

9. CIRCUITE DSP

9.1. Introducere

Reamintindu-ne de jocul de puzzle pe care îl face inginerul care proiectează o aplicație am considerat oportun să prezentăm o piesă a acestui joc, întâlnită în ultimii ani în tot mai multe aplicații, în variate domenii - este vorba despre procesorul digital de semnale cunoscut de avizați ca circuit DSP (Digital Signal Processor). DSP-urile fac parte din categoria circuitelor specializate și sunt microprocesoare proiectate pentru a îndeplini optim sarcini de procesare digitală de semnal.

Dezvoltarea acestui tip de componentă ca varietate și volum de producție a fost dictată de sectorul comercial mai degrabă decât de necesitățile militare sau guvernamentale. DSP-urile au ajuns la utilizatorul obișnuit prin telefoane mobile, aparate CD-palyer, poștă electronică vocală, și alte aplicații. Amintim câteva domenii și categorii de aplicații în care sunt utilizate circuite DSP:

- cercetare științifică (achiziții de date, analiză spectrală, simulare și modelare);
- medicină (stocare/redare de imagini, tomografii computerizate);
- domeniul militar (radar, sonar, comunicații secrete);
- domeniul aerospațial (îmbunătățirea calității imaginilor, compresie de date, analiza informației de la sonde);
- telecomunicații (telefonie, televiziune, radio);
- domeniul comercial (compresie de imagine și sunet, efecte speciale în filme, videoconferințe);
- seismologie;
- industrie (monitorizarea și controlul proceselor, testarea nedistructivă, proiectare asistată).

Plasarea eficientă a unui circuit DSP în jocul de puzzle presupune deținerea unui bagaj de cunoștințe în relativ multe domenii (analiză numerică, probabilități și statistică, procesarea analogică a semnalelor, electronică digitală, electronică analogică, teoria comunicațiilor).

9.2. Arhitectura circuitelor DSP

Majoritatea calculatoarelor zilelor noastre au arhitectură von Neumann (cu bus unic pentru date și instrucțiuni). Această arhitectură nu satisface necesitățile de viteză de procesare ale aplicațiilor de procesare de semnal. Arhitectura Harvard (Fig. 9.1) este mult mai potrivită pentru a fi implementată în DSP-uri.

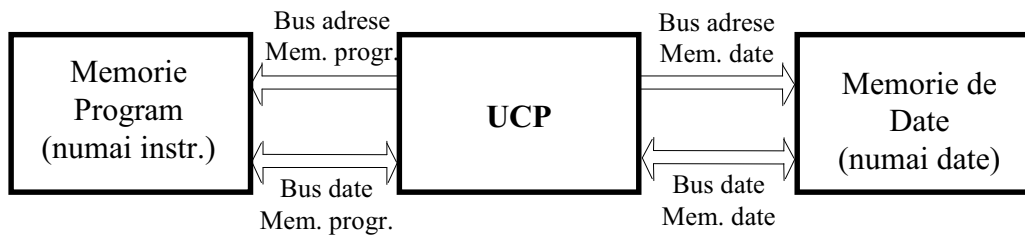


Fig. 9.1 Arhitectura Harvard (memorie program și memorie de date separate)

Inconvenientul acestei arhitecturi, din punct de vedere al vitezei de procesare, este faptul că magistrala memoriei de date este mult mai încărcată decât magistrala memoriei de program.

La înmulțirea a două numere, prin magistrala de memorie de date sunt vehiculate două cuvinte (cele două numere), iar prin magistrala memoriei program, un singur cuvânt (codul instrucțiunii de înmulțire).

Această situație poate fi îmbunătățită dacă o parte din informația din memoria de date este relocalată în memoria program.

Pentru un filtru, de exemplu, se poate păstra informația ce reprezintă eșantioanele în memoria de date și coeficienții în memoria program.

Informația relocalată este referită ca date secundare. Aparent această modificare nu îmbunătățește cu nimic situația; considerând însă specificul operațiilor DSP, cu încă o modificare în arhitectură se poate obține reducerea timpului de procesare; în UCP se include un bloc de memorie cache pentru instrucțiuni. Memoria cache pentru instrucțiuni este binevenită deoarece majoritatea algoritmilor DSP conțin bucle ce se repetă de un număr de ori. O astfel de arhitectură este numită Super-Harvard și este reprezentată în Fig. 9.2.

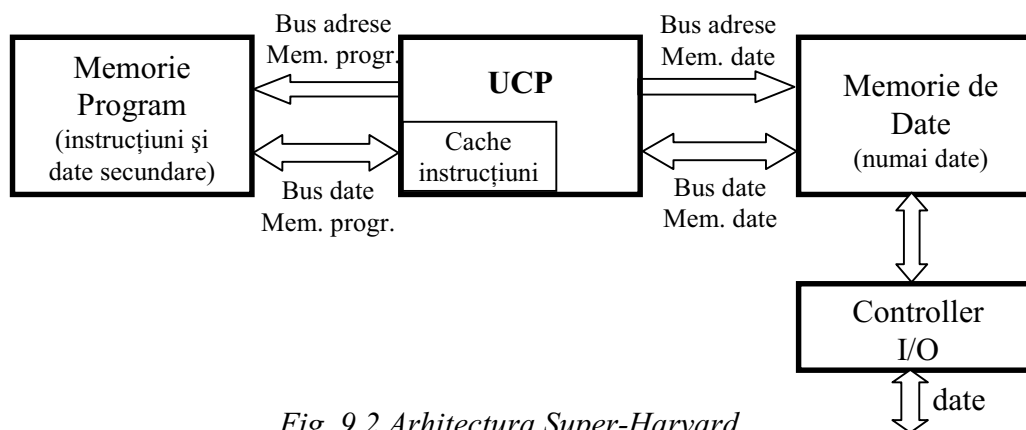


Fig. 9.2 Arhitectura Super-Harvard

Arhitectura Super-Harvard evidențiază și un controller intrare/ieșire dedicat. Controllerul conține porturi rapide de comunicație serială și paralelă. Controllerul intrare/ieșire poate să includă și un modul DMA care să conducă transferul de date între

memorie și periferice fără a consuma timp UCP. Schimbul rapid de informație cu exteriorul este la fel de important la un DSP ca și viteza de procesare a unității centrale. Problema aplicației trebuie privită în felul următor: datele sunt introduse, procesate, după care rezultatul este livrat spre exterior înainte ca următorul eșantion de procesat să fie disponibil la intrare.

O arhitectură tipică pentru un DSP, privită mai în detaliu, este reprezentată în Fig. 9.3.

Corespunzător fiecărui bloc de memorie, există în UCP câte un generator de adrese; în procesoarele obișnuite funcție este realizată de secvențiatorul de program.

Procesarea matematică este împărțită în trei secțiuni: multiplicator, UAL (unitate aritmetică și logică) și modul de shiftare. Multiplicatorul și UAL pot fi accesate în paralel. Conceptul de procesare paralelă este implementat la toate nivelele într-un DSP.

Există multe particularități și artificii specifice fiecărui tip de procesor, implementate în scopul creșterii eficienței procesării semnalelor. Acestea vor fi exemplificate pe cazuri particulare.

