

Caracteristicile sistemelor de prelucrare digitală a semnalelor .

Majoritatea aplicațiilor de procesare a semnalelor (Digital Signal Processing - DSP) necesită o prelucrare în timp real cu o întârziere minimă între momentele de timp ale achiziției semnalului de intrare și generarea semnalului de ieșire. Pentru aceste aplicații datele de intrare nu pot fi stocate (pentru a fi prelucrate " off line") deci procesorul trebuie să fie suficient de rapid pentru a se evita pierderea de date.

Pe de altă parte algoritmi DSP sînt caracterizați de paralelism în ceea ce privește organizarea calculelor; acest paralelism poate fi exploatat de arhitectura procesorului de semnal .

Vedere generală asupra unui procesor de semnal

În figura 1. este prezentată o configurație generică a unui procesor de semnal constituit din: unitatea de procesare, memorie, unitatea de transfer a datelor, procesorul de intrare / ieșire și procesorul de control.

În structură pot exista mai multe unități de prelucrare distribuite, cu memorii private sau comune. Transferul datelor și conflictele de acces la memorie pot constitui elemente de reducere a vitezei globale de operare a procesorului. Pentru evitarea acestui fenomen se utilizează proiectarea modularizată, cu partajarea sarcinilor între unitățile de prelucrare. Acest lucru este valabil atît pentru procesoarele de semnal nou create cît și pentru structurile de prelucrare cu mai multe procesoare integrate interconectate.

Procesorul de intrare - ieșire (I/O) asigură preluarea semnalului de intrare și generarea semnalului de ieșire (filtrare de bandă, conversii, stocare a datelor, protocolul de comunicație cu exteriorul).

Memoria este caracterizată prin: capacitatea de stocare a datelor, lărgimea cuvîntului de date, modurile de adresare a datelor, timpul de acces).

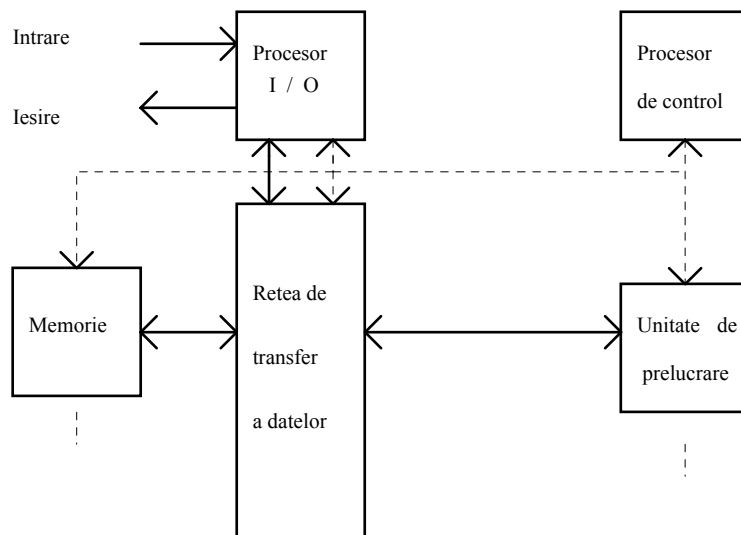


Figura 1 Procesor de semnal generic

Rețeaua de transfer a datelor asigură transmiterea datelor către toate unitățile funcționale ale procesorului de semnal (transmisie de date, rutarea datelor, controlul asupra erorilor).

Procesorul de control este caracterizat de: capacitatea de stocare a instrucțiunilor, viteza de execuție, setul de instrucțiuni.

Unitatea de prelucrare asigură efectuarea operațiilor de bază (operații aritmetice, logice) și este caracterizată de: timpul de execuție a instrucțiunilor, gradul de flexibilitate.

Funcțiile procesorului de control sînt: interpretarea comenzilor primite din exterior, controlul transferului de date, controlul unităților de prelucrare și controlul interfețelor de intrare-ieșire.

Aceste funcții pot fi distribuite și către alte elemente ale sistemului. De exemplu unitățile de prelucrare au posibilitatea de a controla local transferul datelor și funcțiile aritmetice - logice proprii.

Procesorul de control poate efectua prelucrări complementare aplicației DSP (de uz general care nu au cerințe critice de timp: monitorizare, întreținere etc.)

Resursele procesorului de control includ capacitatea de stocare a datelor și instrucțiunilor, setul de instrucțiuni, capabilitatea aritmetică.

Elementele de prelucrare execută algoritmul DSP în timp real. Aceste elemente sînt specializate în efectuarea prelucrărilor specifice algoritmilor DSP.

Rețeaua de transfer a datelor reprezintă un bloc esențial în sistem; acest bloc poate avea influențe majore în viteza de prelucrare globală a sistemului.

Resursele rețelei de transfer a datelor includ: căile de rutare a datelor, lățimea busurilor de transfer a datelor, viteza de transmisie, posibilitatea de transfer pe blocuri de date. Cerințele de control a rețelei de transfer includ protocoalele de comunicație și modurile de adresare. Numărul de busuri reprezintă o problemă care diminuează capacitatea de transfer a datelor între blocurile din figura 1; nu este convenabilă realizarea unei interconectări totale între blocuri datorită complexității tehnologice și a necesității introducerii unor blocuri suplimentare de arbitraj a cererilor de acces multiple.

Resursele memoriilor includ: capacitatea de stocare, lățimea cuvintului de date, modurile de adresare și controlul general al memoriei (decodificare, comenzi).

