTADR => OUTADR,
    WE => we_con,
    INCLOCK => inclk1,
    OUTCLOCK => outclk1,
    Q => q1
);
U7 : Ram    PORT MAP (
    DATA => DATA,
    INADR => ADR,
    OUTADR => OUTADR,
    WE => we_con,
    INCLOCK => inclk2,
    OUTCLOCK => outclk2,
    Q => q2
);
END STRUCTURE;

```

A2.3.1 ENTITATEA Scg_Comm

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

entity Scg_Comm is
  port(
    Clk : in std_logic;
    Start, Reset, Rdy, Go, Sign: in std_logic;
    Ld, Calc, Sclk, Down: out std_logic
  );
end Scg_Comm;

architecture behavior of Scg_Comm is
  type state_type is (waiting, load, delay, active, delay1, delay2,
compute);
  signal current_state, next_state : state_type;
begin
  -- register block
  process (Clk, Reset)
  begin
    if Reset = '1' then
      current_state <= waiting;
    elsif Clk = '1' and Clk'event then
      current_state <= next_state;
    end if;
  end process;

  -- state machine process
  process (current_state, Start, Rdy, Go, Sign)
  begin
    -- default outputs
    Ld <= '0';
    Calc <= '1';
    Sclk <= '0';
    Down <= '0';

    case current_state is

      when waiting =>
        -- transition conditions
        if Start = '1' then next_state <= load;
        else next_state <= waiting;
        end if;

      when load =>
        -- outputs
        Ld <= '1';
        -- transition conditions
        next_state <= delay;

      when delay =>
        -- outputs
        Down<='1';
        -- transition conditions
        if Go = '1' then next_state <= active;
        else next_state <= delay;
        end if;

      when active =>
        -- outputs
        Sclk <= '1';
        -- transitions
        if Sign = '1' then next_state <= delay1;
```

```

        else next_state <= compute;
        end if;

    when delay1 =>
        -- outputs
        -- transition conditions
        next_state <= delay2;

    when delay2 =>
        -- outputs
        -- transition conditions
        next_state <= compute;

    when compute =>
        -- outputs
        Calc<='0';
        -- transition conditions
        if Rdy='1' then next_state<=waiting;
        else next_state<=active;
        end if;

    when others =>
        next_state <= waiting;

    end case;
end process;
end behavior;

```

A2.3.2 ENTITATEA Sign_Calculator

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

entity Sign_Calculator is
    port(
        Ao:std_logic_vector(13 downto 0);
        Bo:std_logic_vector(21 downto 0);
        Clk,Load:in std_logic;
        MSB:out std_logic
    );
end Sign_Calculator;

architecture behavior of Sign_Calculator is
    --signals
    signal A:signed(13 downto 0);
    signal B:signed(21 downto 0);
    signal d:signed(21 downto 0);
    signal s,zero:std_logic;
    --components
    --atributs

begin
    MSB<= s;
    zero<='1' when d="0000000000000000000000" else
        '0';
    process(Clk,Load,A,B,d,Ao,Bo)
        constant C1:signed(13 downto 0):="00000000010000";--dec 16
        constant C1b:signed(21 downto 0):="0000000000000000010000";--dec 16
        constant C2:signed(21 downto 0) :="0000000000000000011000";--dec 24
    begin
        if Load='1' then

```

```

        A<=signed(Ao);
        B<=signed(Bo);
        d<="000000000000000000000000";
        s<='1';
    elsif clk='0' and clk'event then
        if d(17)='1' or zero='1' then
            d<=d+A;
        else
            d<=d+B;
        end if;
    elsif clk='1' and clk'event then
        if s='1' then
            B<=B-C1b;
        else
            A<=A-C1;
            B<=B-C2;
        end if;
        s<=std_logic(d(17)) or zero;
    end if;
end process;
end behavior;

```

A2.3.3 ENTITATEA R_Counter

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.NUMERIC_STD.all;

entity R_Counter is
    port(
        Data:in std_logic_vector(13 downto 0);
        Ld,Clk,Down,Clkscg:in std_logic;
        Go,We:out std_logic;
        Rdy:out std_logic;
        Adr_out:out std_logic_vector(13 downto 0)
    );
end R_Counter;

architecture behavior of R_Counter is
    signal r:unsigned(13 downto 0);
    signal rd:std_logic;
begin
    We<=(not Down) and (not Rd);
    Rdy<=rd;
    Go<='1' when r="0000000000000000" else
        '0';
    process(Clk,Data,Down,Ld,r,Clkscg)
        constant c:unsigned(13 downto 0):="11111111111111";
    begin
        if Ld='1' then
            r<=unsigned(Data);
            Rd<='0';
        elsif Down='1' then
            if Clk='1' and Clk'event then
                r<=r-1;
            end if;
        else
            if Clkscg='1' and Clkscg'event then
                r<=r+1;
                if r=c then
                    Rd<='1';
                end if;
            end if;
        end if;
    end process;
end behavior;