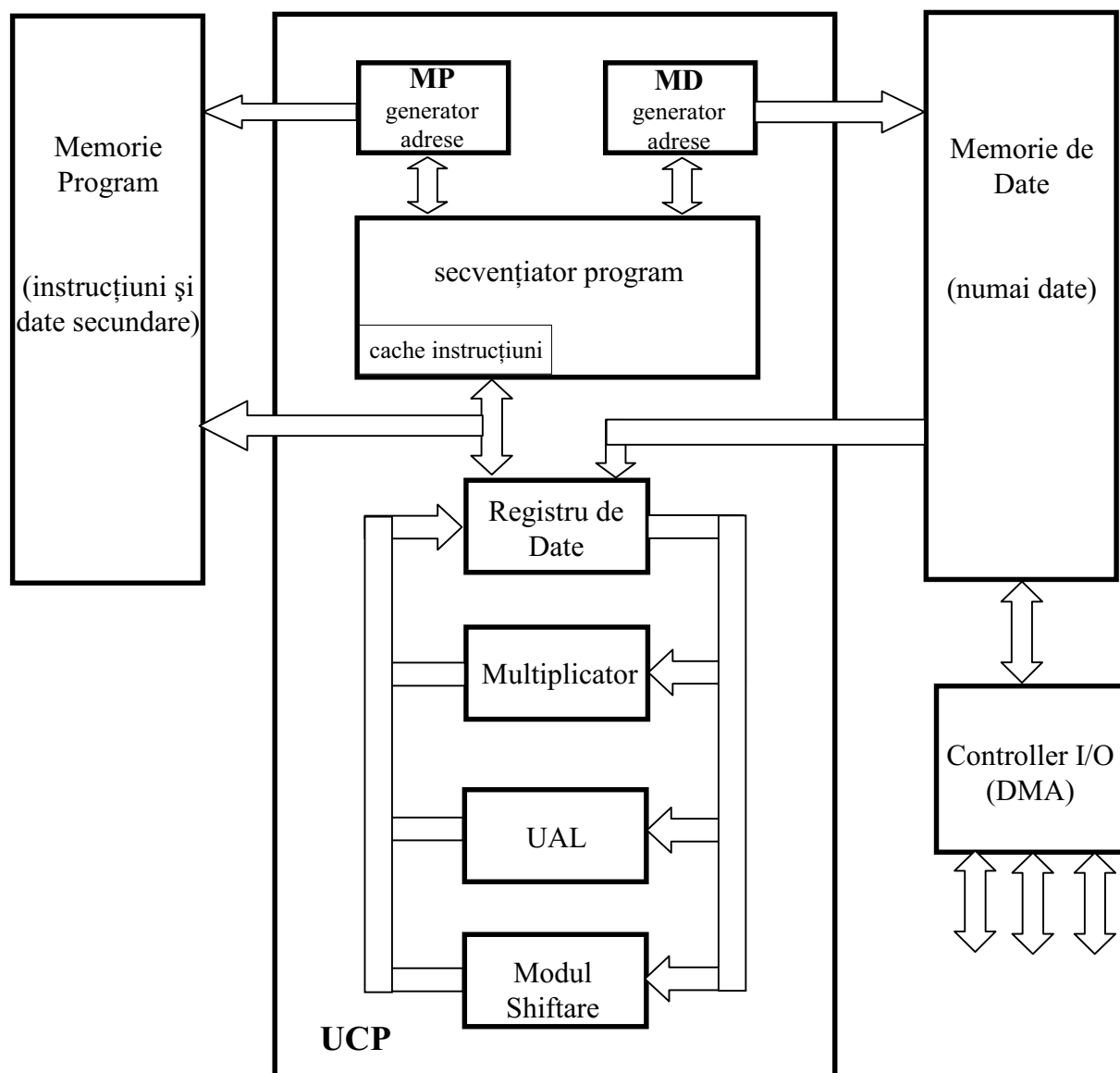


Fig.9.3 Arhitectură tipică DSP

9.3. Reprezentarea informației

În funcție de modul în care sunt reprezentate numerele, procesoarele digitale sunt de două categorii:

- procesoare în virgulă fixă;
- procesoare în virgulă mobilă.

În cazul procesoarelor în virgulă fixă, în mod obișnuit, fiecare număr este reprezentat folosind minimum 16 biți. Spre exemplu, Motorola fabrică procesoare în care se utilizează reprezentarea în virgulă fixă pe 24 de biți.

În reprezentare pe n biți sunt posibile patru moduri de reprezentare a celor 2^n combinații:

- întreg fără semn;
- întreg cu semn;
- număr fracționar fără semn;
- număr fracționar cu semn.

Pentru reprezentarea numerelor cu semn se folosește codul complement față de doi. Pentru $n=16$, pot fi reprezentate numere întregi fără semn de la 0 la 65 535, numere întregi cu semn în intervalul -32 768 la +32 767, 65 536 de numere fracționare fără semn distribuite uniform în intervalul cuprins între 0 și 1 și 65 536 de numere fracționare cu semn distribuite uniform în intervalul cuprins între -1 și +1.

Pentru reprezentarea în virgulă mobilă, în DSP-uri se utilizează minimum 32 de biți, astfel sunt posibile mult mai multe combinații de biți (cel puțin $2^{32}=4\ 294\ 967\ 296$). În reprezentarea în virgulă mobilă codurile nu sunt uniform distribuite în intervalul reprezentat. Cel mai utilizat format de reprezentare în virgulă mobilă este cel reglementat de standardul ANSI/IEEE 754-1985; în acest format cel mai mare număr reprezentabil este $\pm 3.4 \cdot 10^{38}$, iar cel mai mic număr reprezentabil este $\pm 1.2 \cdot 10^{-38}$. Distanța între două numere consecutive este de aproximativ 10 milioane de ori mai mică decât numărul reprezentat - în acest fel, între numere mari consecutive distanța este mai mare, iar distanța între numere mici consecutive este mică.

Toate procesoarele în virgulă mobilă pot opera și cu numere reprezentate în virgulă fixă, mod de lucru necesar atunci când se implementează numărătoare, bucle sau când se operează cu informație de la convertoare analog-digitale sau informație destinată convertoarelor digital-analogice.

Într-un procesor de uz general, aritmetica în virgulă fixă este mai rapidă decât aritmetica în virgulă mobilă; în DSP-uri, prin optimizările arhitecturale specifice, viteza de procesare este aproximativ egală pentru cele două moduri de reprezentare a numerelor. Arhitectura internă a unui DSP în virgulă mobilă este mult mai complicată decât a unui DSP în virgulă fixă. Toate registrele și toate magistralele trebuie să fie de 32 de biți în loc de 16, unitatea de multiplicare și UAL trebuie să execute rapid operații în virgulă mobilă și setul de instrucțiuni trebuie să cuprindă instrucțiuni care permit manipularea atât a numerelor în virgulă mobilă cât și a numerelor în virgulă fixă. Simplitatea procesoarelor în virgulă fixă este reflectată în costul acestora care, oricât de rapidă ar fi evoluția prețurilor, va rămâne întotdeauna mai scăzut decât al DSP-urilor în virgulă mobilă; dinamica prețurilor se reflectă în

raportul celor două costuri. Avantajele utilizării fiecărui mod de reprezentare a numerelor sunt sintetizate în continuare:

- reprezentarea numerelor în virgulă mobilă conferă DSP-urilor următoarele avantaje:
 - ⇒ precizie bună (posibilitatea de a reprezenta numere fracționare foarte mici);
 - ⇒ dinamică mare (intervalul dintre cel mai mare întreg negativ și cel mai mare întreg pozitiv este foarte mare);
 - ⇒ timp scurt de dezvoltare a aplicațiilor.
- reprezentarea numerelor în virgulă fixă conferă DSP-urilor avantajul unui cost scăzut.

DSP-urile operează cu semnale; reprezentarea discretă a semnalului presupune cuantizarea acestuia și implicit survine zgomotul de cuantizare. Zgomotul de cuantizare este în directă legătură cu formatul ales pentru reprezentarea numerelor, depinde de distanța dintre două valori consecutive în codul de reprezentare. Raportul semnal/zgomot, când se utilizează reprezentarea în virgulă mobilă este de aproximativ 30 milioane /1, în timp ce acesta este de 10 mii/1 dacă se utilizează reprezentarea în virgulă fixă.

Presupunem că se implementează un FIR în virgulă fixă; pentru aceasta se operează în buclă, fiecare coeficient este înmulțit cu eșantionul corespunzător din semnalul de intrare și rezultatele se adună în acumulator. Într-un procesor de uz general, acumulatorul este un registru de 16 biți unde numerele sunt reprezentate în virgulă fixă. Pentru a evita depășirea capacității de reprezentare, valorile ce se adaugă în acumulator vor fi rotunjite și se adaugă astfel zgomot de cuantizare la fiecare pas.

Pentru un filtru cu 500 de coeficienți, zgomotul fiecărui eșantion de ieșire va fi de 500 de ori mai mare decât zgomotul avut la intrare.

Această problemă este tratată într-un DSP prin utilizarea unui acumulator pentru precizie extinsă. Acesta este un registru special a cărui dimensiune este de 2-3 ori mai mare decât dimensiunea unui cuvânt (32-40 biți pentru procesoarele pe 16 biți sau până la 80 de biți pentru procesoarele pe 32 de biți). Strategia utilizării extensiei pentru acumulator dă rezultate bune în DSP-urile în virgulă fixă. DSP-urile cu reprezentare în virgulă mobilă, introduc un zgomot de cuantizare atât de mic încât nu este necesare utilizarea extensiei de acumulator.

Alegerea unui DSP în virgulă fixă sau mobilă se face în funcție de particularitățile aplicației considerate. Spre exemplu, semnalele din televiziune, în general semnalele video, sunt convertite analog/digital și apoi digital/analog în mod uzual pe un număr redus de biți (8 biți) - în astfel de aplicații este suficientă precizia unui procesor în virgulă fixă. Aplicațiile audio profesionale folosesc semnale procesate cu convertoare analog/digital și digital/analogice de 20-24 biți, ceea ce presupune necesitatea de a utiliza procesoare în virgulă mobilă pentru a acoperi dinamica semnalului.