Unități aritmetice și de control pentru procesoare DSP

Unitatea aritmetică poate fi organizată în 3 moduri prezentate mai jos:

1. Unitate aritmetică fără multiplicator (numai sumator și shifter); este prezentată în figura 2.

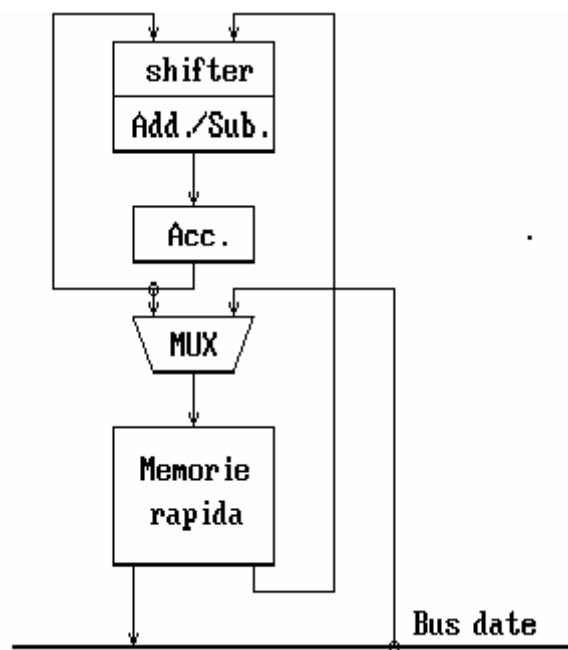


Figura 2.

Notății: Acc - acumulator
 Mux - multiplexor

În acest caz înmulțirea se realizează prin deplasări și adunări (conform unui algoritm secvențial de înmulțire implementat prin program); soluția este foarte economică, dar de viteză mica.

2. Unitate aritmetică cu multiplicator și sumator

a) structura în cascada (figura 3 a)

Structura este potrivită pentru sume de produse; se pot efectua înmulțiri rapide. Timpul de efectuare al unei sume de produs (înmulțire și acumulare la suma anterioară) este destul de mare. Pentru îmbunătățirea performanțelor acestei structuri se pot folosi în paralel înmulțitorul și sumatorul. Multiplicatorul se poate folosi în mod secvențial (se efectuează câte un produs apoi se face acumulara rezultatului) sau în timp ce se efectuează un produs se acumulează rezultatul anterior (utilizarea în paralel a multiplicatorului și a sumatorului).

Dezavantajele acestei structuri sînt: efectuarea sumelor simple se face cu utilizarea suplimentară a multiplicatorului (ceea ce mărește timpul de calcul) și efectuarea produselor multiple se face cu utilizarea suplimentară a sumatorului.

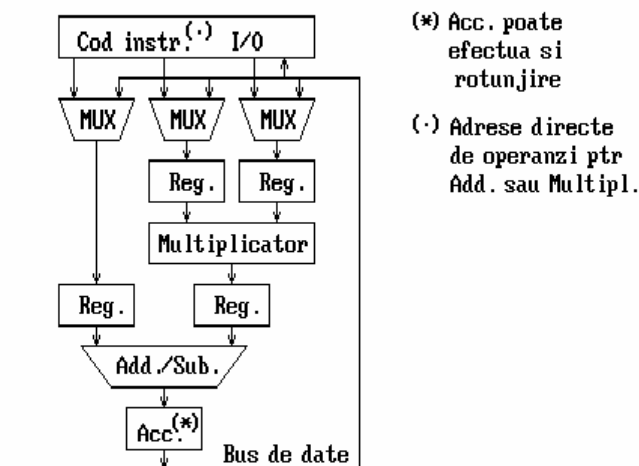


Figura 3 a)

b) structura paralelă (figura 3 b)

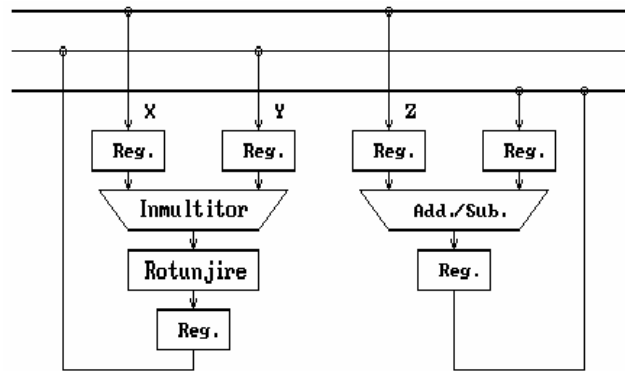


Figura 3 b)

Avantajul acestei structuri este posibilitatea efectuării unor calcule, de tip $x.y+z$, cu viteză foarte mare datorită faptului ca sumatorul și multiplicatorul pot lucra paralel.

Dezavantajele structurii sînt: numărul de busuri interne mare; pentru reducerea numărului de busuri se alege varianta transferurilor pe busuri comune sumatorului și multiplicatorului: paralelismul nu mai este total.

Unitatea de control a unui procesor de semnal poate fi realizată microprogramat utilizînd tehnica pipe-line pentru a mari viteza de lucru. În figura 3.c este ilustrată realizarea unei unități de control microprogramate fără pipe-line. Modificările introduse prin tehnica pipe line sînt ilustrate în figura 3d.

Notățiile utilizate sînt:

t_{secv} - timp de calcul al adresei pentru memoria demicroprograme

t_{acc} - timp de acces la memoria de microprograme

t_{PR}, t_{SR} - timp de încărcare a registrelor R1, R2

t_{stat} - timp de stabilire a stării unității aritmetice (în avans)

t_c - timp de calcul efectiv (operații interne în unitatea aritmetică)

În timp ce unitatea aritmetică execută instrucțiunea (n) secvențorul calculează deja adresa pentru instrucțiunea (n+1) folosind ca informație instrucțiunea (n) și starea unității aritmetice de la instrucțiunea (n-1).