```

```

        end if;
    end if;
end process;
Adr_out<=std_logic_vector(r);
end behavior;

```

A2.3.4 ENTITATEA xClk2

```

library ieee;
use ieee.std_logic_1164.all;
use IEEE.NUMERIC_STD.all;

entity xClk2 is
    port(
        Clk, Reset: in std_logic;
        Clk2 :out std_logic
    );
end xClk2;

architecture behavior of xClk2 is
    signal c:unsigned(1 downto 0);
begin
    Clk2<=std_logic(c(1));
    process (Clk, Reset)
    begin
        if Reset='1' then
            c<="00";
        elsif Clk='1' and Clk'event then
            c<=c+1;
        end if;
    end process;
end behavior;

```

A2.3.5 ENTITATEA reg

```

library ieee;
use ieee.std_logic_1164.all;
use IEEE.NUMERIC_STD.all;

entity reg is
    port(
        Ain:in std_logic_vector(13 downto 0);
        Bin:in std_logic_vector(21 downto 0);
        IDin:in std_logic_vector(13 downto 0);
        As,Bs,IDs,We,Cs:in std_logic;
        Aout:out std_logic_vector(13 downto 0);
        Bout:out std_logic_vector(21 downto 0);
        IDout:out std_logic_vector(13 downto 0)
    );
end reg;

architecture behavior of reg is
begin
    process (Ain, Bin, Idin, As, Bs, Ids, We, Cs)
    begin
        if cs='1' then

            if As='1' and we='1' then
                Aout<=Ain;
            end if;

```

```

        if Bs='1' and we='1' then
            Bout<=Bin;
        end if;

        if IDs='1' and we='1' then
            IDout<=IDin;
        end if;

    end if;
end process;
end behavior;

```

A2.3.6 ENTITATEA RAM

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
LIBRARY lpm;
USE lpm.lpm_components.ALL;

entity RAM is
    port (
        DATA:in std_logic;
        INADR,OUTADR:in std_logic_vector(13 downto 0);
        WE,INCLOCK,OUTCLOCK:in std_logic;
        Q:out std_logic
    );
end RAM;

architecture behavior of RAM is

    signal d:std_logic_vector(1 downto 0);
    signal qout:std_logic_vector(1 downto 0);
    signal rdaddr:std_logic_vector(12 downto 0);
    signal wraddr:std_logic_vector(12 downto 0);

    COMPONENT ram_mem512
        PORT
        (
            data          : IN STD_LOGIC_VECTOR (1 DOWNTO 0);
            wren           : IN STD_LOGIC      := '1';
            wraddress      : IN STD_LOGIC_VECTOR (12 DOWNTO 0);
            rdaddress      : IN STD_LOGIC_VECTOR (12 DOWNTO 0);
            inclock        : IN STD_LOGIC ;
            outclock       : IN STD_LOGIC ;
            q               : OUT STD_LOGIC_VECTOR (1 DOWNTO 0)
        );
    END COMPONENT;

begin
    11:for i in natural range 13 downto 1 generate
        rdaddr(i-1)<= outadr(i);
    end generate;
    12:for i in natural range 13 downto 1 generate
        wraddr(i-1)<= inadr(i);
    end generate;
    q<=qout(0) when outadr(0)='0' else
        qout(1);
    d(0)<=data when inadr(0)='0' else
        d(0);
    d(1)<=data when inadr(0)='1' else
        d(1);

```