Un alt considerent în alegerea tipului de DSP pentru o aplicație este complexitatea algoritmului ce va fi implementat. Pentru algoritmi simpli, cum ar fi filtrele sau alte procesări în domeniul timp, este suficient un DSP în virgulă fixă; pentru algoritmi complecși, cum ar fi convoluția, FFT sau în general procesări în domeniul frecvență, utilizarea unui DSP în virgulă

mobilă ar simplifica scrierea programului, ceea ce implică scurtarea timpului de dezvoltare a aplicației.

În cele din urmă, costul produsului final este de asemenea un element considerat în alegerea tipului de DSP. Alegerea unui DSP în virgulă fixă înseamnă utilizarea unei componente mai ieftine, dar în același timp costul dezvoltării aplicației poate fi mai mare datorită dificultății de implementare în virgulă fixă a unui algoritm complex. Astfel, procesoarele în virgulă fixă, mai ieftine, sunt utilizate cu precădere în produse de larg consum unde o diferență cât de mică de preț poate însemna succesul sau eșecul produsului; un exemplu bun pentru această situație sunt telefoanele celulare. Procesoarele în virgulă mobilă sunt preferate în produse performante, unde costul DSP-ului este insignifiant față de costul global al produsului; pentru acest caz, un exemplu bun sunt tomografele computerizate.

9.4. Programarea DSP-urilor

Programele pentru DSP-uri sunt dezvoltate în limbaj de asamblare sau în C, limbaje utilizate în general în aplicații științifice și de inginerie. Dacă în aplicații implementate pe procesoare de uz general raportul dintre numărul programelor în limbaj de asamblare și numărul programelor în C este de 1/10, în aplicații implementate pe DSP-uri, raportul poate fi 1/1-1/2.

Programele scrise pentru DSP-uri se deosebesc de aplicațiile soft tradiționale sub două aspecte:

- programele sunt de obicei mai scurte;
- viteza de execuție a programului este un aspect critic.

Aceste două aspecte fac ca pentru a programa un DSP să fie utilizat mai mult limbajul de asamblare decât în programarea procesoarelor de uz general.

În ceea ce privește venitul obținut pe seama unui DSP programat în limbaj de asamblare și venitul obținut pe seama unui DSP programat în C, raportul este de 2/1. Considerând performanța (viteza de execuție) și costul de fabricație, programarea în limbaj de asamblare este întotdeauna mai avantajoasă decât programarea în C; codul rezultat dintr-un program scris în C este întotdeauna mai mare decât codul unui program scris în limbaj de asamblare, ceea ce înseamnă că este necesară mai multă memorie, ceea ce înseamnă preț de fabricație mai mare pentru componentă. Tendința în prezent este de a fabrica DSP-uri optimizate pentru programare în C (set de registre de uz general, spațiu de memorie unificat).

În general, în cazul microprocesoarelor, și în particular în cazul DSP-urilor, programarea în limbaj de asamblare este mai dificilă. De aceea se caută dezvoltarea unor compilatoare C eficient și a altor instrumente software pentru eficientizarea codului. Practica curentă este de a scrie programe în principal în C, iar secvențele critice se scriu în limbaj de asamblare. Chiar dacă se alege programarea în C, este necesară cunoașterea unor elemente de arhitectură specifice DSP-urilor și de asemenea cunoașterea setului de instrucțiuni al limbajului de asamblare.

Care este limbajul potrivit pentru dezvoltarea unei aplicații particulare? Răspunsul depinde de ce este mai important pentru aplicație; dacă se impune flexibilitate și timp de

dezvoltare scurt, se alege C, iar dacă se cere performanță sub aspectul timp și utilizarea resurselor, se alege limbajul de asamblare.

Alte considerente sunt enumerate în continuare:

- pentru un program lung și imbricat se alege C, pentru un program mai scurt și mai simplu, se alege limbaj de asamblare;
- pentru un timp de execuție foarte scurt, în limbaj de asamblare se poate obține maximul de performanță; pentru aplicații care nu sunt critice sub aspectul timp de execuție, se alege C;
- pentru aplicații dezvoltate de mai mult de un programator se recomandă C cu scrierea în limbaj de asamblare a secvențelor critice sub aspectul timp;
- dacă primează costul resurselor hardware, se alege limbaj de asamblare, iar dacă primează costul de dezvoltare, se alege C;
- în funcție de experiența anterioară, un programator experimentat în limbaj de asamblare (pentru alte procesoare) va putea alege limbajul de asamblare; un programator experimentat în C va alege C;
- se recomandă a considera și recomandările fabricantului DSP-ului (unele DSP-uri pot fi programate foarte ușor în limbaj de asamblare, altele sunt foarte dificil de programat în limbaj de asamblare).

9.5. Piața DSP-urilor

Creșterea rapidă a pieței DSP-urilor este determinată de cererea mare de produse mai bune și mai ieftine ca: telefoane celulare, calculatoare multimedia, aparatură muzicală de înaltă fidelitate.

Evoluția pieței DSP-urilor este prezentată în Fig. 9.4.

DSP-urile pot fi cumpărate înglobate într-un modul de dezvoltare, ca circuit procesor sau doar a nucleu de procesare (core).

Nucleul de procesare (core) reprezintă secțiunea unde are loc procesarea efectivă. Acesta include registrele de date, multiplicatorul, unitatea aritmetică și logică și secvențiatorul de program. Nucleul de procesare este utilizat în aplicații croite exact pe cerințe particulare; se adaugă nucleului strictul necesar de resurse. Aceste aplicații reprezintă un segment important din piața DSP-urilor.

Procesorul conține pe lângă nucleul de procesare, memorie și interfețe pentru comunicare cu exteriorul. Toate elementele sunt implementate într-un singur circuit intergrat și sunt referite ca procesor.

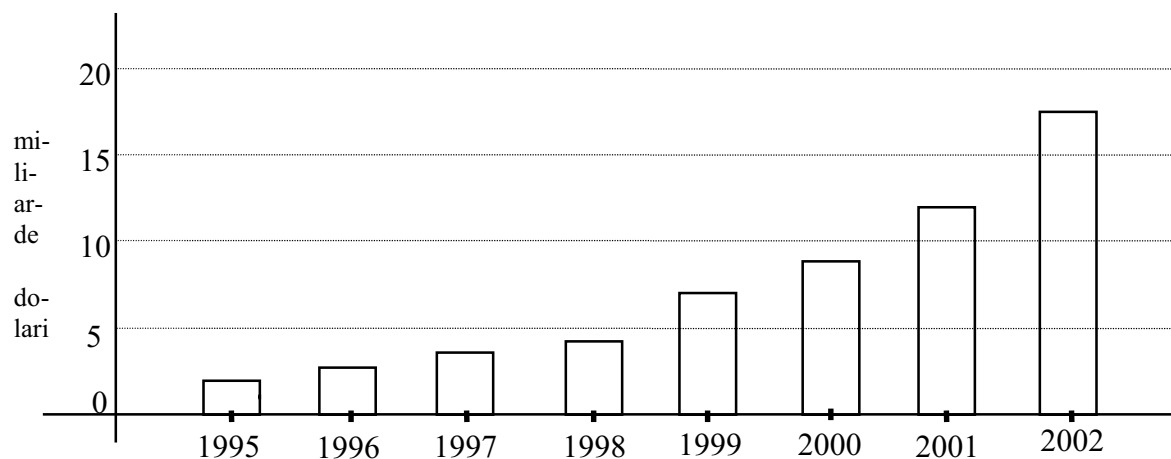


Fig. 9.4 Evoluția pieței DSP-urilor

Modulele de dezvoltare au procesorul DSP montat pe o placă de circuit imprimat pe care de asemenea mai sunt montate și alte resurse cum ar fi: memorie suplimentară, convertoare analog/digitale și digital/analogice, socluri pentru memorie EPROM sau Flash, o serie de controllere și interfețe populare, extensii pentru accesul la semnalele procesorului sau la semnalele magistralei modulului.

Astfel de module sunt fabricate de obicei de companii partenere fabricantului de DSP (Third Party Developer) despre existența cărora se poate afla consultând pagina de web a fabricantului circuitului DSP.

Piața DSP-urilor este dominată de câteva companii, lista acestora și indicativele de codificare a produselor sunt prezentate în continuare:

Analog Devices

ADSP 21xx	procesoare pe 16 biți în virgulă fixă
ADSP 21xxx	procesoare pe 32 biți în virgulă mobilă
ADSP 2199x	procesoare mixed signal
ADSP TigerSHARK101	procesoare pe 32 biți în virgulă mobilă

Motorola

DSP56300	procesoare pe 24 biți program, 24 biți date
DSP56800	procesoare 16 biți în virgulă fixă
DSP9600	procesoare pe 32 biți în virgulă mobilă
MSC8101	procesoare 16 biți în virgulă fixă

Texas Instruments

TMS 320C55x	procesoare pe 16 biți în virgulă fixă
TMS 320C64x	procesoare pe 16 biți în virgulă fixă
TMS 32C67x	procesoare pe 32 biți în virgulă mobilă

Granița între un procesor de tip DSP și un procesor de uz general începe să nu mai fie foarte clară; spre exemplu, procesoarele Intel dispun de instrucțiuni noi proiectate pentru a manipula și a procesa eficient informație video, audio și grafică. Aceste instrucțiuni sunt utile în procesarea paralelă a secvențelor repetitive specifice în operațiile multimedia. Fără îndoială, funcții și atribute specifice DSP-urilor vor fi înglobate în procesoarele de uz general, astfel termenul de DSP este posibil să se contopească în ceea ce numim astăzi microprocesor.

