Arhitecturi de Microprocesoare si Sisteme de Calcul

Obiective

Doua componente esentiale ale structurii unui sistem de calcul sunt reprezentate de catre procesor si placa de baza. Astfel, in acest laborator vom vorbi despre:

- Diferitele abordari ale structurii unui procesor
- Ce probleme si ce imbunatatire ale performantei au motivat aparitia acestor structuri
- Arhitecturi CISC/RISC bazate pe procesoare Intel/AMD - inclusiv modele standard de placi de baza
- Comparatii intre abordarile curente utilizate in implementarea procesoarelor moderne si a placilor de baza
- Solutii de interconectare a procesoarelor AMD si Intel - Hypertransport si QuickPath Interconnect

Instruction level parallelism

Masina Turing executa cate o instructiune la un moment dat. Cand programatorul scrie un program, ii este foarte simplu sa considere ca programul sau va fi executat in acest mod. Pe de alta parte, o masina care executa cate o instructiune este mai lenta decat una care executa mai multe instructiuni in paralel. Pentru a cumula avantajele celor doua abordari, ar trebui ca programatorul sa poata inca scrie cod ca pentru o masina seriala, iar procesorul sa execute acest cod cu un nivel de paralelism cat mai ridicat. Cum este posibil asa ceva? Cineva trebuie sa faca trecerea dintre perspectiva seriala a programatorului si perspectiva paralela pe care ne-ar placea sa o aiba procesorul. Acest cineva poate fi ori un compilator, ori un hardware specializat aflat in structura procesorului.

Aceste considerente influenteaza structura procesorului si au dus la aparitia conceptului de Instruction Level Paralelism (ILP). Astfel, procesorul ia instructiuni dintr-un singur flux de control, le decodifica si executa in paralel. De exemplu, un procesor cu ILP poate sa scrie simultan rezultatele a doua instructiuni in registre, sa faca operatii aritmetice pentru alte trei, sa citeasca operanzii pentru alte doua, sa decodifice alte patru si sa ia (fetch) din fluxul de intrare inca patru instructiuni.

Cateva implementari de ILP includ:

- Pipeline: In acelasi ciclu de ceas, procesorul scrie rezultatul unei instructiuni in registre, executa operatia aritmetica a instructiunii urmatoare, si citeste operanzii instructiunii de dupa instructiunea urmatoare (la doua instructiuni dupa prima) .
- VLIW (Very Long Instruction Word): Lanseaza mai multe instructiuni in acelasi ciclu de ceas. Compilatorul trebuie sa se asigure ca nu exista dependente de date intre acestea. La procesoarele superscalare, numarul de unitati de executie este transparent pentru setul de instructiuni. VLIW este insa constient de numarul de unitati de executie.
- Superscalar: Lanseaza mai multe instructiuni in acelasi ciclu de ceas. Dependenta de date este insa verificata de hardware aditional. Daca nu pot fi lansate in paralel, se va executa cate o instructiune secvential (neavand suport din partea compilatorului, exista si aceasta posibilitate).
- Planificare-dinamica: Instructiunile sunt reordonate in timp ce sunt executate. In modul acesta, poate sa gaseasca usor instructiuni care nu au dependenta de date intre ele, pentru a fi executate simultan.

ILP-ul mareste asadar performanta procesorului. Dar de ce nu executam toate instructiunile deodata in paralel? Acesta ar fi de fapt modul cel mai rapid de a executa un program. Acest lucru nu se intampla deoarece ILP-ul are si anumite limitari, respectiv:

- Dependenta de date: Daca rezultatul instructiunii A este operand pentru instructiunea B, atunci evident B nu poate fi executata inainte ca A sa se fi terminat.
- Numar limitat de unitati functionale: Daca avem 5 sumatoare in procesor, nu putem executa mai mult de 5 sume simultan.
- Numar limitat de instructiuni lansate: Daca unitatea de lansare de instructiuni poate lansa maxim 5 instructiuni simultan, un program cu 500 de instructiuni va avea nevoie de 100 de operatii ale acestei unitati.
- Numar limitat de registre.

Mai multe detalii despre Explicitly Parallel Instruction Computing pot fi gasite aici EPIC.

Comparatie CISC vs. RISC

Cand a aparut CISC, ideea era sa se aduca in hardware stilul de programare specific unui limbaj care sa se aproprie (pe cat e posibil la nivelul hardware) de un limbaj cat mai inalt. Astfel, instructiunile complexe au acelasi efect ca micile secvente de instructiuni simple. Implemenarea acestor instructiuni complexe in hardware insemana insa:

- Hardware complex
- Locul ocupat de hardul pentru instructiunile complexe ar fi putut fi utilizat pentru a avea mai multe unitati de executie (si deci grad de paralelism mai mare)
- Secvente de microcod, care sunt lente comparativ cu restul procesorului

Datorita setului redus de instructiuni de asamblare, compilatoarele optimizate pentru RISC sunt capabile sa organizeze mai eficient fluxul de instructiuni de asamblare. Pe de alta parte insa, compilatoarele optimizate pentru RISC necesita mai mult timp de compilare

decat cele pentru CISC. Aceasta deoarece trebuie sa se ocupe si de managementul benzii de asamblare, anticiparea ramificatiilor (branch prediction) sau reorganizarea codului.

Ca principiu, o arhitectura RISC are mai multe registre generale, in timp ce CISC are mai multe registre speciale. Practic toate procesoarele moderne imprumuta atat caracteristici CISC cat si RISC.

Exista trei tipuri de categorii de instructiuni CISC, si anume:

- Aritmetico-logice
- De control secvential
- De acces la memorie

Formatul instructiunilor RISC are o lungime fixa, cu lungimea unei instructiuni in general egala cu lungimea cuvantului de memorie; in cazul CISC, lungimea unei instructiuni variaza in functie de formatul instructiunii. RISC are un numar mic de moduri de adresare, spre deosebire de CISC, care are un numar mare de moduri de adresare (utilizate mai rar).

Setul de instructiuni RISC este orientat pe registre (peste 32 de registre). Pentru ca accesul la memorie este mult mai lent decat lucrul cu registrele, RISC incurajeaza lucrul cu acestia. Face acest lucru prin cresterea numarului de registre si prin limitarea explicita a acceselor la memorie. In general instructiunile au 2 operanzi (registre) si un registru destinatie.