```
inst_1: ram_mem512 PORT MAP(  
  data=>d,  
  wren=>we,  
  wraddress => wraddr,  
  rdaddress => rdaddr,  
  inclock => inclock,  
  outclock => outclock,  
  q => qout  
);  
end behavior;
```

A2.4 FILTRUL FIR

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
ENTITY FIR_Filter IS PORT (
    O : OUT std_logic_vector(19 DOWNT0 0);
    Din : IN std_logic_vector(127 DOWNT0 0);
    Adr_out : OUT std_logic_vector(13 DOWNT0 0);
    Stop : IN std_logic;
    Reset : IN std_logic;
    Clk : IN std_logic;
    Start : IN std_logic
); END FIR_Filter;
```

ARCHITECTURE STRUCTURE OF FIR_Filter IS

-- COMPONENTS

```
COMPONENT Fir_remez
PORT (
    Din : IN std_logic_vector(7 DOWNT0 0);
    Load : IN std_logic;
    Add : IN std_logic;
    Mul : IN std_logic;
    AddAll : IN std_logic;
    Reset : IN std_logic;
    Enable : IN std_logic;
    Dout : OUT std_logic_vector(19 DOWNT0 0)
); END COMPONENT;
```

```
COMPONENT Fir_com
PORT (
    Start : IN std_logic;
    Stop : IN std_logic;
    Clk : IN std_logic;
    Reset : IN std_logic;
    Enable : OUT std_logic;
    Add : OUT std_logic;
    Mul : OUT std_logic;
    Addall : OUT std_logic
); END COMPONENT;
```

```
COMPONENT add128x1
PORT (
    i : IN std_logic_vector(127 DOWNT0 0);
    o : OUT std_logic_vector(7 DOWNT0 0)
); END COMPONENT;
```

```
COMPONENT fir_adr
PORT (
    reset : IN std_logic;
    clk : IN std_logic;
    en : IN std_logic;
    count : OUT std_logic_vector(13 DOWNT0 0)
); END COMPONENT;
```

-- SIGNALS

```
SIGNAL N00027 : std_logic;
SIGNAL N00017 : std_logic;
SIGNAL N00007 : std_logic;
SIGNAL N00037 : std_logic;
SIGNAL D : std_logic_vector(7 DOWNT0 0);
```