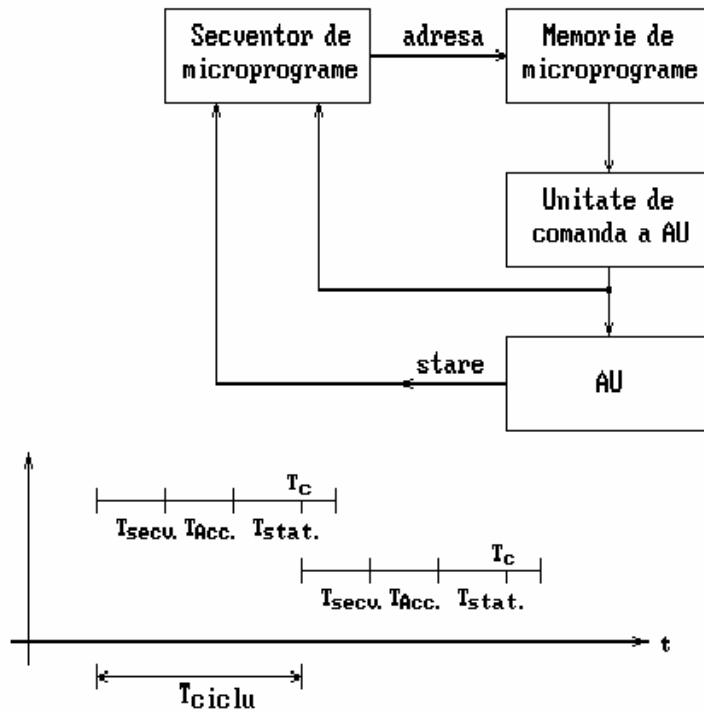


Figura 3 c)

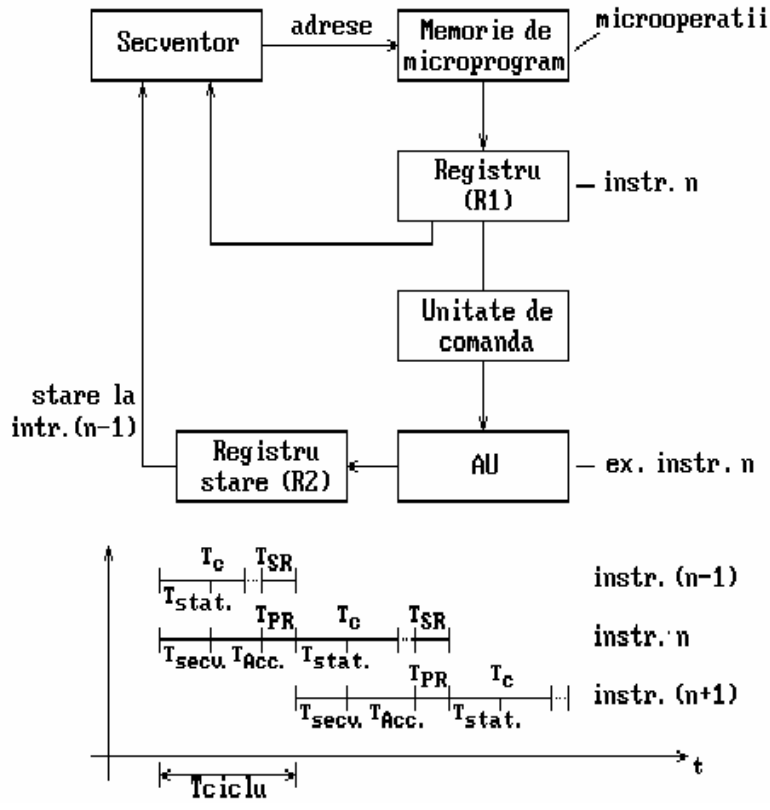


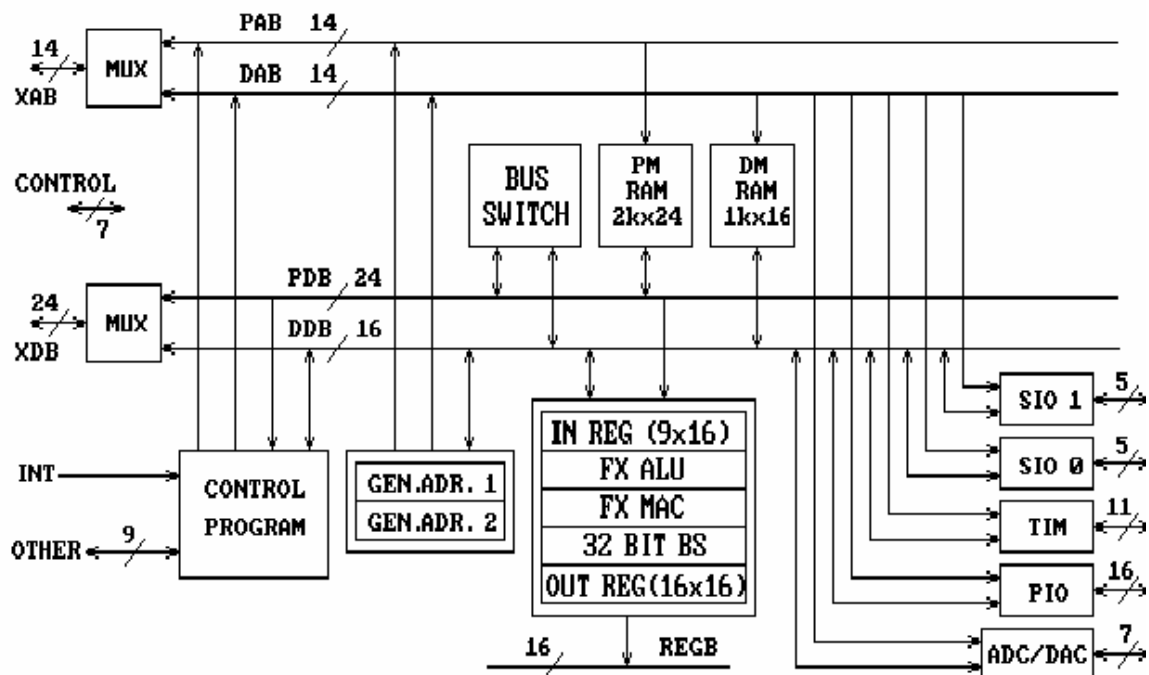
Figura 3 d)

Exemple de arhitecturi de procesoare de semnal

Această secțiune prezintă câteva exemple de familii de procesoare de semnal.