In cadrul arhitecturilor RISC exista o limitare explicita, si anume: singurul mod de acces la memorie este prin load si store. Aceasta se deosebeste fundamental de CISC care are instructiuni cu operanzi locatii de memorie. Totusi, desi RISC impune aceasta disciplina de lucru cu memoria, doar 20-25% din codul unui program e reprezentat de operatii de tip load sau store.

Arhitectura Intel

O schema clasica pentru un sistem CISC este prezentata in Figura 1. Aici se poate distinge usor in partea de sus Procesorul, legat de restul sistemului prin Front-Side-Bus (FSB) de 400/533/800MHz catre North Bridge (i.e. 82865PE MCH). Pe North Bridge se afla controllerul de memorie, si ca atare si memoriile sunt conectate direct aici prin canale intre 2.1GB/s si 3.2GB/s. De asemenea pe North Bridge se conecteaza atat placa grafica (AGP 8x/4x) cat si interfata de retea de mare viteza Gigabit Ethernet.

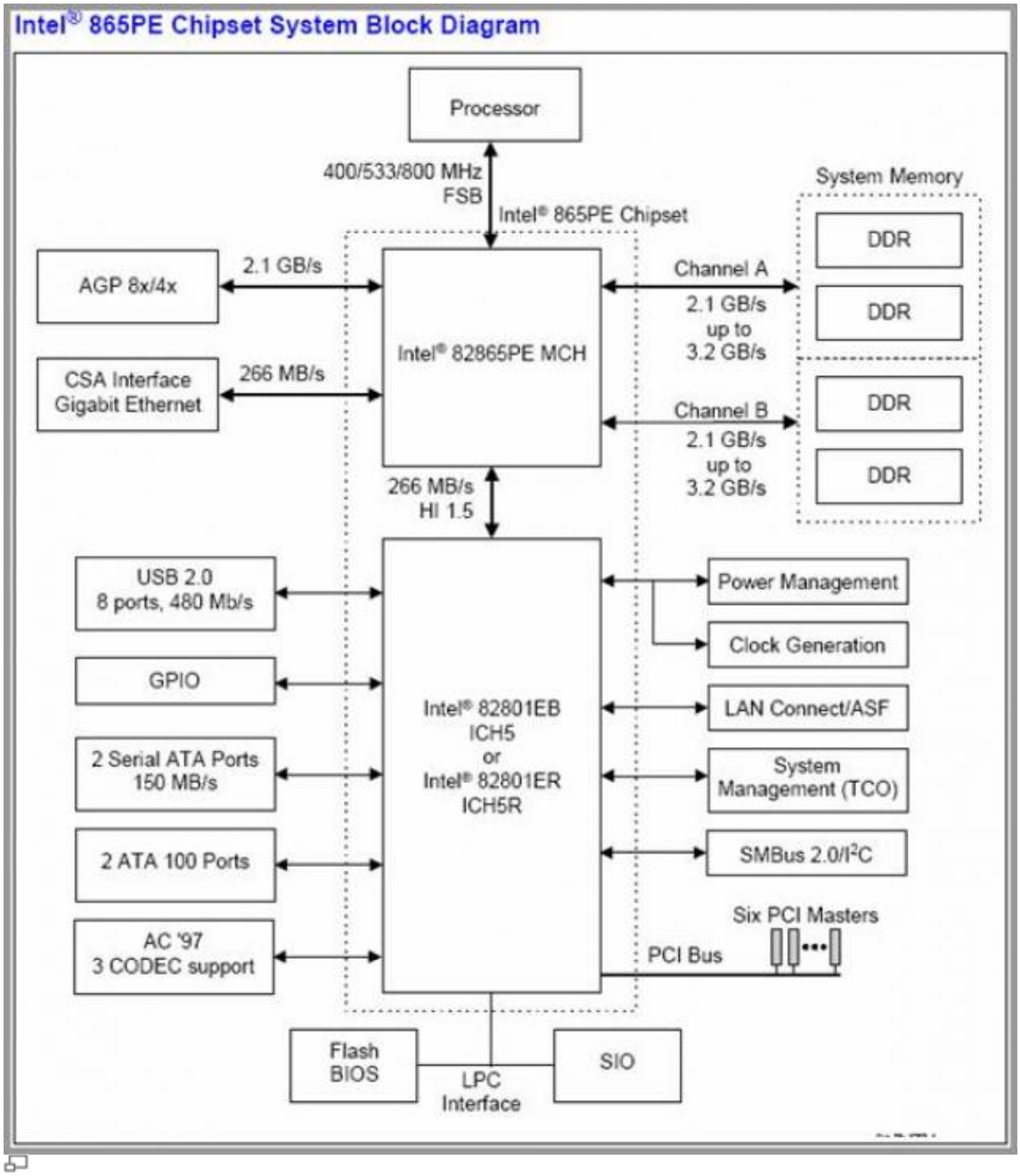

Figura 1. Schema bloc a Chipsetului Intel® 865PE

La randul sau North Bridge-ul este conectat printr-o legatura de 266MB/s catre South Bridge (i.e. 82801EB ICH5 / 82801ER ICH5R). Dupa cum se poate vedea, North Bridge-ul impreauna cu South Bridge-ul formeaza impreauna ceea ce se numeste Chipsetul Intel® 865PE. Urmarind in continuare schema din Figura 1, se observa ca pe South Bridge se conecteaza o multitudine de componente periferice cu o viteza si rata de transfer de date considerabil mai scazuta decat elementele conectate pe North Bridge, cum ar fi: AC97 (placa audio), porturi ATA si Serial ATA, porturi USB, sistem de management si de control al consumului, etc.

Din aceasta schema se poate usor deduce ca punctul vulnerabil al acestor sisteme il constituie integrarea controllerului de memorie pe North Bridge, si in special legaturile de marime limitata intre South si North Bridge, precum si intre North Bridge si procesor. In mod evident, dimensionarea acestora este un compromis de design al sistemelor Intel, menit sa deserveasca majoritatea sistemelor hardware bazate pe acest Chipset, si a aplicatiilor ce ruleaza pe ele.