-- GATE INSTANCES

```
BEGIN
U1 : Fir_remez      PORT MAP (
    Din(7) => D(7),
    Din(6) => D(6),
    Din(5) => D(5),
    Din(4) => D(4),
    Din(3) => D(3),
    Din(2) => D(2),
    Din(1) => D(1),
    Din(0) => D(0),
    Load => CLK,
    Add => N00017,
    Mul => N00027,
    AddAll => N00037,
    Reset => RESET,
    Enable => N00007,
    Dout(19) => O(19),
    Dout(18) => O(18),
    Dout(17) => O(17),
    Dout(16) => O(16),
    Dout(15) => O(15),
    Dout(14) => O(14),
    Dout(13) => O(13),
    Dout(12) => O(12),
    Dout(11) => O(11),
    Dout(10) => O(10),
    Dout(9) => O(9),
    Dout(8) => O(8),
    Dout(7) => O(7),
    Dout(6) => O(6),
    Dout(5) => O(5),
    Dout(4) => O(4),
    Dout(3) => O(3),
    Dout(2) => O(2),
    Dout(1) => O(1),
    Dout(0) => O(0)
);
U2 : Fir_com      PORT MAP (
    Start => START,
    Stop => STOP,
    Clk => CLK,
    Reset => RESET,
    Enable => N00007,
    Add => N00017,
    Mul => N00027,
    Addall => N00037
);
U3 : add128x1     PORT MAP (
    i(127) => DIN(127),
    i(126) => DIN(126),
    i(125) => DIN(125),
    i(124) => DIN(124),
    i(123) => DIN(123),
    i(122) => DIN(122),
    i(121) => DIN(121),
    i(120) => DIN(120),
    i(119) => DIN(119),
    i(118) => DIN(118),
    i(117) => DIN(117),
    i(116) => DIN(116),
    i(115) => DIN(115),
```

```
i(114) => DIN(114),
i(113) => DIN(113),
i(112) => DIN(112),
i(111) => DIN(111),
i(110) => DIN(110),
i(109) => DIN(109),
i(108) => DIN(108),
i(107) => DIN(107),
i(106) => DIN(106),
i(105) => DIN(105),
i(104) => DIN(104),
i(103) => DIN(103),
i(102) => DIN(102),
i(101) => DIN(101),
i(100) => DIN(100),
i(99) => DIN(99),
i(98) => DIN(98),
i(97) => DIN(97),
i(96) => DIN(96),
i(95) => DIN(95),
i(94) => DIN(94),
i(93) => DIN(93),
i(92) => DIN(92),
i(91) => DIN(91),
i(90) => DIN(90),
i(89) => DIN(89),
i(88) => DIN(88),
i(87) => DIN(87),
i(86) => DIN(86),
i(85) => DIN(85),
i(84) => DIN(84),
i(83) => DIN(83),
i(82) => DIN(82),
i(81) => DIN(81),
i(80) => DIN(80),
i(79) => DIN(79),
i(78) => DIN(78),
i(77) => DIN(77),
i(76) => DIN(76),
i(75) => DIN(75),
i(74) => DIN(74),
i(73) => DIN(73),
i(72) => DIN(72),
i(71) => DIN(71),
i(70) => DIN(70),
i(69) => DIN(69),
i(68) => DIN(68),
i(67) => DIN(67),
i(66) => DIN(66),
i(65) => DIN(65),
i(64) => DIN(64),
i(63) => DIN(63),
i(62) => DIN(62),
i(61) => DIN(61),
i(60) => DIN(60),
i(59) => DIN(59),
i(58) => DIN(58),
i(57) => DIN(57),
i(56) => DIN(56),
i(55) => DIN(55),
i(54) => DIN(54),
i(53) => DIN(53),
i(52) => DIN(52),
i(51) => DIN(51),
```

```

i(50) => DIN(50),
i(49) => DIN(49),
i(48) => DIN(48),
i(47) => DIN(47),
i(46) => DIN(46),
i(45) => DIN(45),
i(44) => DIN(44),
i(43) => DIN(43),
i(42) => DIN(42),
i(41) => DIN(41),
i(40) => DIN(40),
i(39) => DIN(39),
i(38) => DIN(38),
i(37) => DIN(37),
i(36) => DIN(36),
i(35) => DIN(35),
i(34) => DIN(34),
i(33) => DIN(33),
i(32) => DIN(32),
i(31) => DIN(31),
i(30) => DIN(30),
i(29) => DIN(29),
i(28) => DIN(28),
i(27) => DIN(27),
i(26) => DIN(26),
i(25) => DIN(25),
i(24) => DIN(24),
i(23) => DIN(23),
i(22) => DIN(22),
i(21) => DIN(21),
i(20) => DIN(20),
i(19) => DIN(19),
i(18) => DIN(18),
i(17) => DIN(17),
i(16) => DIN(16),
i(15) => DIN(15),
i(14) => DIN(14),
i(13) => DIN(13),
i(12) => DIN(12),
i(11) => DIN(11),
i(10) => DIN(10),
i(9) => DIN(9),
i(8) => DIN(8),
i(7) => DIN(7),
i(6) => DIN(6),
i(5) => DIN(5),
i(4) => DIN(4),
i(3) => DIN(3),
i(2) => DIN(2),
i(1) => DIN(1),
i(0) => DIN(0),
o(7) => D(7),
o(6) => D(6),
o(5) => D(5),
o(4) => D(4),
o(3) => D(3),
o(2) => D(2),
o(1) => D(1),
o(0) => D(0)
);
U4 : fir_adr      PORT MAP (
    reset => RESET,
    clk   => CLK,
    en    => N00007,

```

```

        count(13) => ADR_OUT(13),
        count(12) => ADR_OUT(12),
        count(11) => ADR_OUT(11),
        count(10) => ADR_OUT(10),
        count(9) => ADR_OUT(9),
        count(8) => ADR_OUT(8),
        count(7) => ADR_OUT(7),
        count(6) => ADR_OUT(6),
        count(5) => ADR_OUT(5),
        count(4) => ADR_OUT(4),
        count(3) => ADR_OUT(3),
        count(2) => ADR_OUT(2),
        count(1) => ADR_OUT(1),
        count(0) => ADR_OUT(0)
    );
END STRUCTURE;

