Creşterea vitezei de execuție a algoritmilor intensiv computaționali cu ajutorul coprocesoarelor ataşate, bazate pe structuri logice reconfigurabile (FPGA).

4.1.Introducere.

Având în vedere limitările ce caracterizează prelucrarea serială, încă de la început, cercetările în domeniul Calculului de Înaltă Performanță (CIP) s-au orientat către noi abordari, bazate pe procesaore mai performante, pe sisteme multiprocesor, pe prelucrarea paralelă s.a. Aceste noi abordari s-au lovit, inițial, de o serie obstacole în privința paralelizării limitate şi a vitezei scăzute a comunicațiilor între procesoare. Toate acestea au condus, în etapa de început a CIP, la realizarea de sisteme cu un număr redus de procesoare, de ordinul zecilor. Pe măsura dezvoltarii cercetărilor în domeniu s-au găsit soluții pentru obstacolele menționate mai sus: legături de bandă largă între procesoare, noi algoritmi care au condus la un schimb de date mai puțin intens între procesoare, paralelism grosier, procesoare performante la costuri scăzute, soluții care au facilitat apariția unor sisteme de prelucrare constituite din sute de procesoare.

Cu toate că stadiul actual al domeniului CIP se caracterizează prin existența unor sisteme formate din mii de procesoare, cu interconexiuni de bandă largă şi latență redusă, aceste sisteme prezintă limitări pentru anumite aplicații. Astfel, în cele din urmă, regia impusă de calculul paralel prevalează în raport cu avantajele aduse de multiprelucrare, performanța tinzând asimptotic către o limită dată.

În dorința de a depăsi limitările amintite, cercetătorii s-au orientat către noi tipuri de procesoare, bazate pe Arii de Porți Reprogramabile (Field Programmable Gate Array- FPGA). Acestea, spre deosebire de paralelismul grosier, implementat pe sisteme multiprocesor, oferă noi oportunități de implementare a calculului: paralelism fin-masiv şi bandă de asamblare, în cadrul procesorului FPGA individual. FPGA permite realizarea temporară a unui circuit la cerere, care soluționeaza o problemă concretă, în condițiile generării unor resurse hardware suficiente, pentru a exploata paralelismul intrinsec, la nivelul fluxului de date, al algoritmului. Astfel, într-un singur ciclu de ceas se pot executa toate calculele pentru care datele sunt disponibile, ceea ce ofera avantaje substanțiale în raport cu prelucrarea serială tradițională. Procesoarele bazate pe FPGA nu sunt lipsite de inconveniente. Acestea se manifestă prin dificultățile de implementare a sarcinilor microprocesoarelor convenționale, legate de rularea sistemelor de operare şi execuția de operații seriale cum ar fi conectarea la rețele, citirea şi scrierea de la/ la unitățile de discuri. Astfel, se explică

prezența unui microprocesor/microcontroler în asociere cu un circuit FPGA, pe aceeați plachetă, pentru implementarea acestor operații. Placheta se conectează la un sistem gazdă prin porturi diferite (serial, paralel, USB, ethernet etc) sau pe magistrale de tip ISA, PCI s.a., jucând rolul de coprocesor accelerator. În vederea evaluarii corecte a soluției "coprocesor FPGA", pentru o aplicație dată, trebuie examinate limitările de bandă, atât ale hardware-lui, cât şi ale algoritmului, în ceea ce priveşte transferul datelor între sistemul gazda şi coprocesorul FPGA. Dacă acesta din urmă este lent, nu se pot pune în valoare avantajele utilizarii coprocesorului FPGA, pentru accelerarea calculelor.

În cele ce urmează se vor examina câteva tehnici de evaluare a unor asemenea soluții.

4.2. Decizii privitoare la hardware.

Circuitele FPGA au fost inițial utilizate în sistemele încorporate (embedded) pentru înlocuirea logicii convenționale, realizată cu circuite integrate pe scară simplă/medie, cu un singur circuit FPGA.

Pe măsura ce complexitatea circuitelor FPGA a crescut, ele au fost plasate pe plachete dedicate, în scopul cuplării lor la sisteme gazdă, construite cu ajutorul microprocesoarelor convenționale. Astfel, s-a conturat ideea utilizării acestor plachete ataşate pentru a accelera unele taskuri ale aplicațiilor rulate pe sistemul gazdă, taskuri caracterizate prin posibilități de paralelizare. Aceste posibilități nu ar fi putut sa fie exploatate de procesorul serial, din sistemul gazda.