Procesoare de virgulă fixă

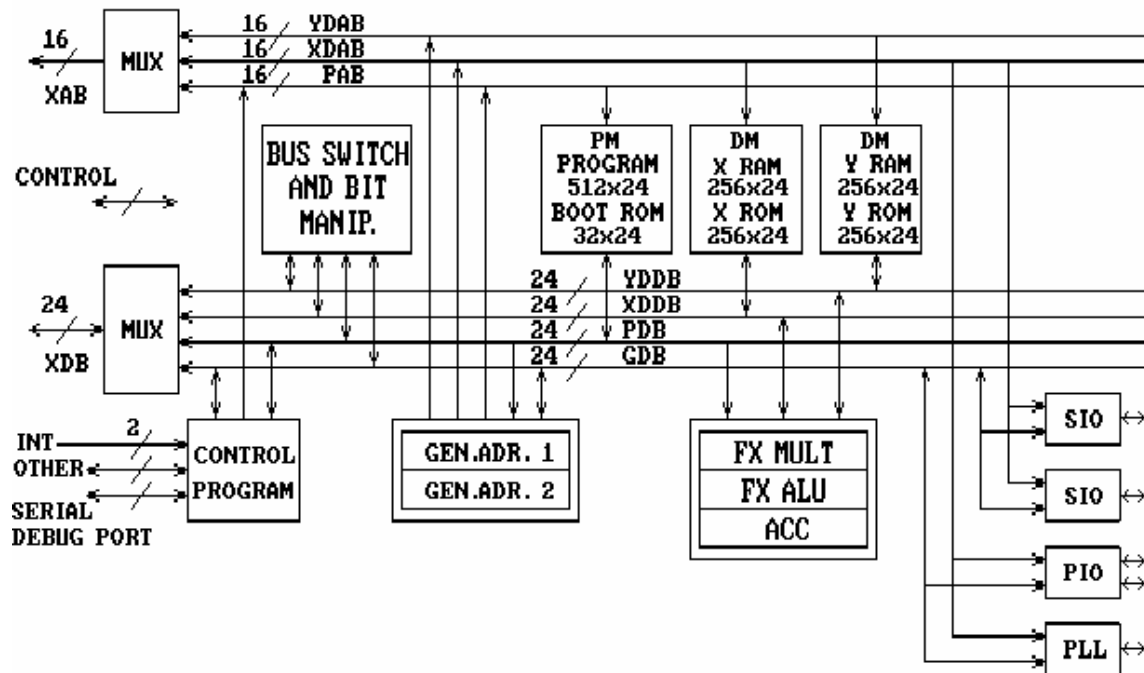
ADSP 2100 (Analog Devices)



Caracteristici

- ciclul masină 13 – 33 ns
- memorie on-chip (maxim 16 k x24 pentru program și 16 k x 16 pentru date, adresabilitate directă, posibilitate de extindere prin overlay)
- porturi seriale ,timer
- stive hardware
- 6-12 niveluri de întrerupere
- aritmetică în virgulă fixă (C2)
- sumator pe 16 biti
- înmultitor 16 x 16 biți cu acumulare pe 40 de biți
- adresare directă, indirectă, circulară, bit-reverse

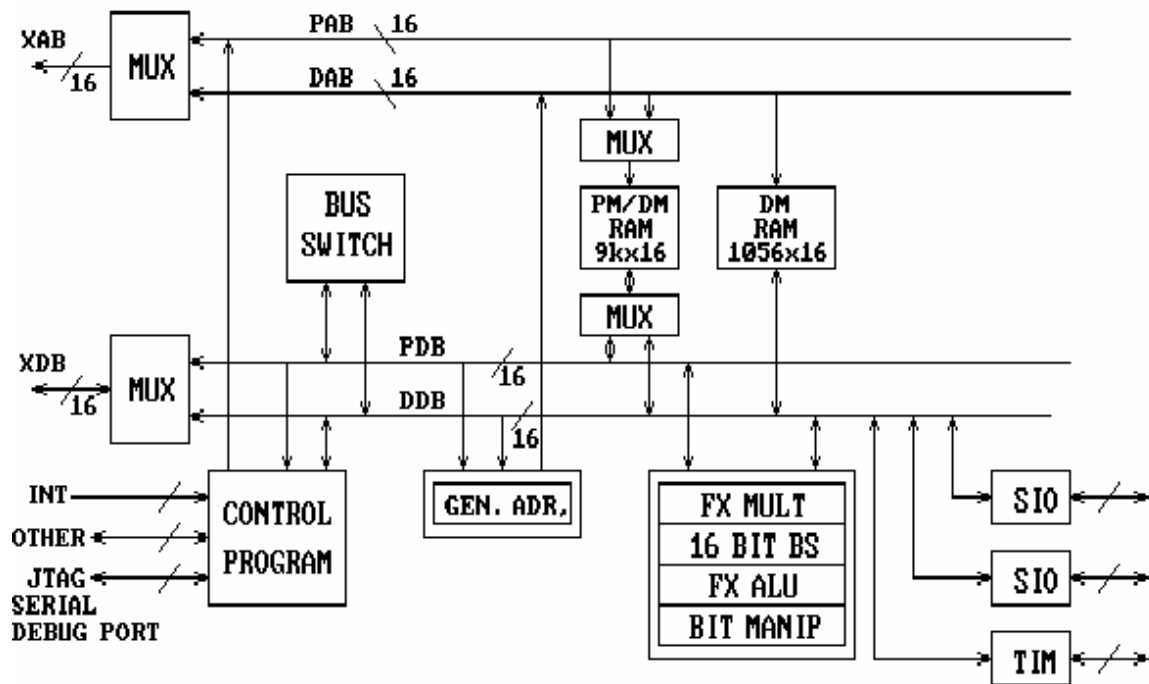
DSP 56001/56002 Motorola



Caracteristici

- ciclul masină 30 ns
- memorie on chip
- port serial și port paralel (pe 8 biti)
- interfață paralelă cu un microcalculator (PC)
- stivă hardware
- aritmetică în virgulă fixă
- operații aritmetice pe 56 biti , logice pe 24 biți
- 32 niveluri de întrerupere