```

A2.4.1 ENTITATEA Fir_remez

```

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

entity Fir_remez is
    port(
        Din:in std_logic_vector(7 downto 0);
        Load,Add,Mul,AddAll,Reset,Enable:in std_logic;
        Dout:out std_logic_vector(19 downto 0)
    );
end Fir_remez;

architecture behavior of Fir_remez is

    --tipuri de date
    --registru de deplasare 128x8 pentru memorarea valorilor de intrare
    type shift_reg is array (0 to 127) of signed(8 downto 0);
    --vectorul coeficientilor de inmultire
    type const_array is array (0 to 63) of signed (19 downto 0);
    --intrari in etajul de insumare a valorilor inmultite cu coeficienti
    simetrici
    type sum_in is array (0 to 127) of signed(9 downto 0);
    --registru de insumare
    type sum_reg is array (0 to 63) of signed(9 downto 0);
    --registru de inmultire
    type mul_reg is array (0 to 63) of signed(29 downto 0);

    --variabile din etajul final de adunare
    --nivelul 1;
    type lev1in is array (0 to 63) of signed(14 downto 0);
    type lev1out is array (0 to 31)of signed(14 downto 0);
    --nivelul 2;
    type lev2in is array (0 to 31) of signed(15 downto 0);
    type lev2out is array (0 to 15)of signed(15 downto 0);
    --nivelul 3;
    type lev3in is array (0 to 15) of signed(16 downto 0);
    type lev3out is array (0 to 7)of signed(16 downto 0);
    --nivelul 4;
    type lev4in is array (0 to 7) of signed(17 downto 0);
    type lev4out is array (0 to 3)of signed(17 downto 0);
    --nivelul 5;
    type lev5in is array (0 to 3) of signed(18 downto 0);
    type lev5out is array (0 to 1)of signed(18 downto 0);

```

```

        --nivelul 6;
        type lev6in is array (0 to 1) of signed(19 downto 0);

--semnale
--registru de deplasare
signal s_reg : shift_reg;
--registru de adunare
signal sum_r : sum_reg;
--registru de inmultire
signal mul_r : mul_reg;
--registru de adunare
signal add_r : signed(19 downto 0);

begin
    --instructiuni concurente
    Dout<=std_logic_vector(add_r);

    --instructiuni secventiale
    process (Din, Load, Add, Mul, AddAll, Reset, Enable, s_reg, sum_r, mul_r)

        --coeficienti de inmultire constanti in hex [ x2^(-22) ]
        constant h : const_array :=(
            x"0002A", x"00040", x"00068", x"00097", x"000C8", x"000F1", x"00103",
            x"000ED",
            x"00097", x"FFFE9", x"FFEC9", x"FFD1F", x"FFADA", x"FF7F1", x"FF467",
            x"FF052",
            x"FEFDA", x"FE73C", x"FE2CA", x"FDEE7", x"FDC06", x"FDA9F", x"FDB26",
            x"FDE05",
            x"FE386", x"FEBC9", x"FF6DE", x"00462", x"013D7", x"02474", x"03530",
            x"044D0",
            x"051F7", x"05B39", x"05F3A", x"05CC9", x"05301", x"04169", x"0280A",
            x"00788",
            x"FE12C", x"FB6E2", x"F8B35", x"F612F", x"F3C3B", x"F1FF7", x"F0FF9",
            x"F0F99",
            x"F21AE", x"F4857", x"F84CA", x"FD725", x"03E5C", x"0B82D", x"14130",
            x"1D4EE",
            x"26E19", x"306CA", x"398CD", x"41DF7", x"4907D", x"4EB46", x"52A35",
            x"54A6B"
        );
        --variabile
        variable sin : sum_in;

        variable l1i:lev1in;
        variable l1o:lev1out;
        variable l2i:lev2in;
        variable l2o:lev2out;
        variable l3i:lev3in;
        variable l3o:lev3out;
        variable l4i:lev4in;
        variable l4o:lev4out;
        variable l5i:lev5in;
        variable l5o:lev5out;
        variable l6i:lev6in;
        variable l6o:signed(19 downto 0);

    --inceput de process
    begin
        --functionare registru de deplasare
        if (Reset = '1') then
            for n in natural range 127 downto 0 loop
                s_reg(n)<="0000000000";
            end loop;
        elsif (Enable='1') then
    
```

```

    if (Load='1' and Load'event) then
        for n in natural range 127 downto 1 loop
            s_reg(n)<=s_reg(n-1);
        end loop;
        for n in natural range 7 downto 0 loop
            s_reg(0)(n)<=din(n);
        end loop;
        s_reg(0)(8)<=din(7);
    end if;
end if;

--primul etaj de insumare
for n in natural range 127 downto 0 loop
    sin(n):=resize(s_reg(n),10);
end loop;
if (Add='1' and Add'event) then
    for n in natural range 63 downto 0 loop
        sum_r(n)<=sin(n)+sin(127-n);
    end loop;
end if;

--etaul de inmultire
if (Mul='1' and Mul'event) then
    for n in natural range 0 to 63 loop
        mul_r(n)<=sum_r(n)*h(n);
    end loop;
end if;

--etajul de adunare 64 --> 1
--nivelul 1;
for n in natural range 0 to 63 loop
    for m in natural range 13 downto 0 loop
        l1i(n)(m):=mul_r(n)(m+16);
    end loop;
    l1i(n)(14):=mul_r(n)(29);
end loop;
for n in natural range 0 to 31 loop
    l1o(n):= l1i(2*n)+l1i(2*n+1);
end loop;
--nivelul 2;
for n in natural range 0 to 31 loop
    l2i(n):=resize(l1o(n),16);
end loop;
for n in natural range 0 to 15 loop
    l2o(n):=l2i(2*n)+l2i(2*n+1);
end loop;
--nivelul 3;
for n in natural range 0 to 15 loop
    l3i(n):=resize(l2o(n),17);
end loop;
for n in natural range 0 to 7 loop
    l3o(n):=l3i(2*n)+l3i(2*n+1);
end loop;
--nivelul 4;
for n in natural range 0 to 7 loop
    l4i(n):=resize(l3o(n),18);
end loop;
for n in natural range 0 to 3 loop
    l4o(n):=l4i(2*n)+l4i(2*n+1);
end loop;
--nivelul 5;
for n in natural range 0 to 3 loop
    l5i(n):=resize(l4o(n),19);
end loop;