Pentru a avea o imagine actualizată a pieței DSP-urilor, și nu numai, consultarea revistelor ca EDN (Electronic Design News, www.adnmag.com) și ECN (Electronic Component News, www.ecnmag.com) este binevenită.

9.6. MSC8101

MSC8101 este primul reprezentant al unei familii de procesoare digitale de semnal (DSP) bazat pe unitatea centrală StarCore140. Acest procesor vine în întâmpinarea necesităților născute de aplicațiile de comunicații în rețea. MSC8101 este special proiectat pentru a fi utilizat într-o infrastructură de comunicații mobile de generația a III-a (3G).

G II - comunicații wireless cu următoarele caracteristici:

- viteza de transfer date de la 9.6 Kbps până la 28.8 Kbps;
- suportă semnal vocal și mesaje (SMS);
- afișare în mod text.

G III - comunicații wireless cu următoarele caracteristici:

- viteze de transfer date de la 384 Kbps până la 2 Mbps;
- suportă semnal multimedia;
- afișează imagini statice sau în mișcare

Principalele aplicații ale DSP-ului MSC8101 sunt poziții în infrastructura unei rețele de telecomunicații. Dintre sarcinile pe care le poate îndeplini un DSP amintim:

- În stațiile de bază (BSC) din infrastructura comunicațiilor mobile efectuează codarea de voce și/sau eliminarea ecourilor și nu numai sarcini de procesare de semnal clasice, ci poate realiza și anumite sarcini de interfațare cu rețeaua.
- În Gateway-urile media (voce, fax, date) DSP-ul realizează compresia vocală, eliminarea ecourilor, compresia de date, detecția și corecția de erori.

Principalele componente ale unui DSP MSC8101 sunt reprezentate în Fig. 9.5.

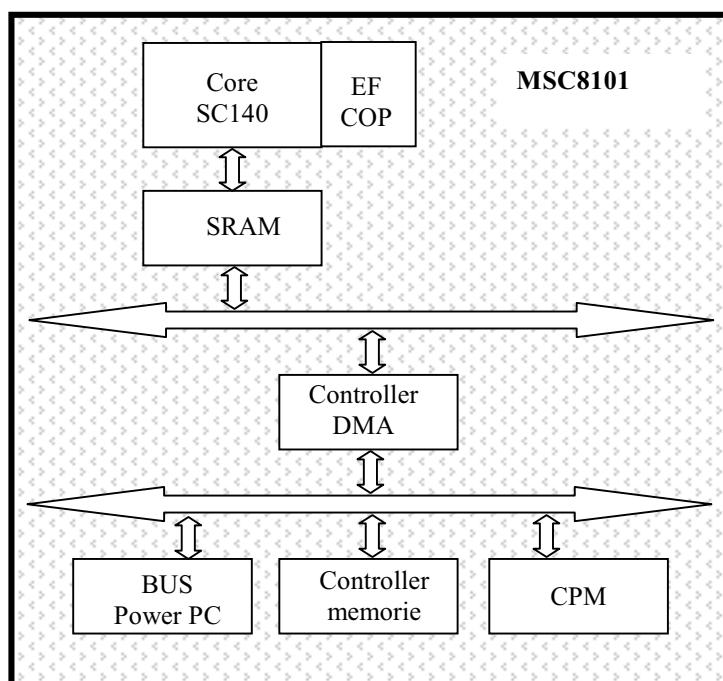


Fig.9.5 MSC8101 - Schema bloc generală

EFCOP - Enhanced Filter COProcessor

CPM - Communications Processor Module

În principal, MSC8101 este compus din core, memorie și un procesor de comunicație. Resursele comunică prin magistrale interne sub controlul unor blocuri funcționale specializate, ca în orice sistem performant; există în această idee controller de memorie, controller DMA și controller de magistrală, care în acest caz este PowerPC.

În continuare sunt prezentate pe scurt blocurile funcționale ale circuitului.

Core-ul SC140

Modulul de procesor SC140 pe 16 biți conține patru unități aritmetice și logice (ALU) fiecare constând dintr-o unitate MAC (Multiply and ACcumulate) și o unitate logică. Unitățile MAC sunt de înaltă performanță și pot fi folosite în implementarea algoritmilor de filtre FIR (Finite Impulse Response), IIR (Infinite Impulse Response) și a algoritmilor pentru FFT (Fast Fourier Transform). Toate cele patru ALU sunt identice. Pe lângă cele patru ALU core-ul mai conține două unități aritmetice de adresare AAU (Address Arithmetic Unit) cu moduri de adresare specifice DSP-urilor. SC140 poate iniția și executa până la 6 instrucțiuni într-un clock. Cele patru ALU și cele două AAU pot opera simultan în orice combinație. De exemplu, SC140 poate executa patru operații MAC într-un singur clock, sau o operație MAC, două operații aritmetice și logice și o operație pe bit. Faptul că cele patru ALU sunt identice conferă o mare flexibilitate în asignarea și executarea instrucțiunilor; unitățile de execuție pot fi ținute ocupate în orice ciclu, permițând astfel programului să beneficieze de arhitectura paralelă a procesorului.

Clock-ul maxim acceptat este de 300 MHz, iar la această viteză poate executa 1200 de operații DSP pe secundă. Un DSP MIPS este echivalentul a mai multor MIPS-uri RISC, măsură a performanței la alte DSP-uri.

SRAM

Capacitatea memoriei SRAM cu care este echipat circuitul este de 512KB, cu geometria 256Kx16 biți. Într-un ciclu la 300 MHz core-ul poate accesa simultan, fără stări de așteptare, 256 de biți (128 biți de date și 128 biți de program). Într-un ciclu al magistralei interne PowerPC este posibil accesul simultan la 64 biți. Capacitatea memoriei SRAM este suficient de mare pentru a stoca tot codul unui program și toate datele necesare acestuia în majoritatea aplicațiilor, eliminând astfel inconveniente legate de preț, performanță și spațiu ale unei memorii externe. Dacă totuși nu este suficientă memoria internă, MSC8101 poate adresa 4 GB de memorie externă prin magistrala PowerPC externă.

Controllerul DMA

Blocul de control DMA gestionează transferul de date între memoria internă și externă, perifericele interne și externe, pe 16 canale unidirecționale independente, fără intervenția core-ului. Transferul poate avea loc prin magistrala internă PowerPC, prin magistrala externă PowerPC sau între cele două magistrale.

Blocul DMA dispune de registre FIFO care pot stoca temporar date; de exemplu, dacă magistrala internă nu este liberă și DMA are de transferat date din memoria externă în

memoria internă, nu se așteaptă eliberarea magistralei, ci datele se transferă în registrele FIFO proprii blocului DMA urmând să fie transferate de aici în memoria internă atunci când magistrala internă este disponibilă.

O altă particularitate a blocului DMA din circuitul MSC8101 este modul de lucru *flyby* - datele sunt transferate direct de la sursă la destinație într-un singur ciclu. Acest mod poate funcționa între un periferic intern și memoria internă sau între un periferic extern și memoria externă.

CPM - Modulul procesor de comunicații

Modulul CPM este componenta care determină domeniul de aplicabilitate al MSC8101 - rețele și telecomunicații. CPM este un procesor care dispune de o arhitectură și de rutine microcod capabile să gestioneze protocoale de mare viteză cum ar fi ATM (până la 155 Mbps full-duplex) și Fast Ethernet (până la 100 Mbps full duplex).