Pentru a atinge insa performante mai inalte, mai ales in contextul aparitiei sistemelor multi-procesor si multi-core, este insa nevoie de imbunatatiri ale acestei abordari, cum se poate observa la Chipsetul Intel 7300 din Figura 2.

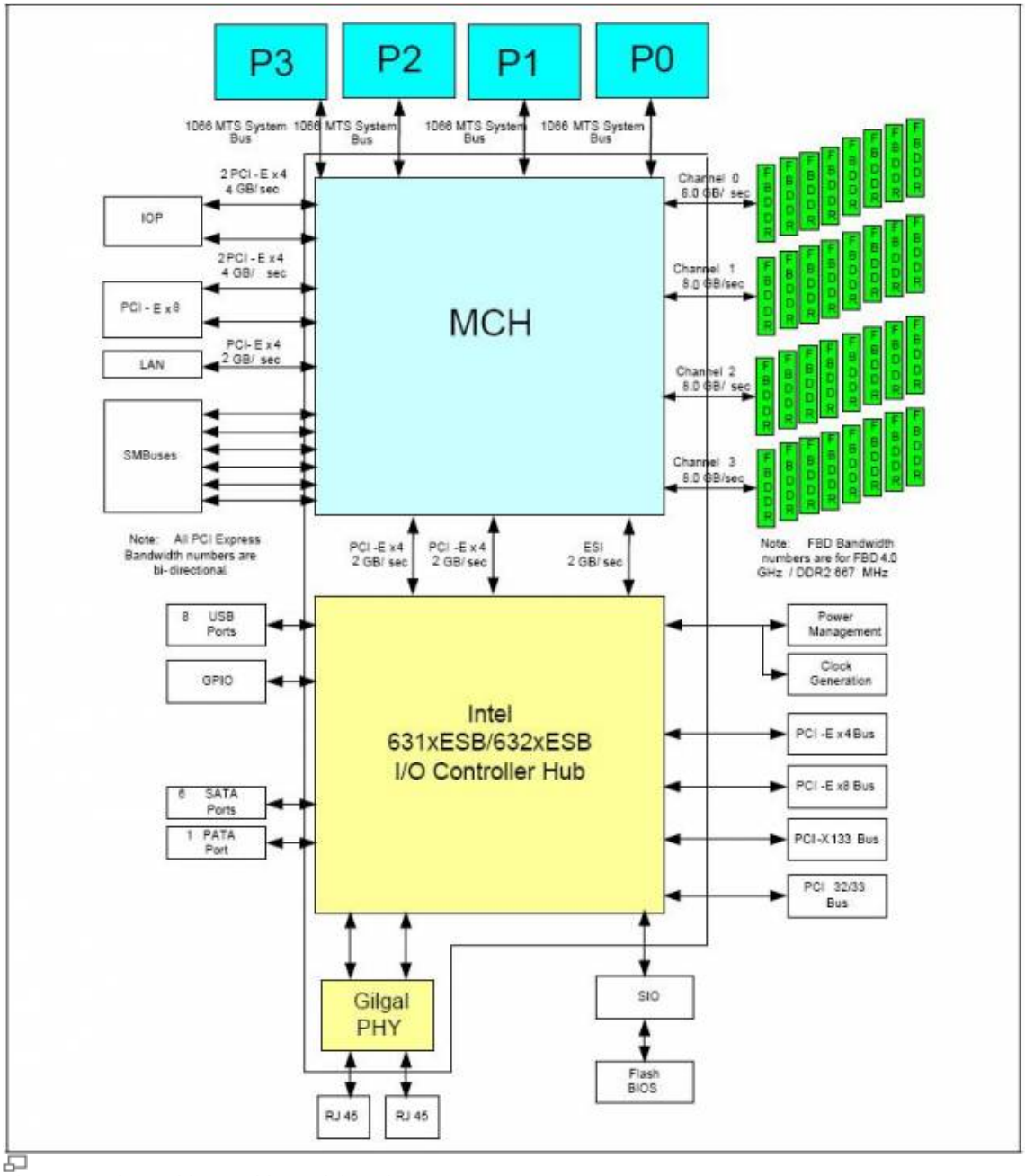

Figura 2. Schema bloc a Chipsetului Intel® 7300

Aici se poate observa conectarea a patru procesoare catre North Bridge (MCH) prin canale distincte FSB, de 1066MHz fiecare, menite sa asigure o alimentare eficienta cu date a acestora. Controllerul de memorie ramane pe North Bridge insa memoria se conecteaza prin patru canale de 8GB/s. Apar pe North Bridge conexiuni multiple PCI-Express (PCI-E), placa de retea devine doar “una dintre acestea”;, iar largimea de banda este crescuta pentru fiecare dintre acest componente fata de versiunile anterioare de Chipseturi.

Pentru simplitate, si conexiunea intre North Bridge si South Bridge (631xESB I/O Controller) este realizata prin conexiuni PCI Express 2x sau chiar 4x. Este sugestiv de asemenea faptul ca South Bridge-ul poarta acum numele de “I/O Controller” si se consfinteste astfel si prin nume direct rolul South Bridge-ului. Porturile ATA sunt inlocuite acum de SATA si PATA, apar din nou numeroase porturi PCI-E si PCI-X, USB, de gestiune a consumului, a biosului sau placi aditionale de retea pentru management.

Din prezentarea celor doua chipseturi 865PE si 7300 se poate vedea evolutia arhitecturala din jurul procesoarelor CISC de la Intel, cu o modularitate si o flexibilitate considerabil mai mare a chipsetului 7300, care ofera un potential de performanta mult crescut pentru sistemele ce il utilizeaza. Atentie, atat procesoarele Intel considerate pentru Chipset-urile prezentate, cat si cele AMD din sectiunile urmatoare, sunt din familii dedicate sistemelor de inalta performanta si cele mai puternice din clasa lor. Discutia prezentata este insa relevanta, la o scara corespunzator mai scazuta, si pentru celelalte sisteme si procesoare oferite de cele doua mari firme.