TMS 320C2X / 320C5X (Texas Instruments)



Caracteristici

- ciclu masină 20 ns
- memorie on chip
- porturi seriale
- stivă hardware
- aritmetică în virgula fixă
- ALU pe 32 biți
- înmultitor 16 x 16 biti cu acumulare pe 32 biti

Comentarii despre arhitecturi

DSP 56000/56001

Secțiunea aritmetică

- Unitate aritmetică rapidă pe 24 (în)/48 (out)/56 (ext) biți
- Execută operațiile ALU într-un singur ciclu

Părțile componente sînt:

- 4 registre de 24 biți
- unitate MAC (înmultitor paralel) - într-un singur ciclu
- 2 acumuloare de 48 biți
- 2 registre de extensie de 8 biți
- 1 shifter (1 poziție)
- 2 circuite de shiftare/limitare (pentru saturarea aritmetică); de exemplu:

operandul sursă este 01.100 (+1.5), destinația are 4 biți deci rezultă suma 1.100 (bitii LSB) adică (-1.5) rezultat eronat! Pe 4 biți a apărut o depășire care prin limitare dă 0.111 adică +0.875 - eroarea este mai mică.

- acumulator separat pentru MAC

Registrele servesc pentru introducerea operandilor în ALU sau MAC. Se pot citi /scrie direct de pe bus-ul de date - se elimină astfel problemele anunțate anterior (înmulțiri multiple și adunări multiple - vezi schema b) de unitate aritmetică). După înmulțire se poate face rotunjire.

Concluzii:

- Unitatea aritmetică este destinată efectuării calculelor aritmetico-logice, inmulțiri și sume de produse cu operanzi din 2 zone de date diferite (X sau Y).
- Nu există pipe-line;
- Există un set de indicatori condiționali.

Unitatea de adresare a datelor

- Permite adresare liniară, modulo,cu inversarea bitilor, *bit-reverse*.
- Poate adresa date din memoria de program.
- Generează adresa pentru 2 operanzi într-un singur ciclu
- Adresa este de 16 biți
- Calculul adresei se face astfel: $adresa = (R+N) \bmod M$

Secvențorul de program

Funcțiuni:

- generarea adresei de instrucțiune (prefetch)
- decodarea instrucțiunii
- bucle DO (implementate HW, fara "overhead")
- controlul programului
- controlul întreruperilor (32 întreruperi pe 4 nivele de prioritate: '*fast*' și '*long* '; pentru intreruperile '*fast*' se executa maxim 2 intrucțiuni de la adresa fixă apoi se revine în programul întrerupt în mod automat fără instructiunea de revenire din intrerupere -RETI)

Exista stivă HW (extensibilă SW). Instrucțiunile sînt de tip "pipe-line" cu 3 nivele (fetch,decodificare,executie); acest lucru permite execuția într-un singur ciclu a majorității instrucțiunilor.

TMS 320C25

Secțiunea aritmetică

- un singur acumulator pentru operatii aritmetice/logice și MAC;

Nu se pot încarcă simultan 2 operanzi simultan în ALU ci:

- un operand din acumulator;
- un operand - prin bus-ul de date;
- ieșirea multiplicatorului;

O adunare presupune ștergerea acumulatorului, încărcarea cu o dată din memorie și apoi adunarea cu altă dată din memorie (2 cicluri pentru o singură adunare).

ex.: ZALH <data mem adr>
 ADDH <data mem adr>

- nu toate operațiile ALU se desfășoară într-un singur ciclu;
- nu toate operațiile ALU se desfășoară cu prefixul REP (de repetare);
- nu are înmulțitor/acumulator dedicat pentru MAC;
- înmulțirea nu se desfășoară într-un singur ciclu:
 - mai întâi se încarcă registrul TR al MPY;
 - apoi se efectuează înmulțirea cu un operand din memoria de date sau de program.
- înmulțirea cu acumulare se efectuează într-un singur ciclu .

Shifter-ul efectuează deplasări cu:

- 0-16 biți la stânga/dreapta (înaintea efectuării operațiilor aritmetice);
- 6 biți la dreapta (pentru protecție overflow) sau 1-4 biți la stânga după efectuarea unei înmulțiri (după înmulțitor).
- 0-7 biți la stânga după acumulator (înainte de stocarea în memorie, după efectuarea operațiilor aritmetice).

Adresarea datelor

- folosește registre auxiliare pentru stocarea adreselor (adresare indirectă);
- are un singur registru modificador (+/-1 numai);
- nu pot fi adresate 2 date (operanzi) simultan;
- nu are adresare "modulo" (deci nu are "buffere circulare");
- adresare directă: pagini de 128 cuvinte de 16 biți (un registru de pagina memorează pagina curentă de date);
- numărul registrului auxiliar care conține adresa este stocat în ARP (*auxiliar register pointer*) citit prin ARB (*auxiliar register buffer*);
- registrele auxiliare pot fi utilizate pentru stocarea unor date;

- există modul *bit-reverse*.