CPM conține următoarele blocuri funcționale:

- o unitate RISC pe 32 de biți cu un set de instrucțiuni optimizat pentru comunicații;
- 2 controllere seriale DMA (SDMA) prin care pot fi executate două transferuri simultan;
- 3 controllere de comunicații seriale rapide FCC (Fast Communication Controller) care suportă protocoalele IEEE 802.3 și Fast Ethernet. Fiecare FCC poate fi configurat să transmită în mod transparent și să recepționeze în mod HDLC și invers. Două FCC au pini dedicați, iar al treilea operează în mod TDM (Time-Division Multiplex); FCC1 suportă protocolul ATM de 155 Mbps prin intermediul interfeței UTOPIA2;
- 2 controllere multicanal MCC (Multi-Channel Controller) care pot gestiona canale HDLC în limitele 256x64 Kbps multiplexate în până la patru interfețe TDM;
- 4 controllere de comunicații seriale full-duplex SCC (Serial Communication Controller). Cele patru controllere suportă următoarele moduri de comunicație: IEEE 802.3, HDLC, UART, Synchronous UART, BISYNC și mod transparent. Două SCC au pini dedicați, celelalte două pot opera doar în mod TDM;
- 2 unități de management al transferului serial (Serial Management Controller) full-duplex care suportă interfețe generale de circuit GCI (General Interface Circuit), UART și operații transparente;
- controllere de interfață SPI (Serial Peripheral Interface);
- controller de magistrală I²C (Inter-Integrated Circuit);
- 8 generatoare independente de baud-rate și 10 pini de intrare pentru semnale de clock pentru sincronizare necesară canalelor seriale FCC, SCC și SMC;
- 4 timere-e de 16 biți independente care pot fi interconectate ca doua timer-e de 32 de biți.

EFCOP - Procesorul îmbunătățit pentru filtre

EFCOP realizează operațiile de filtrare și eliminare a ecourilor, funcții vitale pentru un DSP. Operațiile de filtrare includ filtrări neadaptive și filtrări adaptive, filtre FIR și IIR. Coprocesorul lucrează în precizie de 32 de biți și conține o unitate de multiplicare 32*32 și un acumulator de 72 de biți. Circuitele din EFCOP pot realiza într-un ciclu de clock un pas de filtrare cu un consum foarte mic. EFCOP poate actualiza coeficienții într-un filtru adaptiv.

EFCOP lucrează în paralel cu core-ul procesorului, ceea ce înseamnă că operații ca anularea ecourilor pot fi procesate în paralel cu operațiile de compresie a vocii, facilitate utilă în telefonia prin Internet. Considerând un indice de utilizare de 70% al EFCOP, o medie tipică pentru aplicațiile DSP, contribuția la performanța sistemului este de 210 MIPS peste cei 1200 MIPS ai core-ului SC140.

HDI16 - Interfața gazdă

Pe lângă interfața PowerPC, MSC8101 are și o interfață HDI16 care suportă o varietate mare de magistrale standard și permite interconectarea comodă cu microcontrollere, microprocesoare sau alte DSP-uri. Interfața HDI16 poate fi utilizată în mod concurrent cu interfața PowerPC, ceea ce face ca sistemul să fie flexibil în proiectare. De exemplu, în sistemele este vehiculată o cantitate mare de informație între un procesor gazdă și un banc de procesoare MSC8101, DSP-urile vor comunica cu procesorul gazdă prin interfața PowerPC în mod 64 de biți; în sistemele cu trafic mai redus, procesorul gazdă și DSP-ul pot comunica prin interfața HDI16 pe 16 biți, și în același timp DSP-ul poate fi conectat la o memorie partajată prin intermediul interfeței powerPC în mod 32 de biți.

Nucleul de procesare SC140

StarCore este numele core-ului de procesor proiectat de centrul de dezvoltare cu același nume înființat de Motorola și Lucent (actuală Agere) Technologies în anul 1998 în vederea producerii unui core de procesor care să echipeze generațiile următoare de DSP-uri. Prima arhitectură StarCore a fost anunțată în anul 1999, iar prima implementare a acestei arhitecturi a fost core-ul SC140.

SC140 excelează prin gradul înalt de paralelism de care dispune; clock-ul maxim cu care poate lucra este de 300 MHz. SC140 este în prezent (2002) cel mai rapid DSP de uz general realizat în siliciu. SC140 este un core de DSP cu un consum redus de putere. Din punct de vedere al programării, core-ul este foarte flexibil și compilatoarele existente realizează coduri eficiente și compacte. În execuția instrucțiunilor este implementat conceptul VLES (Variable Length Execution Set - set de execuție cu lungime variabilă).

Performanța SC140, relativ la procesoarele DSP Motorola, este ilustrată în Fig. 9.6.

SC140 a fost proiectat cu scopul de a oferi o platformă care să satisfacă cerințele crescânde ale aplicațiilor DSP datorate următoarelor condiții:

- standarde și servicii de comunicații noi;
- canale de bandă largă și rată mare de transfer;
- interfețe noi pentru utilizatori și media.

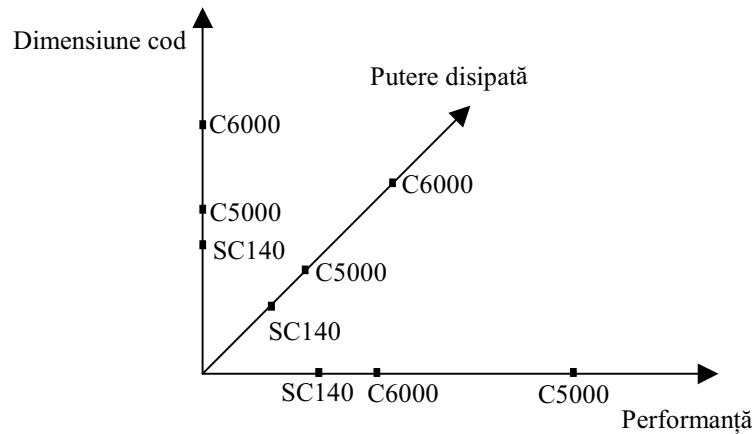


Fig. 9.6 Caracteristici SC140 raportate la procesoarele C5000 și C6000

SC140 este gândit pentru a fi utilizat în aplicații de mare performanță în comunicații mobile, cum ar fi gateway-uri, stații de bază și aplicații portabile (terminale celulare). Aplicațiile către care țintește SC140 sunt:

- wireless, internet, multimedia;
- comunicații de date în rețea;
- infrastructura wireless;
- sisteme handset wireless de generația a treia cu servicii de date de bandă largă;
- terminale handset configurabile prin software;
- codarea, sinteza și recunoașterea vocii;
- codarea semnalului muzical.

Dintre însușirile care diferențiază SC140 pe piață pot fi amintite următoarele:

- permite un nivel înalt de abstractizare a software-ului pentru aplicații:
 - ⇒ aplicațiile pot fi dezvoltate în C; un compilator de optimizare generează instrucțiuni paralele și menține o densitate mare a codului;
 - ⇒ setul ortogonal de instrucțiuni permite compilatoarelor să genereze cod eficient;
 - ⇒ hardware-ul suportă reprezentarea numerelor întregi și subunitare cu sau fără semn, modul de reprezentare este la alegerea programatorului.
- oferă performanță scalabilă:
 - ⇒ SC140 dispune de șase unități de execuție (patru unități aritmetice și logice și două unități de adresare);
 - ⇒ la o alimentare redusă sunt posibile patru milioane de operații MAC (Multiply and Accumulate) pe secundă pentru fiecare MHz din frecvența clock-ului;
 - ⇒ dispune de suport pentru implementarea de acceleratori de aplicație.
- se poate obține densitate mare de cod datorită:
 - ⇒ unui set de instrucțiuni bogat și ortogonal;
 - ⇒ execuției de tip VLES;
 - ⇒ codarea instrucțiunilor pe 16 biți.

- dispune de suport îmbunătățit pentru aplicații multi-tasking:
 - ⇒ hardware-ul dispune de suport optimizat de stive;
 - ⇒ magistralele de date sunt largi și oferă suport pentru schimbarea de context.
- Optimizarea managementului puterii (proiectat pentru consum de energie redus, moduri noi de economisire a energiei, clock de control dinamic);
- dispune de o interfață eficientă pentru memorie și dispozitive I/O:
 - ⇒ spațiu unificat pentru memoria de date și program;
 - ⇒ spațiu de adresare pe 32 de biți pentru date și program;
 - ⇒ SRAM cu zero wait-state pe chip;
 - ⇒ clock independent pentru memoria externă.
- Organizarea core-ului suportă configurații SOC (System On-Chip) flexibile.

Un exemplu de configurație SOC este prezentat în Fig. 9.7.

SC140 poate să execute șase instrucțiuni simultan (patru operații aritmetice și logice și două operații de adresare). Instrucțiunile grupate pentru a fi executate în paralel sunt referite ca set de execuție. Instrucțiunile sunt grupate pentru execuție în paralel în timpul compilării sau de către programatorul în limbaj de asamblare.