Anexa:
Magistrale uzuale si largimea lor de banda maxima

Bus	Max Bandwidth
PCI	132 MB/s
AGP 8X	2,100 MB/s
PCI Express 1x	250 [500]* MB/s
PCI Express 2x	500 [1000]* MB/s
PCI Express 4x	1000 [2000]* MB/s

Bus	Max Bandwidth
PCI Express 8x	2000 [4000]* MB/s
PCI Express 16x	4000 [8000]* MB/s
PCI Express 32x	8000 [16000]* MB/s
IDE (ATA 100)	100 MB/s
IDE (ATA 133)	133 MB/s
SATA	150 MB/s
Gigabit Ethernet	125 MB/s
IEEE 1394B [Firewire]	100 MB/s

PCI Express este o magistrala seriala (datele pot circula simultan in ambele directii). In tabelul de mai sus cele doua valori pentru largimea de banda corespund largimii de banda intr-o singura directie respectiv in ambele directii (combinat).

Arhitectura AMD Hammer

Din familia Hammer, sau AMD64, face parte cel mai putermic procesor de la AMD, si anume Opteron. Opteronul este echivalentul familiilor Intel Itanium si Intel Xeon, destinat serverelor si sitemelor de inalta performanta. Opteron este un procesor out-of-order si in interiorul unitatii de executie ordinea instructiunilor este schimbata, pentru a maximiza eficienta. Pentru utilizatorul extern insa, instructiunile par a se executa in aceeasi ordine in care au fost lansate. De asemenea, el este 3-way superscalar, adica poate decoda, executa si incheia trei instructiuni x86 la fiecare ciclu masina. Desi poate lucra in paralel la 3 instructiuni, aceasta nu insemna neaparat ca cele 3 instructiuni sunt procesate in intregime pe acea perioda de ceas. Opteronul a fost creat pentru a putea lucra in sisteme multiprocesor si fiind primul care a oferit o scalabilitate sporita, el a acaparat la vremea respectiva o portiune semnificativa din piata comerciala de servere.

Printre cele mai importante imbunatatiri arhitecturale cu care vine Hammer (generatia 8) fata de Athlon si AthlonXP (generatia 7) se numara: doua stagii in plus la pipeline, algoritmi imbunatatiti de predictie a ramificatiilor, suport pentru SSE2 (Streaming Multimedia Instructions), controller de memorie integrat in CPU si extensie completa pentru setul de instructiuni pe 64 biti pentru x86. Toate procesoarele Hammer au viteze de maxim 2.5GHz. Pe de alta parte, Intel s-a concentrat mai mult pe cresterea vitezei, fara a se preocupa excesiv de paralelism, acest trend a fost insa oprit din 2007-2008, cand si Intel, si AMD au trecut la producerea de sisteme multi-core. Acest lucru este important in lumea serverelor, unde disiparea caldurii este o problema principala. Intel insa a venit si cu o noua “arma” si anume tehnologia de integrare bazata pe dielectrici High-K, ce a dus la o scadere drastica a consumului, si a permis de asemenea performante considerabile la frecvente de executie scazute.

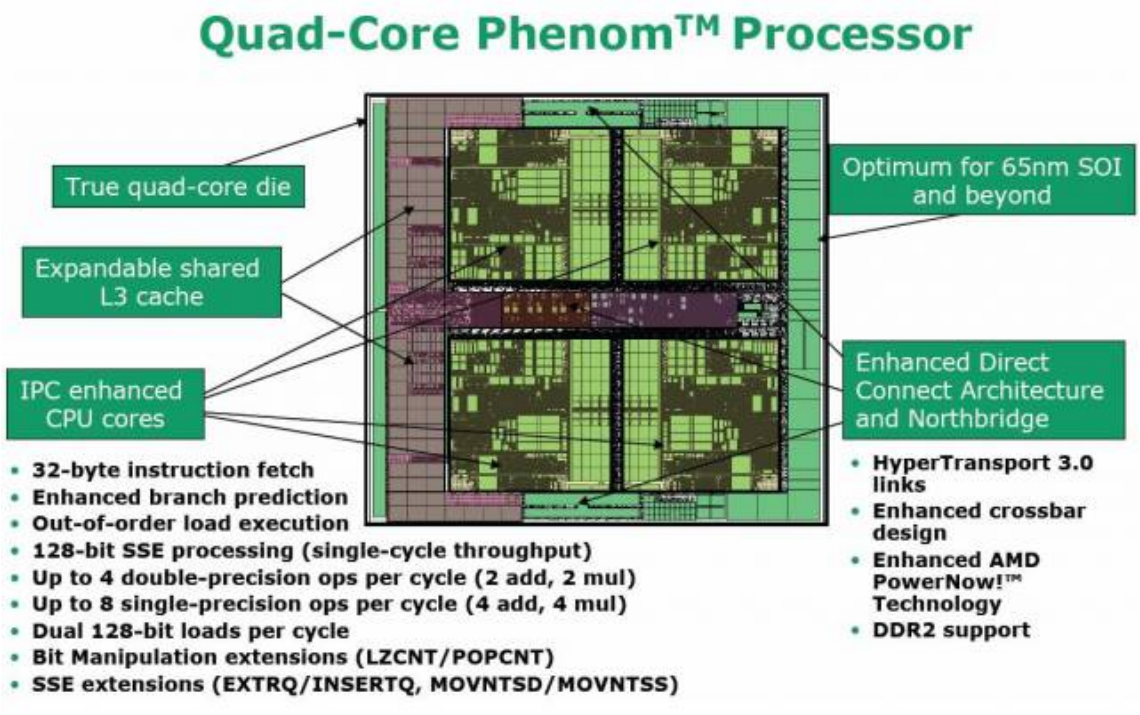

Figura 3. Arhitectura Procesorului Quad-Core Phenom

Dupa cum am spus, Hammer pastreaza compatibilitatea cu 16 si 32 biti (fata de Intel, care renunta complet la x86 si trece la IA-64). Pentru a putea face acest lucru, Hammer are doua moduri de operare: Legacy- si Long-Mode. Long Mode este subdivizat si el in Compatibility mode si 64bit Mode. Legacy mode este destinat exclusiv sistemelor de operare pe 16 si 32 biti. Compatibility mode este destinat sistemelor de operare pe 64 de biti, dar care ruleaza programe scrise pentru 32 biti. Astfel, desi programul in sine nu beneficiaza de facilitatile 64 biti, managamentul resurselor, facut de sistemul de operare pe 64 de biti beneficiaza de totate avantajele date de rularea pe 64 biti.