Pentru a obține performanța impusă de aplicație, este necesar să se asigure o conexiune cu lărgime mare de bandă între placheta ataşata şi sistemul gazdă. Plachetele ataşate, conectate la sistemele gazdă, prin magistrale relativ lente (VME, PCI), care nu suportă operații aritmetice pe 64 de biți, din cauza resurselor hardware limitate, nu satisfac necesitățile impuse de CIP. Stadiul actual al dezvoltării domeniului FPGA asigură resursele hardware necesare pentru implementarea a zeci de unități funcționale pe 64 de biți. Astfel, utilizând circuitul Xilinx- Virtex-4 LX200 se pot implementa, theoretic, circa 70 unități de înmultire în virgulă mobilă sau 128 de sumatoare în virgulă mobilă [42], [43].

În ceea ce priveşte conexiunea de bandă largă între sistemul gazda şi placheta ataşată, s-a apelat la interfețe mai rapide cum sunt PCI-X sau HT (Hyper Transport). Dacă în cazul interfeței PCI se lucra la o frecvență de 33MHz şi o lungime de cuvânt de 32 de biți, ceea ce asigura o lărgime totală de bandă de 133MB/s, în cazul interfeței PCI-X frecvența de lucru este de 133MHz, cu cuvinte de lungime de 64 de biți, ceea ce conduce la o lărgime totală de bandă de 1.064 MB/s. Interfața HT,

care se bazează pe un protocol de conectare a dispozitivelor direct la procesor, avand o latență minimă, asigură o lărgime de bandă de 1,6 GB/s, în fiecare direcție.

În continuare se vor evalua performanțele teoretice ale celor trei interfețe: PCI, PCI-X şi HT [44]. Se va presupune că în memoria sistemului gazdă se află stocate 1.000 de numere în virgulă mobilă, pe 64 de biți, la care se intenționează să se adune, cu ajutorul plachetei ataşate un numar constant, pe 64 de biți, în virgulă mobilă. Un număr va avea 8 octeți, deci la transferul cu placheta ataşată se vor transmite 8.000 octeți într-un sens şi tot atâția în sens invers. Pentru a calcula durata transferurilor, în ambele sensuri, se va folosi formula:

$$T_{interfață} = 2 \times N_{r.}Octeți/Lărgimea\_de\_bandă_{interfață} \quad (4.1)$$

ceea ce se va concretiza în:

$T_{PCI} = 2 \times N_{r.}Octeți/Lărgimea\_de\_bandă_{PCI} = 2 \times (8.000/133) \times 1024 \times 1024 \approx 114,80\ \mu s$

$T_{PCI-X} = 2 \times N_{r.}Octeți/Lărgimea\_de\_bandă_{PCI-X} = 2 \times (8.000/1.064) \times 1024 \times 1024 \approx 14,40\ \mu s$

Interfața HT poate transfera simultan în ambele direcții:

$T_{HT} = N_{r.}Octeți/Lărgimea\ \_de\_bandă_{HT} = (8.000/1,6) \times 1024 \times 1024 \times 1024 \approx 4,70\ \mu s$

În continuare se neglijează timpul de adunare în circuitele implementate pe placheta ataşată.

Se consideră că operațiile vor avea loc în sistemul convențional gazdă, care: funcționează la o frecvență de 2,5 GHz, manipulează numere de 64 de biți şi execută o operație de adunare într-un ciclu de ceas. Astfel, în procesorul gazdă 1.000 de adunări vor fi efectuate în 0,4 μs, în condițiile disponibilitații operandului într-un ciclu de ceas. Lărgimea de bandă procesor- memorie este de 6,4 GB/s, ceea ce conduce la un timp total de transfer, în ambele sensuri, dat mai jos:

$T_{P-M} = 2 \times N_{r.}Octeți/Lărgimea\_de\_bandă_{P-M} = 2 \times (8.000/6,4) \times 1024 \times 1024 \times 1024 \approx 2,40\ \mu s$

Arhitectura de interconectare	Timpul total de transfer şi calcul
Interfața PCI	114,80 μs
Interfața PCI-X	14,40 μs
Interfața HT	4,70 μs
Magistrala Procesor-Memorie + Calcul	2,44 μs

Fig. 4.1. Timpii de transfer şi calcul pentru diverse arhitecturi de interconectare.

Din acest exemplu, simplificat, se poate constata că soluția bazată pe procesorul convențional, din sistemul gazdă, în ipoteza că timpul de calcul pe placheta ataşată este neglijabil, va fi mai performantă.