```

```

    for n in natural range 0 to 1 loop
        l5o(n):=l5i(2*n)+l5i(2*n+1);
    end loop;
    --nivelul 6;
    for n in natural range 0 to 1 loop
        l6i(n):=resize(l5o(n),20);
    end loop;
    l6o:=l6i(0)+l6i(1);
    -- incarcare registru de iesire
    if (Addall='1' and Addall'event) then
        --for n in natural range 11 downto 0 loop
            add_r<=l6o;
        --end loop;
    end if;
end process;

end behavior;

```

A2.4.2 ENTITATEA Fir_com

```

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

entity Fir_com is
    port(
        Start, Stop : in std_logic;
        Clk, Reset : in std_logic;
        Enable, Add, Mul, Addall : out std_logic
    );
end Fir_com;

architecture behavior of Fir_com is
    type state_type is (waiting, Add_load, del1, del2, del3, del4, del5,
        Mul_load, del6, del7,
        del8, del9, del10, Addall_load, del11, del12, del13);
    signal current_state, next_state : state_type;
begin
    -- register block
    process (Clk, Reset)
    begin
        if Reset = '1' then
            current_state <= waiting;
        elsif Clk = '1' and Clk'event then
            current_state <= next_state;
        end if;
    end process;

    -- state machine process
    process (current_state, Start, Stop) begin
        -- default outputs
        Enable <= '1';
        Add <= '0';
        Mul <= '0';
        Addall<='0';

        case current_state is

            when waiting =>
                -- transition conditions
                if Start = '1' then next_state <= add_load;
                else next_state <= waiting;

```

```

    end if;

when Add_load =>
    -- outputs
    Add <= '1';
    -- transition conditions
    next_state <= del1;

when del1 =>
    -- outputs
    -- transition conditions
    next_state <= del2;

when del2 =>
    -- outputs
    -- transition conditions
    next_state <= del3;

when del3 =>
    -- outputs
    -- transition conditions
    next_state <= del4;

when del4 =>
    -- outputs
    -- transition conditions
    next_state <= del5;

when del5 =>
    -- outputs
    -- transition conditions
    next_state <= Mul_load;

when Mul_load =>
    -- outputs
    Mul<='1';
    -- transition conditions
    next_state <= del6;

when del6 =>
    -- outputs
    -- transition conditions
    next_state <= del7;

when del7 =>
    -- outputs
    -- transition conditions
    next_state <= del8;

when del8 =>
    -- outputs
    -- transition conditions
    next_state <= del9;

when del9 =>
    -- outputs
    -- transition conditions
    next_state <= del10;

when del10 =>
    -- outputs
    -- transition conditions
    next_state <= Addall_load;