Secventorul de program

- instrucțiuni cu "pipe-line" (instrucțiunile sînt "citite" cu adresa generată de "prefetch counter" și sînt depuse în "coada de instrucțiuni"). În funcție de execuția instrucțiunii curente mai multe instrucțiuni pot fi depuse în coada de instrucțiuni (controlul programului este dificil de înțeles; apar 3 cicluri suplimentare în cazul salturilor, necesare pentru stergerea cozii de instrucțiuni).
- stiva pe 8 niveluri;
- nu se salvează complet starea microprocesorului la apariția unei întreruperi;
- nu se pot implementa hardware bucle de tip DO UNTIL; aceste bucle se pot implementa numai prin software (apare un ciclu suplimentar pentru tratarea condiției de terminare a buclei).

ADSP 2101

Secțiunea aritmetică:

- încărcare a 2 operanți (pe busul de date al memoriei de program PMD și pe busul de date al memoriei de date DMD);
- instrucțiuni într-un singur ciclu;
- instrucțiuni condiționale;
- structură foarte flexibilă;
- MAC cu rotunjire aritmetică;
- deplasări cu 0-32 poziții stînga / dreapta într-un singur ciclu;

Secțiunea de adresare a datelor

- adresare a 2 operanți simultan (din memoria de date DM și din memoria de program PM) într-un singur ciclu;
- adresare - directă;
- indirectă;

- adresare liniară, modulo, *bit-reverse*;
- adrese pe 14 biți - 16k cuvinte de memorie.

Secvențorul de program

Selectează adresa pentru PM

- din registrul *program counter* PC
 - adresa liniară;
 - adresa de salt (jump, call);
 - stiva PC (reintoarcere din subrutine și intreruperi);
 - vector de intrerupere.

- controler de întreruperi (exista o stivă hardware pentru salvarea completă a stării procesorului);

- bucle DO UNTIL implementate hardware (sînt permise pînă la 4 bucle înlanțuite, există o stivă hardware pentru contorul de iterații și stivă hardware pentru salvarea condiției de terminare a buclei - 4 niveluri fiecare);

- nu apar cicluri suplimentare (nici la bucle nici la salturile condiționate);

- este ușor de calculat timpul de execuție al programului.

Comparația între cele trei arhitecturi este centralizată în tabelele următoare:

| Posibilități aritmetice | ADSP 2101 | TMS 320C50 | DSP 56000/1 |
|---|--------------------------|--------------------------|-------------------------------|
| Toate operațiile ALU într-un singur ciclu | DA | NU | DA |
| Multiplicarea într-un singur ciclu | DA | NU | DA |
| MAC într-un singur ciclu | DA | DA (în mod repetitiv) | DA |
| Shiftare într-un singur ciclu | 0-32 biți stînga/dreapta | 0-16 biți stînga/dreapta | 0-56 sau 0-24b stînga/dreapta |
| Protecție la overflow a acumulatorului | 8 biți | 1 bit | 8 biți |
| Multiplicări mod cu și fara semn și mixt | DA | nu are mod mixt | nu are mod mixt |

| | | | |
|---|------------------|-------------------|------------------------------------|
| Normalizări într-un singur ciclu | DA | NU | DA |
| Possibilități de adresare a datelor | ADSP2101 | TMS320C50 | DSP56000/1 |
| Încarcare a 2 operanzi într-un singur ciclu din memoria on-chip | DA | NU | DA |
| Modificarea 2 adrese cu 2 valori diferite la fiecare ciclu | DA | NU | DA |
| Mod "bit-reverse" | DA | DA | DA |
| Buffere circulare cu pas variabil (+/-) | DA | numai cu pas 1 | DA |
| Possibilitati de control al programului DSP56000/1 | ADSP 2101 | TMS 320C50 | |
| Adâncimea stivei PC | 16 | 8 | 32x16 (globala PC,bucle, etc.) |
| Bucle înlantuite | 4 | NU | DA în limita stivei globale |
| Instructiuni aritmetice condiționale | DA | NU | NU(numai de transfer și salt) |
| Salturi și bucle fără cicluri suplimentare | DA | NU | NU (cu excepția buclelor hardware) |
| Salvare automată a stării mașinii la întrerupere | DA | NU | DA |
| Vitezele de lucru ale celor 3 procesoare DSP | (*) DSP 56000 | (*) TMS 320C25 | ADSP 2101 |
| Inmultire matrici 3x3 | 1659ns | 1800ns | 1600ns |
| FFT în 1024 puncte | 3,231ms | 9,08ms | 4,23ms |
| Filtru IIR cu 8 poli | 2800ns | 3520ns | 3280ns |

| | | | |
|-------------------|---|---|-----------|
| DTMF | - | - | 12 canale |
| LPC ord 10 | - | - | 0,36ms |
| ADPCM full-duplex | - | - | 0,068ms |

Procesoare de semnal de virgulă mobilă

Formatul datelor utilizate de un procesor de semnal determină capacitatea de a prelucra semnale cu o anumită precizie, gamă dinamică și raport semnal zgomot. Totuși trebuie luate în considerare și aspectele referitoare la modul de manevrare a datelor și la costul structurii de prelucrare.

Precizia de prelucrare este corelată cu precizia convertorului A/D și cu cerințele impuse de algoritmul DSP .

Algoritmii de compresie și decompresie utilizați în dispozitivele de eșantionare operează pentru semnale de bandă cunoscută și au avantajul că permit utilizarea unor structuri de prelucrare de precizie mai mică , deci mai ieftine și mai ușor de implementat. Există totuși situații în care o gamă dinamică mare este necesară (de exemplu filtrarea adaptivă și prelucrarea imaginilor). De asemenea o gamă dinamică mare permite obținerea unui raport semnal / zgomot mare (de exemplu pentru recunoașterea vorbirii).