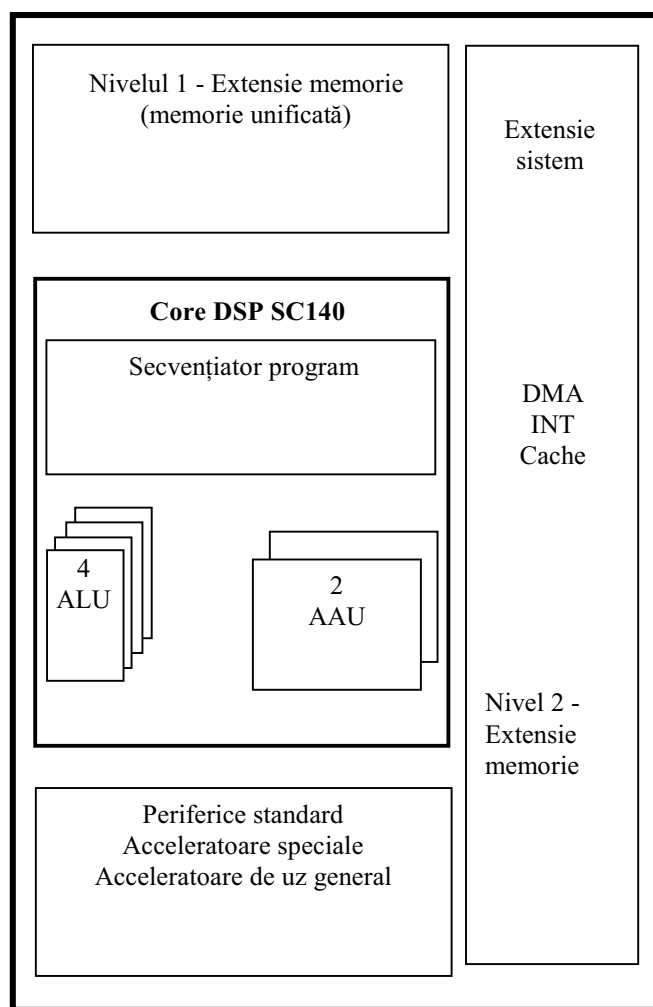


Fig. 9.7 Configurații SOC cu SC140

Schema bloc a SC140

Schema bloc este reprezentată în Fig. 9.8.

Principalele blocuri funcționale ale SC140 sunt:

- Unitatea aritmetică și logică de date - DALU (Data Arithmetic and Logic Unit);
- Unitatea de generare de adrese - AGU (Address Generator Unit);
- secvențiatorul de program - PSEQ (Program SEQuencer).

Schimbul de date între core și blocurile funcționale din circuitul DSP este realizat prin următoarele magistrale:

- două magistrale de memorie (două perechi adrese/date: ABA, DBA și ABB, DBB) pentru a accesa datele din memorie;
- două magistrale de memorie (PAB, PDB) pentru a transfera cuvinte instrucțiune;
- magistrale speciale pentru conectarea în imediata apropiere a core-ului un set de instrucțiuni de accelerare.

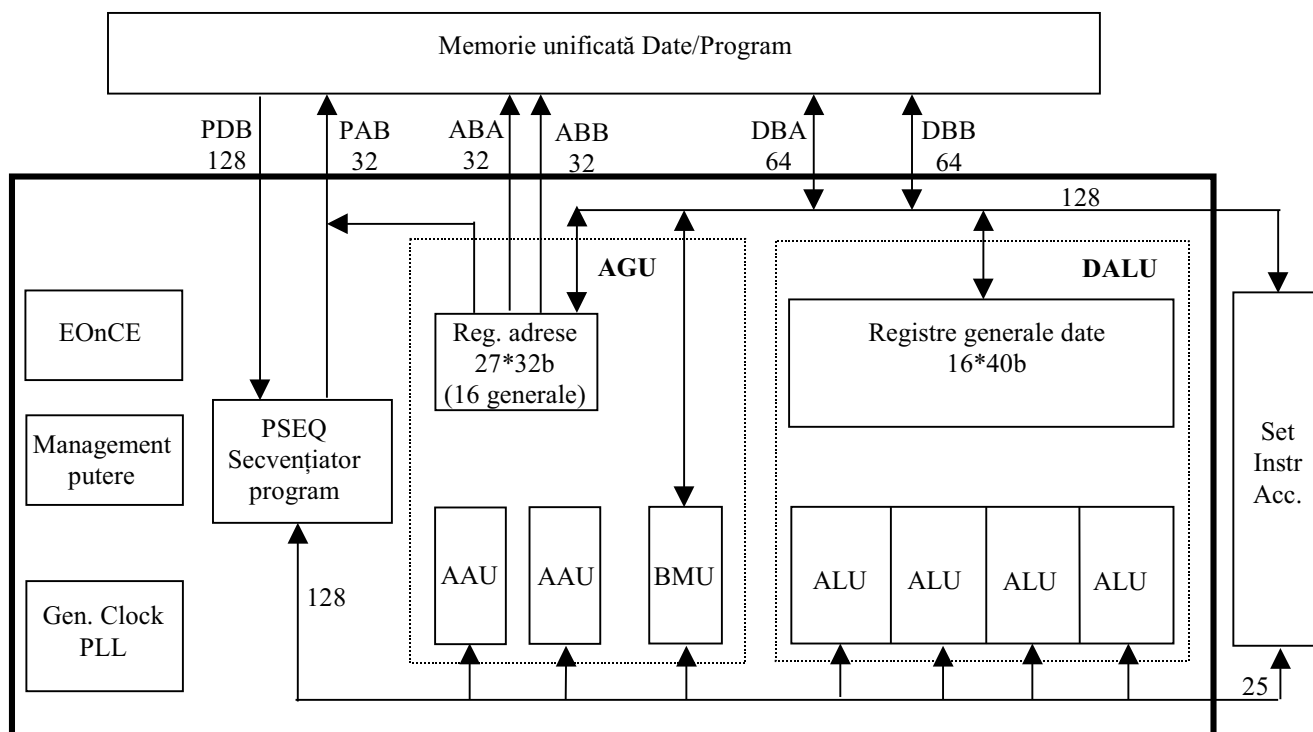


Fig. 9.8 SC140 - Schema bloc funcțională

Unitatea aritmetică și logică de date - DALU

DALU execută operații logice și aritmetice asupra operanzilor. Componentele DALU sunt:

- un bloc de 16 registre de 40 biți;
- patru unități aritmetice și logice (ALU) fiecare conținând câte o unitate MAC (Multiply and ACcumulate) și câte o unitate BFU (Bit Field Unit);
- opt registre de deplasare/limitare ca legătură a blocului de 16 registre generale de 40 de biți spre magistrala de date cu memoria de 128 de biți.

Unitățile MAC pot realiza multiplicări 16×16 întregi sau subunitare de numere cu semn sau fără semn; produsul este rotunjit la dreapta, se extinde semnul și poate fi adunat într-un registru general de 40 de biți. Într-un ciclu de clock pot fi executate patru instrucțiuni ALU simultan.

Unitățile BFU conțin câte un registru de 40 de biți bidirecțional, o unitate de generare de mască și o unitate logică. BFU sunt folosite în următoarele operații:

- deplasare pe mai multe poziții (dreapta/stânga);
- rotire cu un bit;
- inserarea sau eliminarea unui bit;
- numărarea de biți 1 sau 0;
- operații logice;
- extensie de semna sau 0.

Registrele generale de date pot fi scrise/citite prin magistralele DBA, DBB. Un registru poate fi sursă pentru patru operații ALU (simultane), dar nu poate fi destinația a mai multor operații simultane. Operanzii aritmetici provin de obicei din registrele generale; un registru poate fi de asemenea destinația pentru rezultatul unei operații, și acest registru poate fi folosit ca sursă în operația imediat următoare.

Datorită faptului că cele două magistrale de date ale DALU spre memorie sunt fiecare de câte 64 de biți, (DBA, DBB) este posibil să se execute în paralel două transferuri de date, MOVE, fiecare de câte un cuvânt de până la 64 de biți sau transferuri multiple în limita lărgimii de 64 de biți. Sunt posibile următoarele tipuri de transferuri:

- MOVE.B - transferă cuvinte de 8 biți;
- MOVE.W sau MOVE.F - transferă întregi, respectiv numere subunitare reprezentate pe 16 biți;
- MOVE.2W, MOVE.2F sau MOVE.L - transferă două numere întregi, două numere subunitare, respectiv cuvinte lungi, fiecare de câte 32 de biți;
- MOVE.4W sau MOVE.4F - transferă patru întregi, respectiv patru numere subunitare reprezentate fiecare pe câte 64 de biți;
- MOVE.2L - transferă două cuvinte lungi, adică 2×32 biți.

Structura permite executarea a oricărei combinații de două instrucțiuni MOVE în paralel într-un ciclu de clock, astfel rata maximă de transfer cu memoria de date este de 4.8 GBytes/s (la 300 MHz).

Unitatea de generare a adreselor - AGU

AGU conține registre de adrese și execută calculul adreselor pentru accesarea memoriei utilizând aritmetica numerelor întregi. AGU operează în paralel cu alte resurse ale core-ului.

AGU dispune de două unități aritmetice de adresare (AAU - Address Arithmetic Unit) cu care poate genera două adrese de memorie în fiecare clock. Cele două unități AAU sunt identice. Într-un ciclu de execuție pot fi executate două operații de adresare în paralel.