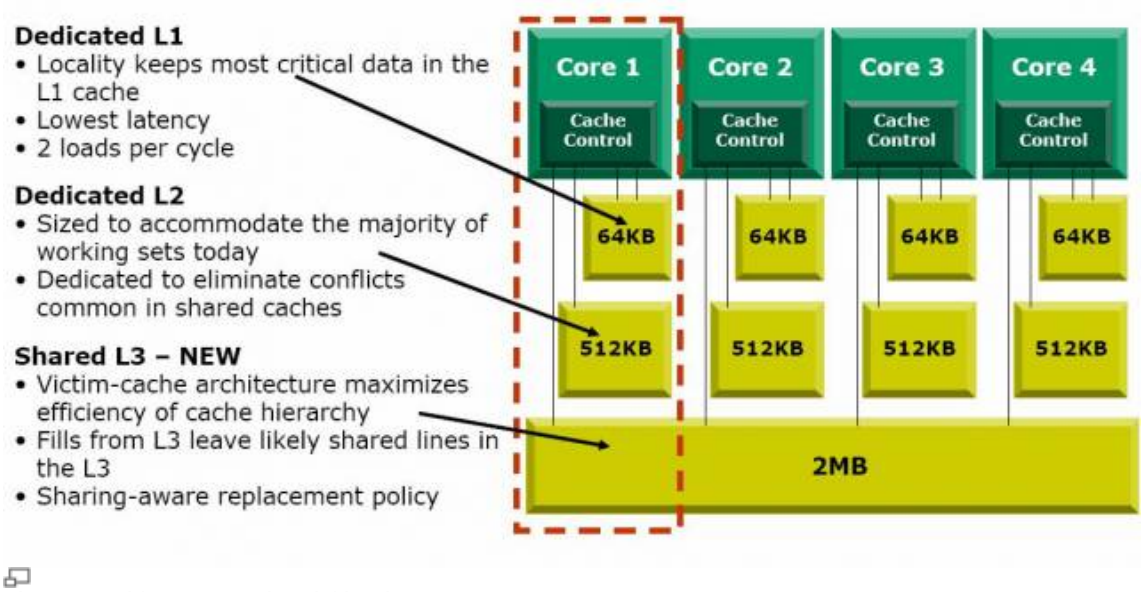

Figura 4. Arhitectura Cache-ului la Phenom

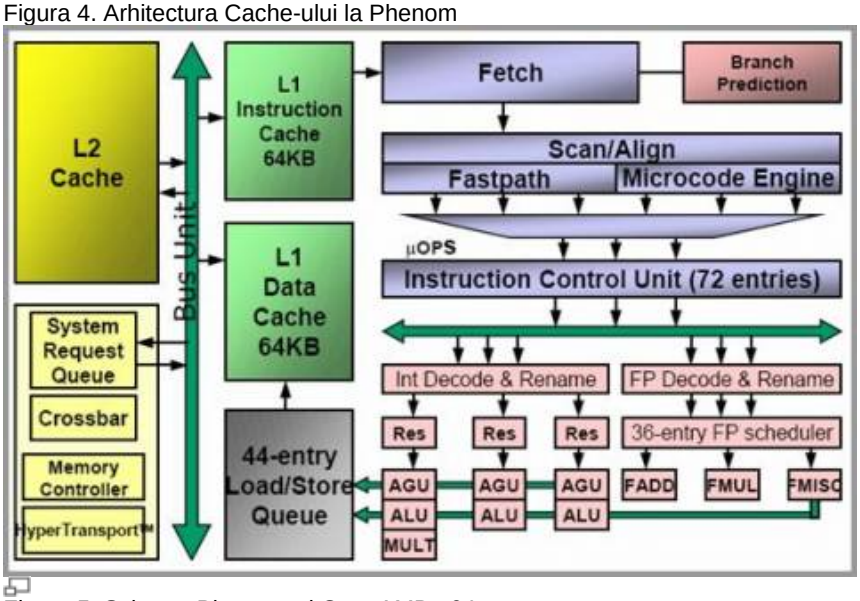

Figura 5. Schema Bloc a unui Core AMD x64

Controller-ul de memorie integrat in procesor la AMD

In general, un procesor foloseste doua cipuri de pe placa de baza pentru a accesa memoria si perifericele. Aceste sunt numite North Bridge si South Bridge, dupa cum au fost descrise in sectiunile anterioare. Se observa ca North Bridge-ul joaca un rol esential, el facand legatura cu memoria. De acea, de la generatia Hammer, AMD a integrat in cipul procesorului North Bridge-ul. In felul acesta se obtine o latenta de acces la memorie redusa cu cel putin 20%. Acest controller are o legatura de 128 biti cu memoria. In plus, el functioneaza dupa ceasul procesorului, acest lucru marind inca o data viteaza; permite de asemenea ca marirea frecventei de ceas a procesorului sa imbunatateasca si performantele controller-ului.

Controller-ul de memorie are grija si de coerenta cacheului. Cipul integrat se concentreaza acum doar pe comunicarea cu memoria. Alte functionalitati ale North Bridge-ului, cum erau de exemplu comunicarea cu AGP sau Placa de Retea, au fost mutate pe un cip extern.