Familia de procesoare de semnal ADSP 21010 / 21020

Arhitectura procesorului de semnal ADSP 21020 este prezentată în figura 4

Se observă faptul că arhitectura procesorului ADSP 21020 nu diferă , ca principiu de organizare, de arhitectura procesorului ADSP 2100 (de virgulă fixă).

Există cele 3 unități de prelucrare (MAC, ALU, SHIFTER) conectate în paralel. Busul de date este de 32 biți (extins la 40 biți), iar busul de instrucțiuni este de 48 biți.

Unitatea MAC poate opera atât în virgulă mobilă cât și în virgulă fixă, în ceea ce privește înmulțitorul. Registrul de acumulare al unității MAC este de virgulă fixa. Utilizând ca registru acumulator orice registru de virgulă mobilă se pot efectua operații de tip suma de produse (cu ajutorul unor instrucțiuni multifuncție) .

Se mențin ca principiu toate caracteristicile familiei de procesoare ADSP 2100 (implementate diferit) și se adaugă noi facilități pentru realizarea unor structuri hardware și a unor programe mai simple și mai flexibile.

Tendința de viitor este realizarea unor procesoare de semnal cu un grad de integrare foarte ridicat - ca de exemplu procesoarele de semnal SHARC (*Super Harvard Architecture Computer*).

Procesorul SHARC - ADSP 21060

Acest procesor preia arhitectura procesorului ADSP 21020 și înglobează suplimentar următoarele:

- memorie internă SRAM (configurabilă ca memorie de program sau ca memorie de date)

- un procesor de intrare / ieșire

- interfață cu un calculator gazdă

Procesorul de intrare / ieșire are structura din figura 5.

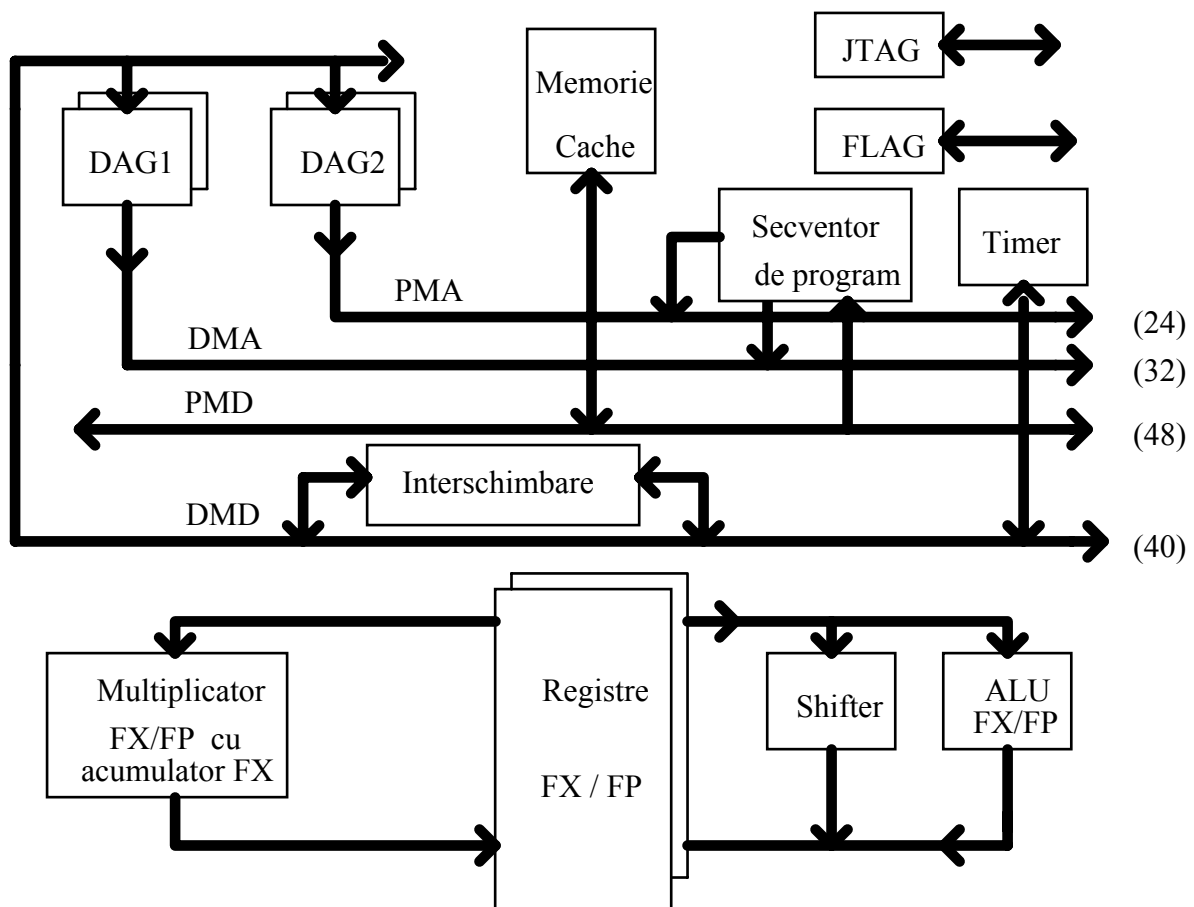


Figura 4. Arhitectura procesorului de semnal ADSP21020

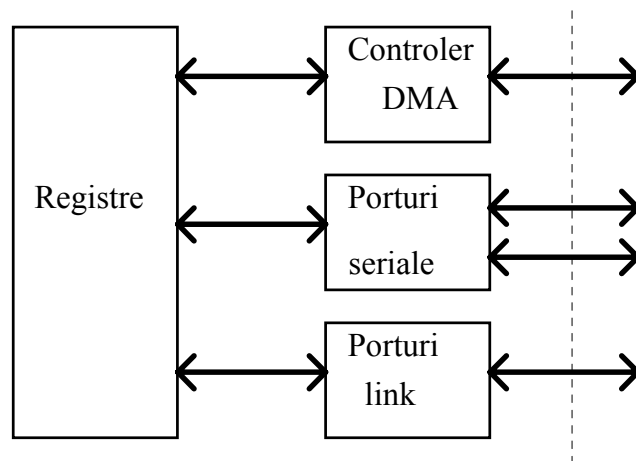


Figura 5. Procesorul I/O din structura circuitului SHARC - ADSP21060

Interfața cu calculatorul gazda permite conectarea procesorului ADSP 21060 cu procesoare de uz general standard pe 16 sau 32 biti cu hardware adițional minim.