Blocul de registre de adrese este compus din mai multe registre de 32 de biți, dintre care 16 sunt de uz general iar celelalte sunt utilizate pentru a specifica tipul operației utilizate pentru în calculul adresei. O unitate AAU actualizează un registru de adresare într-un ciclu de clock. SC140 dispune de instrucțiuni explicite pentru a opera indicatorii de adresă, capabilitate care poate fi utilizată și pentru a efectua calcule aritmetice asupra datelor.

AGU generează adrese pentru salturi în program și actualizează indicatorii de stivă corespunzător. În blocul registrelor de adresare există două registre indicator de stivă, fiecare de câte 32 de biți; existența a doi indicatori de stivă permite alocarea separată a stivei utilizate de către sistemul de operare și de fiecare aplicație ce rulează, ceea ce optimizează utilizarea memoriei în sistem multi-tasking.

Unitatea pentru mască de bit BMU (Bit Mask Unit) facilitează operații de setare, resetare, inversare sau testare a unui grup de biți dintr-un registru sau dintr-o locație de memorie (biții pot să nu fie adiacenți). Deoarece există o singură unitate BMU, într-un ciclu de execuție poate fi executată o singură instrucțiune cu mascare de bit. Un subgrup al instrucțiunilor cu mască de bit (BMTSET) furnizează suport pentru semaforizare.

Unitatea secvențiator de program - PSEQ

PSEQ este unitatea care execută fetch-ul instrucțiunii, dispatch-ul instrucțiunilor, controlul buclelor hardware, procesarea excepțiilor și este compus din trei blocuri hardware:

- PDU (Program Dispatch Unit) - detectează setul de execuție dintr-unul sau două cicluri fetch și alocă instrucțiunile blocurilor funcționale unde acestea vor fi decodificate;
- PCU (Program Control Unit) - controlează desfășurarea programului;
- PAG (Program Address Generator) - generează conținutul pentru numărătorul de program pentru fetch-ul instrucțiunilor, inclusiv buclele hardware.

Unitatea de emulare în circuit - EOnCE

EOnCE (Enhanced On-Chip Emulator) este modulul care permite interacțiunea cu core-ul SC140 și cu periferia printr-o interfață JTAG. Utilizatorul poate examina registrele, memoria sau perifericele din circuit, facilitate utilă atât în dezvoltarea hardware cât și dezvoltarea de software.

Setul de instrucțiuni de accelerare

Setul de instrucțiuni de bază SC140 poate fi completat de utilizator în scopul creșterii performanței în aplicații particulare. Instrucțiunile suplimentare sunt executate într-un modul

exterior core-ului; noile instrucțiuni sunt adăugate asamblorului și compilatorului prin biblioteci intrinseci.

Setul de instrucțiuni

Setul de instrucțiuni a fost proiectat în așa fel încât executarea paralelă a algoritmilor DSP și a codului de control să fie cât mai eficientă. Pentru execuția instrucțiunilor în paralel trebuie respectate reguli de programare deoarece resursele utilizate în paralel sunt limitate; deoarece există patru ALU și două AAU, un set de execuție poate să conțină cel mult patru instrucțiuni ALU sau cel mult două instrucțiuni AGU.

Setul de instrucțiuni SC140 este destul de ortogonal deoarece majoritatea instrucțiunilor sunt simple și specifică o singură operație. Spre deosebire de unele procesoare VLIW (Very Long Instruction Word), setul de instrucțiuni SC140 este format din instrucțiuni destul de scurte, de 16 biți, a căror funcționalitate poate fi extinsă folosind prefixe sau extensii. Instrucțiunile scurte, de multe ori determină proiectanții procesoarelor să impună restricții pentru folosirea registrelor. La SC140 sunt evitate restricțiile utilizării registrelor prin utilizarea cuvintelor de prefix. Cu excepția instrucțiunilor speciale pentru decodare Viterbi, toate instrucțiunile SC140 sunt fără restricții. Există în schimb restricții în gruparea instrucțiunilor într-un set de execuție, ceea ce complică programarea în limbaj de asamblare.

Pentru execuția rutinelor pentru procesarea de semnal, instrucțiunile pot fi grupate pentru a fi executate în paralel. Grupul de instrucțiuni executate în paralel reprezintă un set de execuție, fiecare instrucțiune din setul de execuție utilizează o unitate de execuție. Instrucțiunile din cadrul aceluiași set de execuție își încep execuția în același moment; un nou set își începe execuția numai după ce s-a terminat execuția tuturor instrucțiunilor din setul anterior. Sunt utilizate două metode pentru a specifica faptul că o instrucțiune face parte dintr-un set de execuție:

- grupare serială- folosește cei mai semnificativi doi biți din instrucțiune pentru a determina sfârșitul unui set de execuție;
- grupare prefixată- se adaugă un prefix de unul sau două cuvinte unui set de execuție care definește numărul de instrucțiuni incluse în setul de execuție.

Gruparea instrucțiunilor pentru execuție în paralel folosește modelul VLES (Variable Length Execution Set). Acest model menține densitatea codului compactă pentru aplicații. Modelul VLES :

- definește un mecanism pentru specificarea lungimii unui set de execuție;
- adaugă biți adiționali la generarea codului doar dacă este necesar;
- codul este generat transparent de către asamblor.

Unelte de dezvoltare a programelor

În generarea codului pentru SC140, programatorul este asistat de:

- SC100 C Compiler;
- SC100 Assembler;
- SC100 Linker;
- SC100 Utilities;

- Sc100 Simulator.

Compilerul C are proprietatea de a transfera codul sursă C într-un cod de asamblare optimizat. Compilerul permite schimbarea nivelului de optimizare, pentru rezultatele cele mai bune, în final, se aplică optimizarea globală. Compilerul C are următoarele caracteristici:

- este conform cu standardul ANSIC;
- are medii și biblioteci runtime;
- permite integrarea ușoară a secvențelor de programare.

Asamblorul convertește codul de asamblare scris de programator sau generat de compiler în fișiere obiect cu extensia ELF. Caracteristicile asamblorului sunt:

- evaluarea expresiilor folosind constante de tip șir și numerice, operatori și funcții încorporate;
- programarea modulară folosind secțiuni;
- macrouri care permit variabile ca argument;
- asamblare condițională.

Link-editorul combină fișierele obiect și arhivă, realocă datele lor, ajustează referința la simboluri și generează un fișier executabil. Link-editorul permite ca un program lung să se poată desface în module mai mici, manipulabile mai ușor, care pot fi asamblate separat.

Utilitarele SC100 folosesc la procesarea sau interpretarea fișierelor obiect; ele se numesc Archiever, Disassembler, ELF File Dump, Name, Size.

Simulatorul SC100 este o unealtă interactivă de simulare care permite măsurarea timpului de execuție cu acuratețea ciclurilor de clock. Simulatorul duplică funcțiile core-ului SC140 incluzând actualizarea memoriei și a registrelor asociate cu execuția codului și activitatea de procesare a excepțiilor. Funcțiile principale ale simulatorului sunt:

- încarcă fișierele executabile în memoria simulatorului;
- citește valorile din memorie pentru un fișier de intrare sau salvează valorile din memorie într-un fișier de ieșire;
- raportează sesiunile de simulare;
- examinează sau schimbă conținutul registrelor și al memoriei;
- controlează execuția programului prin:
 - ⇒ setare de breakpoint-uri;
 - ⇒ salt peste un număr specificat de instrucțiuni.

Modulul de dezvoltare MSC8101 ADS

Modulul MSC8101 ADS (Application Development System) este o platformă pentru dezvoltări software și hardware de aplicații cu procesorul DSP MSC8101. Utilizând resursele modulului și debuggerul, este posibilă încărcarea și rularea de cod executabil, se pot seta breakpoint-uri, poate fi vizualizat conținutul memoriei și al registrelor. Conectarea la modul este posibilă prin magistrale și conectori de extensie.