Performanța poate fi puternic influențată de lungimea cuvintelor transferate şi prelucrate. Dacă în locul cuvintelor în virgula mobilă, pe 64 de biți, vor fi utilizate cuvinte întregi pe 8 biți, timpii de transfer între sistemul gazdă şi placheta ataşată se vor reduce de 8 ori:

T_{PCI} = 14,35 µs

T_{PCI-X} = 1,80 µs

T_{HT} = 0,59 µs

Astfel, soluțiile PCI-X şi HT devin mai performante decât soluția bazată pe sistemul convențional. În practică performanța soluției "Magistrală Procesor-Memorie" este dată de modul în care microprocesorul manipulează datele pe 8 biți. De exemplu, datele pot fi împachetate câte 8 pe un cuvânt de 64 de biți, ceea ce presupune efectuarea de deplasări şi mascări, pentru accesul la nivel de cuvânt de 8 biți. Toate acestea vor contribui la reducerea performanței soluției "Magistrala Procesor-Memorie", favorizând utilizarea plachetei ataşate, echipate cu circuit FPGA.

4.3. Cerințele aplicației.

Avantajele utilizării FPGA-urilor ataşate sunt evidente în cazul algoritmilor intensiv-computaționali, când ponderea timpilor de transfer este mică în comparație cu timpul de calcul. În scopul ilustrării observației de mai sus se vor compara performanțele celor trei tipuri de interfețe: PCI, PCI-X şi HT, în condițiile implementării pe placheta ataşată, care conține un circuit FPGA, a doua subrutine DAXPY şi DGEMM, din BLAS (Basic Linear Algebra Subroutines). Algoritmii care stau la baza acestor subrutine sunt descrişi mai jos.

DAXPY este o operație, pe 64 de biți din BLAS, în care un vector **X** este înmulțit cu o constantă **a** şi adunat cu alt vector **Y**, având forma de mai jos, unde **n** este numărul de componente ale vectorilor:

```
for i = 1 to n
    y(i)=a*x(i)+y(i)
end
```

Pentru fiecare rezultat individual se vor efectua: doua transferuri (**x(i), y(i)),** doua operații aritmetice (***,** +) şi un transfer al rezultatului (**y(i)**). Dacă se ia pentru **n** valoarea 1000, se pot utiliza argumentația şi rezultatele calculelor de la paragraful 4.2, pentru tansferurile operanzilor şi ale rezultatului se obțin, în cazul arhitecturilor PCI, PCI-X, HT, următoarele valori:

T_{PCI} = 172,2 μs

$T_{PCI\text{-}X}$ = 21,6 μs

T_{HT} = 9,2 μs

S-au neglijat întârzierile asociate operațiilor aritmetice în FPGA, deoarece au loc în circuite combinaționale şi se suprapun cu operațiile de transfer. Pe de altă parte, dacă microprocesorul este organizat în bandă de asamblare, se poate considera că operațiile de înmulțire şi adunare au loc într-un ciclu de ceas. Timpul total al operațiilor aritmetice fiind de 0,4 μs. La o lărgime de bandă a magistralei Procesor-Memorie de 6,4 GB/s se obține pentru transferul a 1000 de valori pe 64 de biți un timp de 1,2 μs. Astfel, timpul total de transfer şi calcul pentru sistemul gazdă va fi de: 3 x 1,2 μs + 0,4 μs = 4 μs. Rezultatele obținute sunt prezentate mai jos:

Arhitectura de interconectare	Timpul total de transfer şi calcul
Interfața PCI	172,2 μs
Interfața PCI-X	21,6 μs
Interfața HT	9,2 μs
Magistrala Procesor-Memorie + Calcul	4,0 μs

Fig. 4.2. Timpii de transfer si calcul al DAXPY-BLAS**,** pentru diverse arhitecturi de interconectare.

Soluția care presupune efectuarea operațiilor la nivelul sistemului gazdă este mai eficientă în condițiile unor algoritmi mai puțin intensivi-computaționali şi cu date mai puține.

Pentru a exemplifica influența algoritmilor intensivi-computaționali asupra performanței, se va considera subrutina **DGEMM-BLAS**, care asigură înmulțirea a două matrici **n** x **n** cu operanzi pe 64 de biți. Se observă că algoritmul posedă doua bucle imbricate:

```
for i = 1 to n
  for j = 1 to n
    c(i,j) = 0
    for k = 1 to n
      c(i,j) = c(i,j) + a(i,k) * b(k,j)
    end
  end
end
```

Matricile care se înmulțesc sunt **A** şi **B,** cu dimensiunile **n** x **n**; matricea care rezulta **C** are aceeaşi dimensiune **n** x **n.** Astfel, pentru a se citi operanzii **A** şi **B** sunt necesare $\mathbf{2.n^2}$ transferuri spre circuitul FPGA, iar pentru stocarea rezultatului **C** în memoria sistemului gazda, alte $\mathbf{n^2}$ transferuri. În ceea ce priveşte calculul propriu-zis, acesta constă într-o înmulțire şi o adunare, în bucla internă triplu imbricată, ceea ce conduce la $\mathbf{2.n^3}$ operații. Aceasta subrutina este evident computațional-intensivă.