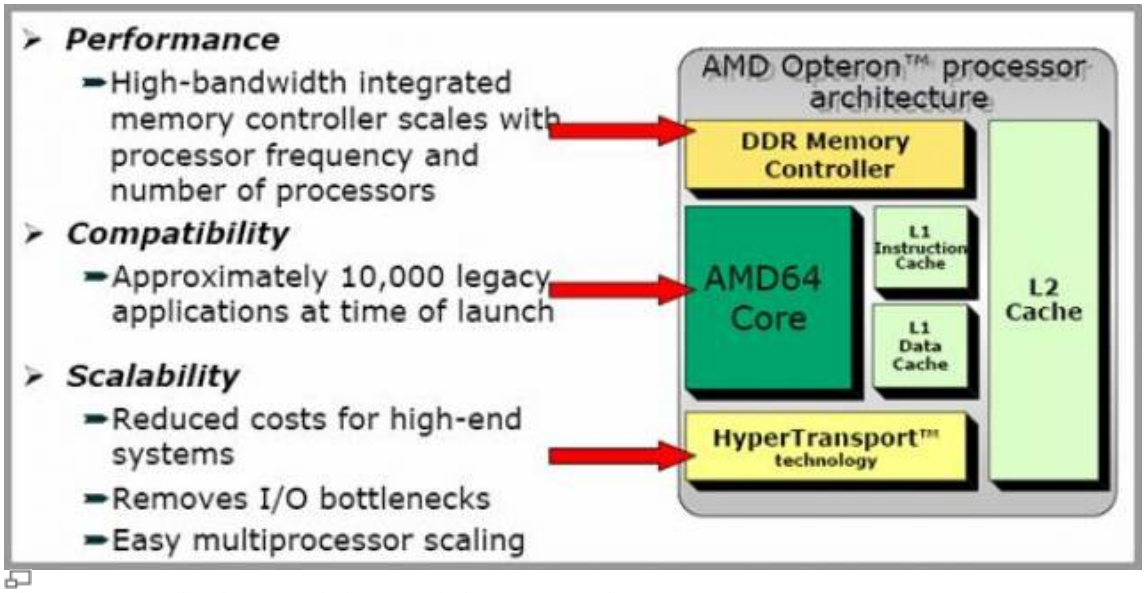

Figura 6. Controller de memorie integrat de la procesoarele Opteron

Hypertransport

Hypertransport este o tehnologie pentru I/O dezvoltata initial de AMD. Ea este o alternativa la sistemele actuale de bus. Foloseste legaturi duble, punct la punct, pentru a lega componentele intre ele. Este, in termeni de retele, echivalentul unei legaturi full-duplex punct la punct fata de o topologie buss.

O astfel de lagatura poate avea intre 2 si 64 biti, si poate opera la viteze de 400Mhz-2.6GHz. Datele sunt impachetate si trimise folosind un protocol care prevede trimiterea de pachete multiplu de 4 bytes, cu marimi intre 4 si 64 bytes.
Hypertransport e compatibil cu PCI, de aceea a fost usor de introdus. El poate lucra in doua moduri: coerent si non-coerent. Modul coerent e folosit pentru comunicatiile interprocesor. Modul non-coerent e optimizat pentru comunicatiile I/O.

Integrarea in Arhitectura Hammer

E folosit pentru a lega controller-ul de memorie integrat (fostul NorthBridge) de memorie. In mod similar este folosit in sistemele multiprocesor pentru comunicarea interprocesor, folosind modul coerent.

AMDOpteron are 3 legaturi Hypertransport. Seria 100 are 3 legaturi non-coerente, deoarece, fiind destianta monoprocesoareleor, nu are nevoie de comunicatie interprocesor. Seria 200 are 2 linii non-coerente si una coerenta, pentru unica legatura dintre cele doua procesoare (seria 200 e pentru dual-procesor). Si seria 300 are toate cele 3 legaturi coerente.

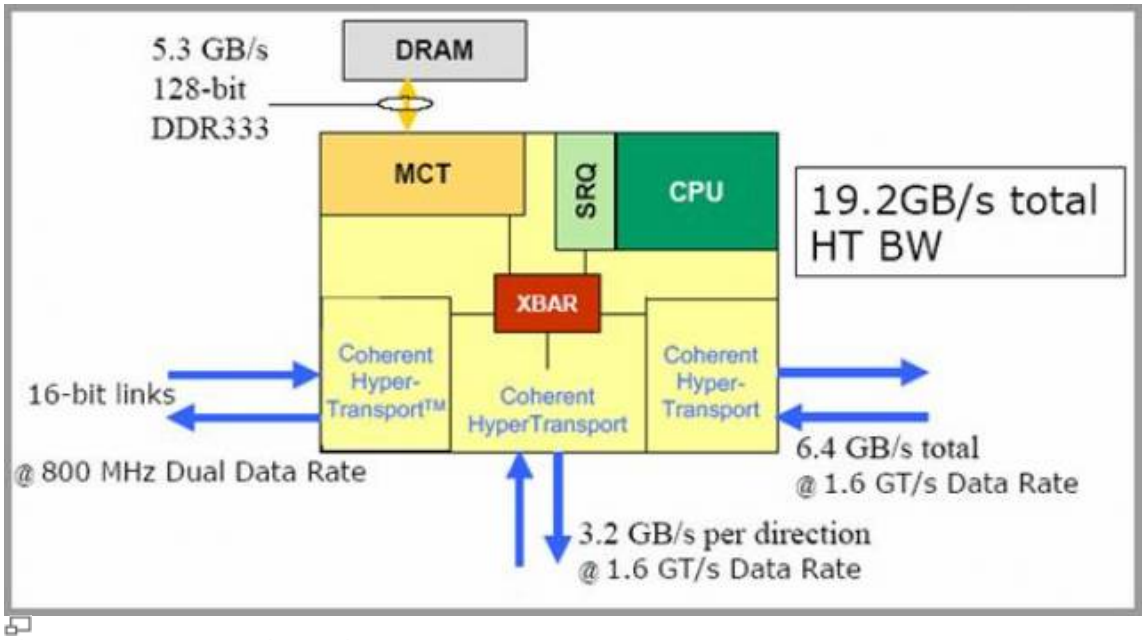

Figura. 7 Hypertransport intern si extern
Liniile sunt de 16 biti, bidirectionale cu frecvente intre 200 Mhz si 800Mhz, de aici rezultand o viteza de 6.4Gbytes/sec (3.2Gbytes/sec in fiecare directie). Cum Opteron are 3 astfel de lagaturi, poate comunica deci 19.2 Bytes/sec.

Cipuri ce conecteaza prin Hypertransport core-uri AMD

- AMD8151 Hypertransport AGP Tunnel: Controller grafic AGP3.0 . Este practic ce a mai ramas din NorthBridge dupa integrarea controller-ului
- AMD8131 Hypertransport PCI-X Tunnel: Are rol de buss cu PCI-X

- AMD8111 Hypertransport I/O Hub: Are functionalitate standard de SouthBridge, incluzand controller PCI, BIOS,USB,hard disk, retea si audio.