Controlerul DMA poate fi utilizat pentru transferuri directe cu memoria (internă și externă sau memoria calculatorului gazdă).

Procesorul ADSP 21060 este optimizat pentru interconectarea în sisteme multiprocesor cu procesoare de semnal.

De asemenea porturile "link" permit comunicația între procesoare în sisteme multiprocesor.

Considerente pentru proiectarea procesorului de semnal

Prelucrarea digitală a semnalelor presupune utilizarea unor procesoare de semnal cu performante ridicate.

Performantele nu pot fi masurate numai prin viteza de multiplicare/acumulare (MIPS). Deoarece o instrucțiune pentru un procesor de semnal (DSP) nu este echivalentă cu o instrucțiune pentru un alt procesor de semnal, rata MIPS poate să nu fie concludentă și poate conduce la erori de apreciere.

Arhitectura unui DSP și performanțele pentru fiecare bloc funcțional din aceasta arhitectură (cum ar fi unitatea aritmetică, unitatea de adresare și secvențorul de program) pot fi mult mai importante în stabilirea caracteristicilor unui procesor de semnal. Ceea ce distinge un procesor de semnal de un procesor de uz general este reprezentat de următoarele:

1. Aritmetica rapidă și flexibilă

Un procesor de semnal trebuie să permită calcule într-un singur ciclu (multiplicări, împărțiri și acumulări, deplasări, operații aritmetice și logice). Unitatea aritmetică trebuie să permită implementarea unui algoritm de prelucrare a semnalelor fără ca acesta să fie reformulat (modificat) pentru a putea fi implementat.

2. Dinamica extinsă pentru multiplicari/acumulari

Dinamica extinsă este fundamentală pentru algoritmi de prelucrare a semnalelor. Aceasta asigură protecția la depășirile aritmetice care apar în decursul sumelor parțiale (de exemplu efectuarea calculelor pe 40 biți cu 8 biți de extensie a bitului de semn).

3. Încărcarea a 2 operanzi într-un singur ciclu (din memoria internă sau externă)

În calculele cu sumă de produse (convoluții numerice) trebuie încărcati 2 operanzi pentru a efectua calculele. Capacitatea de adresare flexibilă pentru 2 date din memorie este importantă.

4. Buffere circulare hardware (în memoria internă sau externă)

O clasă largă de algoritmi de prelucrare a semnalelor necesită buffere de memorie circulare; dacă există posibilitatea hardware de a utiliza astfel de buffere se reduce timpul de execuție (de prelucrare) și se simplifică implementarea algoritmului.

5. Bucle hardware și salturi condiționate fără cicluri suplimentare ("zero overhead")

Algoritmii de prelucrare sînt în mod natural repetitivi și pot fi exprimați simplu prin bucle de program. Secvențorul de program trebuie să permită implementarea buclelor și a salturilor condiționate fără cicluri suplimentari datorati testării condiției de terminare a buclei sau testării condiției de salt; aceasta conduce la performanțe mai bune din punct de vedere al timpului de prelucrare.

Proiectarea unui procesor de semnal trebuie să ia în considerare tipul algoritmilor DSP ce urmează a fi implementați de procesor.

Procesorul de semnal nu trebuie să aibă un bloc de control centralizat deoarece majoritatea grafurilor DFG pot fi gândite cu prelucrări paralele (astfel încît fiecare unitate de

prelucrare sa fie utilizată în paralel cu alte unități - acest lucru este posibil dacă graful aplicației este realizat cu un anumit grad de paralelism). Ideea principală este aceea ca fiecare unitate de prelucrare să posede un bloc de control propriu (pentru operația specifică); blocul de control centralizat se distribuie către fiecare unitate de prelucrare.

În funcție de complexitatea aplicației se poate opta pentru una din formele de procesor de semnal prezentate anterior (pipe-line , vectoriale , matriciale , paralele). Apar însă probleme în ceea ce privește interconectarea elementelor funcționale în cadrul acestor procesoare (sincronizare, comunicație).

Pentru aplicațiile specifice în telecomunicații (cu excepția aplicațiilor multimedia) sistemele "clasice" cu un procesor integrat specializat performant conduc la rezultate bune.

Există două categorii de arhitecturi de procesoare de semnal : organizate în timp (*time-line*) și organizate după date (*data-driven*).

Pentru arhitecturile organizate în timp execuția fiecărei operații se realizează într-un punct specificat al secvenței de prelucrare. Deoarece prelucrarea este modificată în timp în sensul ștergerii unor operații sau inserării unor operații noi în secvența de prelucrare rezultă algoritmi de planificare și distribuire a operațiilor destul de complexi , ceea ce necesită un timp de execuție mare; aplicația poate să nu mai fie de timp real.

Arhitecturile organizate după date preiau datele de intrare într-o coadă de intrare ; apariția datelor în cozile de intrare determină planificarea și distribuirea operațiilor ; rezultatele sunt plasate într-o coadă de iesire. Dacă dimensionarea cozilor este corectă caracterul de aplicație de timp real se menține chiar și pentru secvențe de operații complexe.

În procesorul de semnal poate apărea o întârziere între momentul în care unitatea de prelucrare este pregătită de operare și momentul când începe execuția operației. Această întârziere apare în mod aleator datorită faptului că blocul de control este dinamic.