Evaluarea timpilor de transfer pentru încărcarea componentelor matricilor A şi B (2×8000^2 octeți), cât şi pentru stocarea componentelor matricei produs C (8000^2 octeți), utilâzand interfețele PCI, PCI-X şi HT, în situația **n** = 1000, conduce la următoarele valori: 1080 ms, 115 ms, 75.2 ms.

În implementarea pe sistemul gazdă se obțin următoarele rezultate: timpul de transfer al datelor inițiale din memorie la procesor şi al rezultatului de la procesor la memorie: 28 ms şi 800 ms. Admițând că se neglijează timpul de calcul în FPGA, soluția bazată pe sistemul gazdă este comparabilă cu cea în care se foloseşte interfața PCI, dupa cum reiese din tabelul de mai jos, şi devine net dezavantajoasă în cazul prelucrării matricilor cu **n** > 1000.

Arhitectura de interconectare	Timpul total de transfer şi calcul
Interfața PCI	1.377,6 ms
Interfața PCI-X	172,8 ms
Interfața HT	112,8 ms
Magistrala Procesor-Memorie + Calcul	828 ms

Fig. 4.3. Timpii de transfer si calcul al DGEMM –BLAS**,** pentru diverse arhitecturi de interconectare.

Până acum s-a neglijat timpul de calcul în circuitele FPGA, ceea ce nu corespunde realității, cu toate că acest timp de calcul este sensibil mai mic decât timpul de calcul al procesorului.

4.4. Calculul vitezei de operare a circuitelor FPGA.

Numărul de operații realizate de către un circuit FPGA, într-un ciclu de ceas, depinde de lărgimea de bandă a memoriei din structura acestuia. Circuitul Xilinx Virtex-4 LX200 posedă 336 Blocuri RAM (BRAM), cu porturi duale, la o capacitate de 1024 cuvinte de 18 biți/BRAM.

Pentru a obține un operand de 64 de biți trebuie efectuate două citiri, din două blocuri BRAM, adică 4 cuvinte de 18 biți. Astfel, lărgimea de bandă BRAM, pentru cuvinte de 64 de biți, este de 336/2 = 168 cuvinte într-un ciclu de ceas. Fiecare parcurgere a buclei interioare a subrutinei

DGEMM necesită citirea a două cuvinte **a(i,k)**, **b(k,j)** şi stocarea unui rezultat **c(i,j)**, ceea ce conduce la execuția a 168/3 = 56 de bucle interioare, într-un ciclu de ceas.

Circuitul LX200 dispune de resure hardware capabile sa implementeze 128 de sumatoare sau 70 multiplicatoare pe 64 de biți, cu condiția să nu mai fie necesară realizarea altor circuite logice în FPGA-ul dat. Astfel, se poate aprecia că circuitul LX200 dispune de 30 sumatoare şi de 30 multiplicatoare pe 64 biți. Subrutina DAGEMM necesită 2×1.000^3 operații. În situația în care se efectuează 56 de operații într-un ciclu de ceas, la o frecvență a ceasului de 200MHz, pentru circuitul dat, vor fi necesare $2 \times 1.000^3/56 \times 200 \times 10^6 = 178,6$ ms, în vederea efectuării tuturor calculelor. Dacă la această valoare se adaugă durata transferului HT, de 112,8 ms se obține un timp total (*calcul + transfer*) de 291,3 ms, sensibil mai mic decât cel înregistrat în cazul soluției bazate pe sistemul gazdă, de 828 ms.

Mai jos se dau timpii: *transfer + calcul* şi reducera timpului total față de soluția *gazdă* ai execuției subrutinei DGEMM -BLAS**,** pentru *diverse arhitecturi de interconectare "placheta* FPGA-*calculator gazdă*"..

Examinarea tabelului de mai sus evidentiază faptul că numai arhitectura bazată pe interfața PCI este inferioară ca performanța soluției convenționale, în timp ce soluțiile PCI-X şi HT sunt de circa 2,4 şi, respectiv, 3 ori mai rapide.