```

```

    when Addall_load =>
        -- outputs
        Addall<='1';
        -- transition conditions
        next_state <= del11;

    when del11 =>
        -- outputs
        -- transition conditions
        next_state <= del12;

    when del12 =>
        -- outputs
        -- transition conditions
        if Stop='1' then next_state<= waiting;
        else next_state <= Add_load;
        end if;

    when others =>
        next_state <= waiting;

end case;

end process;

end behavior;

```

A2.4.3 ENTITATEA add128x1

```

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

entity add128x1 is
    port(
        i: in std_logic_vector(127 downto 0);
        o: out std_logic_vector(7 downto 0)
    );
end add128x1;

architecture behavior of add128x1 is

    type 10i is array(127 downto 0) of unsigned(1 downto 0);
    type 10o is array(63 downto 0) of unsigned(1 downto 0);
    type 11i is array(63 downto 0) of unsigned(2 downto 0);
    type 11o is array(31 downto 0) of unsigned(2 downto 0);
    type 12i is array(31 downto 0) of unsigned(3 downto 0);
    type 12o is array(15 downto 0) of unsigned(3 downto 0);
    type 13i is array(15 downto 0) of unsigned(4 downto 0);
    type 13o is array(7 downto 0) of unsigned(4 downto 0);
    type 14i is array(7 downto 0) of unsigned(5 downto 0);
    type 14o is array(3 downto 0) of unsigned(5 downto 0);
    type 15i is array(3 downto 0) of unsigned(6 downto 0);
    type 15o is array(1 downto 0) of unsigned(6 downto 0);
    type 16i is array(1 downto 0) of unsigned(7 downto 0);

begin
    process(i)

        variable i0:10i;
        variable o0:10o;
        variable i1:11i;
        variable o1:11o;
    
```

```

variable i2:12i;
variable o2:12o;
variable i3:13i;
variable o3:13o;
variable i4:14i;
variable o4:14o;
variable i5:15i;
variable o5:15o;
variable i6:16i;
variable o6:unsigned(7 downto 0);

begin

  for n in natural range 127 downto 0 loop
    if i(n)='1' then
      i0(n):="01";
    else
      i0(n):="00";
    end if;
  end loop;
  for n in natural range 63 downto 0 loop
    o0(n):=i0(2*n+1)+i0(2*n);
  end loop;

  for n in natural range 63 downto 0 loop
    i1(n):=resize(o0(n),3);
  end loop;
  for n in natural range 31 downto 0 loop
    o1(n):=i1(2*n+1)+i1(2*n);
  end loop;

  for n in natural range 31 downto 0 loop
    i2(n):=resize(o1(n),4);
  end loop;
  for n in natural range 15 downto 0 loop
    o2(n):=i2(2*n+1)+i2(2*n);
  end loop;

  for n in natural range 15 downto 0 loop
    i3(n):=resize(o2(n),5);
  end loop;
  for n in natural range 7 downto 0 loop
    o3(n):=i3(2*n+1)+i3(2*n);
  end loop;

  for n in natural range 7 downto 0 loop
    i4(n):=resize(o3(n),6);
  end loop;
  for n in natural range 3 downto 0 loop
    o4(n):=i4(2*n+1)+i4(2*n);
  end loop;

  for n in natural range 3 downto 0 loop
    i5(n):=resize(o4(n),7);
  end loop;
  for n in natural range 1 downto 0 loop
    o5(n):=i5(2*n+1)+i5(2*n);
  end loop;

  for n in natural range 1 downto 0 loop
    i6(n):=resize(o5(n),8);
  end loop;
  o6 := i6(1)+i6(0);
  o<=std_logic_vector(o6-"01000000");

```

```
    end process;  
end behavior;
```

A2.4.4 ENTITATEA fir_adr

```
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.numeric_std.all;  
  
entity fir_adr is  
    port(reset, clk, en : in std_logic;  
          count : out std_logic_vector(13 downto 0)  
    );  
end;  
  
architecture behavior of fir_adr is  
    signal count_n: unsigned(13 downto 0);  
begin  
    process (reset, clk, count_n,en)  
    begin  
        if reset='1' then  
            count_n <= "0000000000000000";  
        elsif en='1' then  
            if (clk='1' and clk'event) then  
                count_n <= count_n + "1";  
            end if;  
        end if;  
    end process;  
    count <= std_logic_vector(count_n);  
end;
```