În continuare sunt prezentate principalele componente ce intră în echiparea modulului:

- controller MSC8101 - StarCore, 64b, 300MHz
- magistrală PowerPC - 100 MHz
- 8M Flash SIMMx32, 16M SDRAMx32

- FPGA pentru BSCR (Board Control and Status Register)
- conectori de extensie pentru analizor logic MICTOR
- interfață optică ATM - 155Mbps
- port Ethernet10/100-Base-T

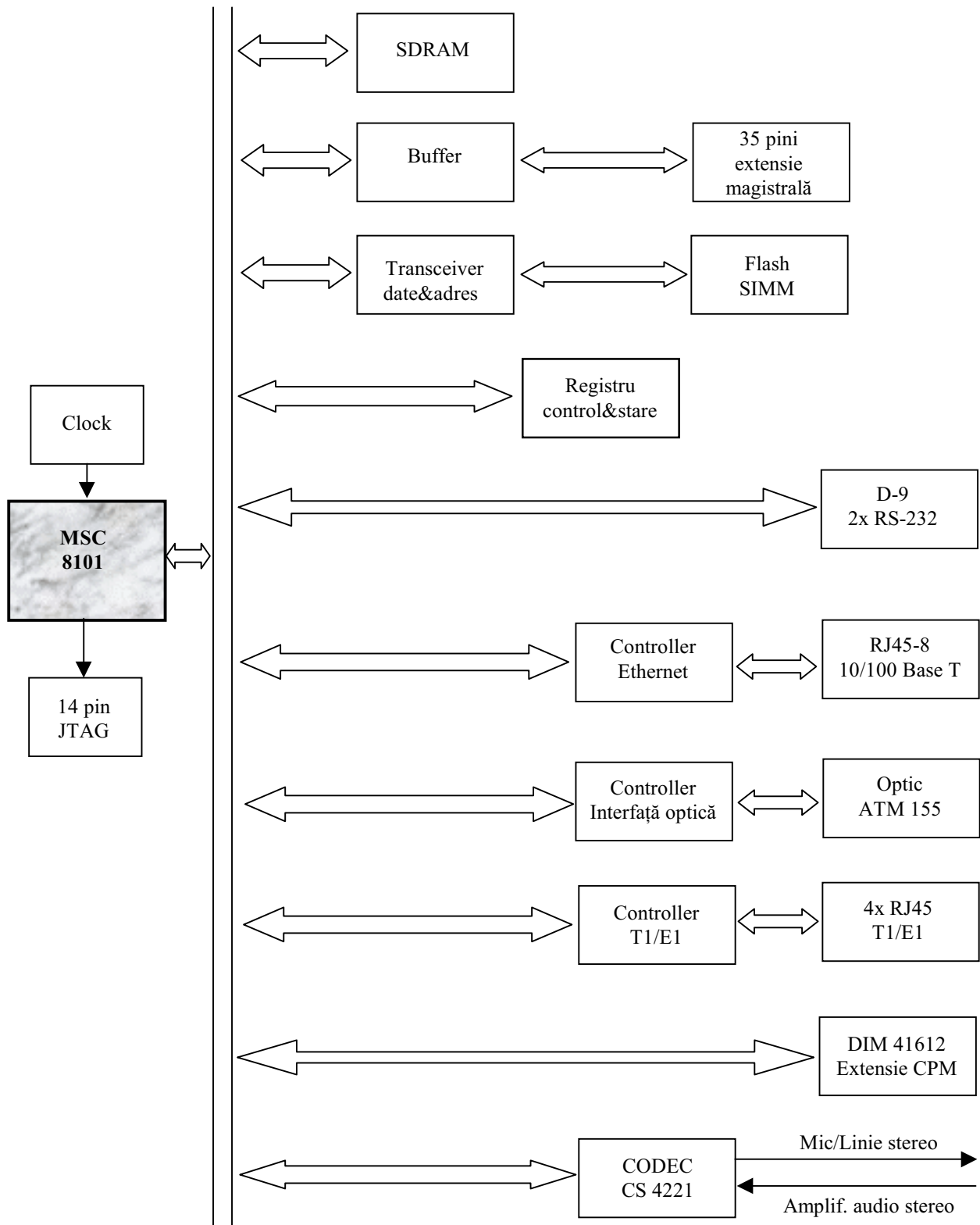


Fig. 9.9 Schema bloc a modului MSC 8101 ADS

- 4 canale T1/E1
- CODEC audio stereo pe 24 biți
- 2 porturi RS-232
- port JTAG

Codul aplicației software pentru MSC8101 poate fi transferat modulului prin portul JTAG. Scrierea codului și generarea executabilului necesită un calculator compatibil PC pe care să ruleze unul din sistemele de operare Solaris, Windows NT 4.0 sau Windows 9x. Pentru scrierea și depanarea programului este utilizat un mediu integrat de dezvoltare.

Schema bloc a modulului este reprezentată în Fig. 9.9.

Imaginea modulului este prezentată în Fig. 9.10.

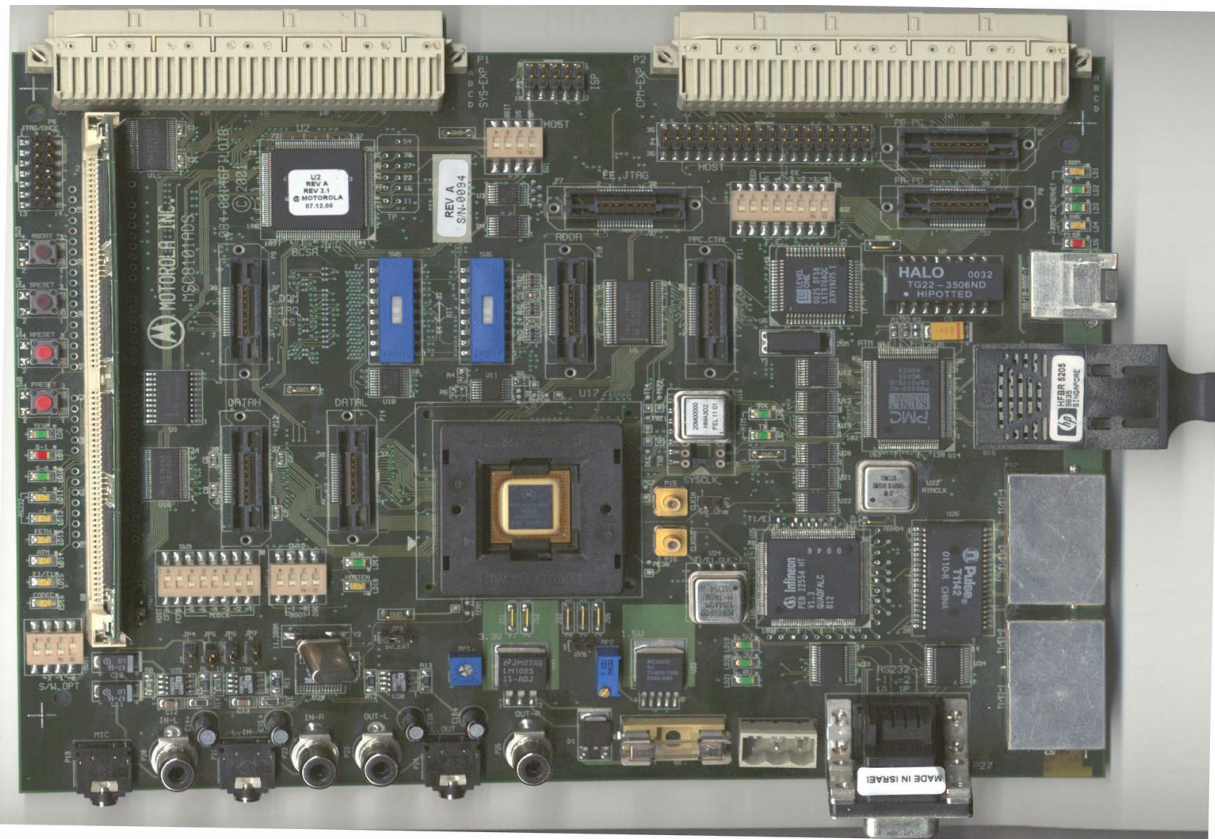


Fig.9.10 Modulul de dezvoltare MSC8101 ADS

Modulul de dezvoltare poate opera în trei moduri:

- operare controlată de un calculator gazdă prin portul OnCE (un subset al portului JTAG);
- operare prin interfața HDI16;
- operare independentă.

În modul de operare controlată, se realizează o conexiune între interfața JTAG/OnCE a modulului și portul paralel al unui calculator PC printr-un convertor de interfață. Această configurație permite depanarea aplicațiilor care rulează pe modul utilizând un debugger care rulează pe calculatorul PC.

În modul de operare prin interfața HDI16, modulul folosește o interfață HDI16 conectată direct la magistrala de date a unui procesor gazdă. În acest mod de operare este posibilă o legătură full-duplex pe 16 biți. Interfața HDI16 suportă o conexiune cu un microprocesor/microcontroller sau cu un controller DMA.

În modul de operare independentă, modulul nu este conectat la calculatorul gazdă prin portul JTAG/OnCE, ci prin una din interfețele RS-232, T1/1, ATM155 (optic) sau Ethernet. În modul de operare independent este necesare o sursă de boot-are și software care să controleze interfața utilizată.

Bibliografie

1. Steven W. Smith, "The Scientist and Engineer's Guide to DSP", California Technical Publishing, San Diego California, 1999
2. *****, SC 140 Reference Manual, Motorola, 2001
3. *****, MSC 8101 User's Manual, Motorola, 2001
4. *****, MSC 8101 ADSUM/D, User's Manual, Motorola, 2001