În măsura în care intensitatea calculelor creşte, reducerile timpilor totali vor fi mult mai accentuate. Producătorii de circuite FPGA sunt într-o continuă competiție pentru a mări cantitatea de resurse hardware pe circuit, cât şi pentru a mări frecvența de lucru a ceasului. Toate acestea contribuie la alegerea unor soluții bazate pe circuite FPGA, pentru implementarea algoritmilor intensivi-computaționali. Spre exemplu, algoritmii caracterizați prin manipularea de octeți nealiniați impun ca procesorul să efectueze operații de deplasare şi mascare, pentru extragerea grupului dat de biți, la nivelul fiecărui cuvânt.

Arhitectura de interconectare	Timpul total (transfer + calcul)	Reducerea timpului total față de soluția gazdă
Interfața PCI + Calcul	1.556,2 ms	0.532
Interfața PCI-X + Calcul	351,4 ms	2.357
Interfața HT + Calcul	291,3 ms	2.842
Magistrala Procesor-Memorie + Calcul	828 ms	1.000

Fig. 4.4. Timpii transfer + calcul şi reducera timpului total față de soluția gazdă ai DGEMM -BLAS**,** pentru diverse arhitecturi de interconectare.

În cazul soluției FPGA operațiile se pot efectua în paralel, pe grupuri. În lucrarea [45], care se referă la implementarea FPGA a algoritmului Smith-Waterman, folosit pentru decodificarea codului genetic, se arată o creştere de viteză a calculelor, de 64 de ori, față de soluția convențională. Datele despre amino-acizi, reprezentate pe 5 biți, au fost prelucrate cu 24 de unități pentru întregi.

În scopul utilizării avantajoase a FPGA-urilor, uneori, se apelează la reducerea preciziei, pentru a scădea cantitatea de date transferate şi a creşte numărul de operații efectuate într-un ciclu de ceas. Cantitatea de date transferate se va reduce la jumatate, numărul de unități funcționale şi de BRAM-uri implementate în FPGA se va dubla, în condițiile utilizării preciziei simple pe 32 de biți.

Lucrarea [46] prezintă o aplicație în care datele pe 32 de biți, în virgula mobilă, precizie simplă, au fost convertite în virgulă fixă pe 16 biți, pentru a se calcula Transformata Fourier Rapidă. Rezultatele reprezentate ca întregi pe 32 de biți au fost trunchiate pentru a se obține mantisele de 23 de biți, ale reprezentărilor în virgula mobilă, precizie simplă. În urma comparării acestor rezultate cu cele obținute prin soluția clasică, folosind aritmetica în virgulă mobilă, precizie simplă, s-a constatat aceeaşi precizie.

4.4. Considerații de ordin practic.

Utilizarea circuitelor FPGA pe plachete ataşate, pentru accelerarea unor algoritmi intensivi-computaționali, presupune un proces de evaluare, care trebuie sa aibă în vedere aspecte ca:

- lărgimea benzii căii dintre memoria microprocesorului şi FPGA;
- dimensiunea şi viteza de lucru ale FPGA;
- cantitatea de memorie asociată circuitului FPGA;
- cerințele algoritmului care va fi mapat în circuitul FPGA.

După ce s-a ajuns la concluzia că o soluție bazată pe FPGA conduce la o accelerare a calculului, trebuie să se aibe în vedere alți factori, care sunt de natură să creeze dificultăți în procesul de implementare optimă, în vederea obținerii performanței aşteptate:

- selectarea uneltelor de proiectare potrivite;
- lipsa de experiență a dezvoltatorului, care nu poate scrie un cod eficient.

Dezvoltatorul se poate confrunta cu o multitudine de platforme de proiectare, elaborate de numeroase companii [47]. La nivelul cel mai de jos, dezvoltatorul poate elabora proiectul într-un limbaj de descriere hardware (HDL), cum ar fi VHDL sau Verilog, care oferă un control direct asupra proiectului, pentru a obține cea mai bună performanță. Aceşti dezvoltatori sunt experți în hardware şi nu în programarea aplicațiilor din domeniul CIP. La capătul superior al ierarhiei de

niveluri de descriere sa află limbajele de nivel înalt, care estompează complexitatea HDL, uşurând programarea FPGA, pentru programatorii din domeniul CIP. Aceste facilități pot anula, în unele cazuri, avantajele soluțiilor bazate pe FPGA.

Circuitele configurabile sunt în continuă dezvoltare şi inovare, ceea ce necesită, din partea proiectanților urmărirea cu atenție a apariției de noi circuite, cât şi a platformelor de dezvoltare.