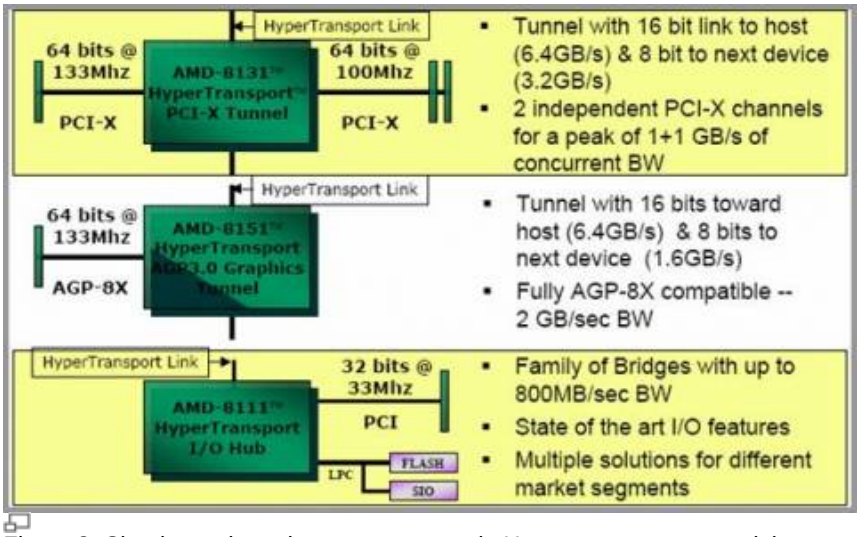

Figura 8. Cipuri ce asigura interconectarea prin Hypertransport a core-ului

Raspunsul Intel la Hypertransport - QuickPath Interconnect

In mod evident Intel nu putea ramane indiferent avantajelor oferite de sistemul de interconectare oferit de catre HyperTransport. Si astfel putem vedea in figura urmatoare cum a aparut QuickPath Interconnect:

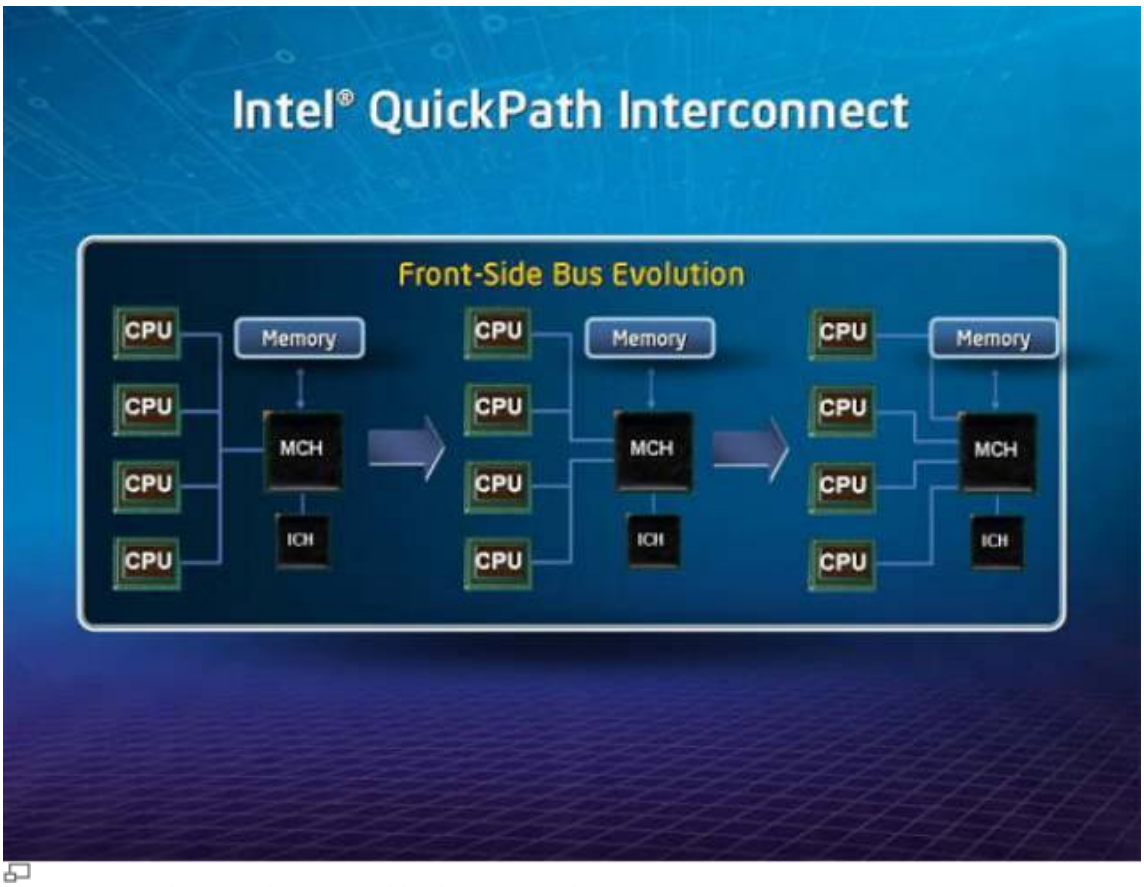

Figura 9. Evolutia Front-side-Bus-ului in sistemele Intel

Acest sistem de interconectare este practic identic functional cu HyperTransport. Intre timp, procesoarele Intel au incorporat si ele controller-ul de memorie, si astfel cei doi mari competitori pe piata procesoarelor de uz general sunt pregatiti in aceeasi masura pentru sisteme multi-core cu multe procesoare, memorie multa si aplicatii multi-pthreading pe scara larga:

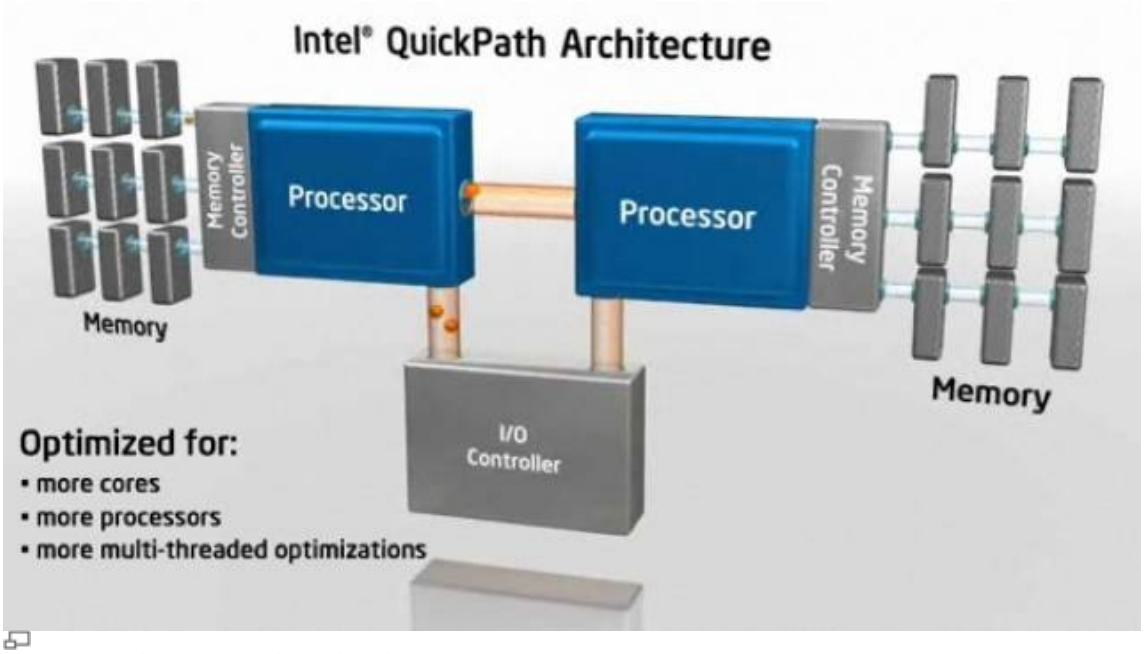

Figura 10. Arhitectura Intel QuickPath Interconnect

Sistemul QuickPath ofera o rata de transfer de pana la 25.6GB/s pentru fiecare port (pereche de linii). Legatura poate fi utilizata atat pentru interconectarea controller-ului I/O cu CPU-urile, sau a CPU-urilor intre ele. Astfel atat sisteme mono- cat si multi-procesor pot fi realizate cu usurinta cu acest tip de interconectare. Mai multe detalii pot fi vazute in figura urmatoare:

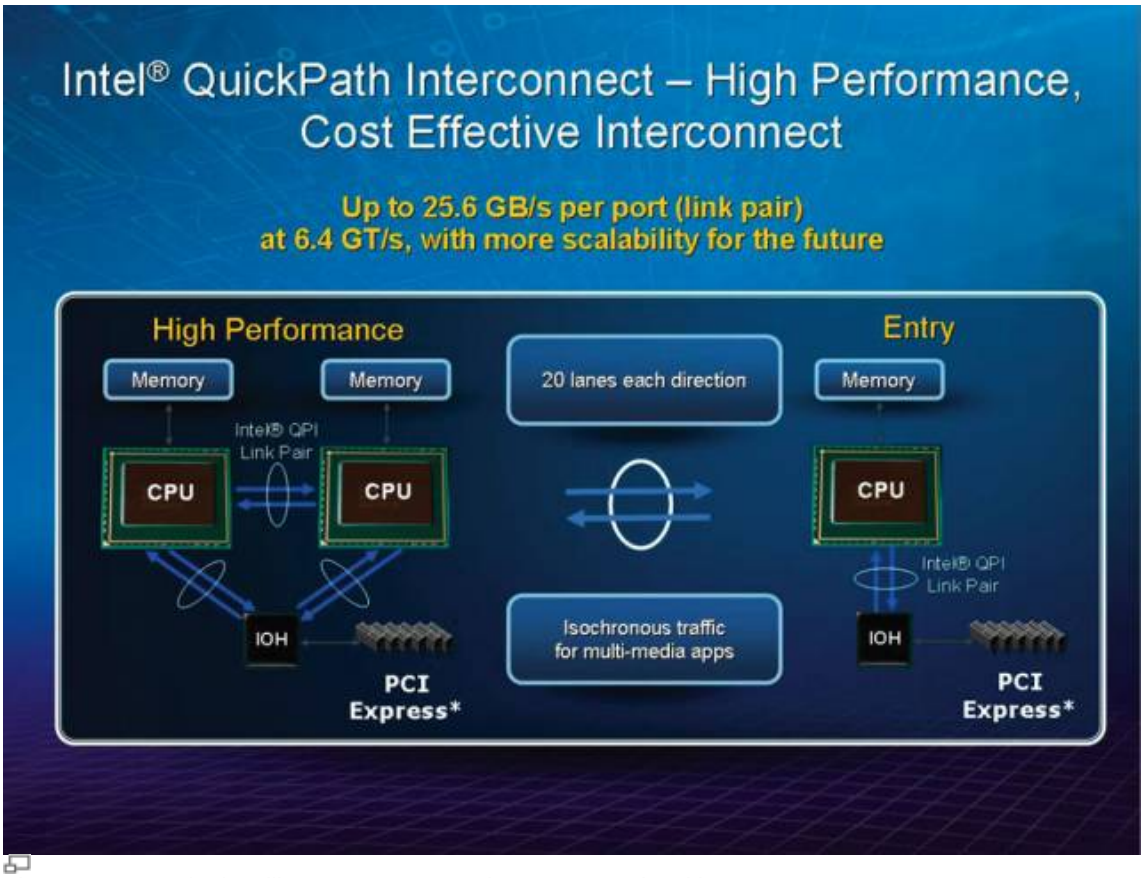

Figura 11. Exemple de utilizare a QPI pentru sisteme mono si multi-procesor

In mod similar cu sistemul Hypertransport, si interconectarea QuickPath poate fi utilizata pentru a lega sisteme HPC cu mai multe procesoare si siteme de intrare iesire, dupa cum se poate observa in urmatoarea figura:

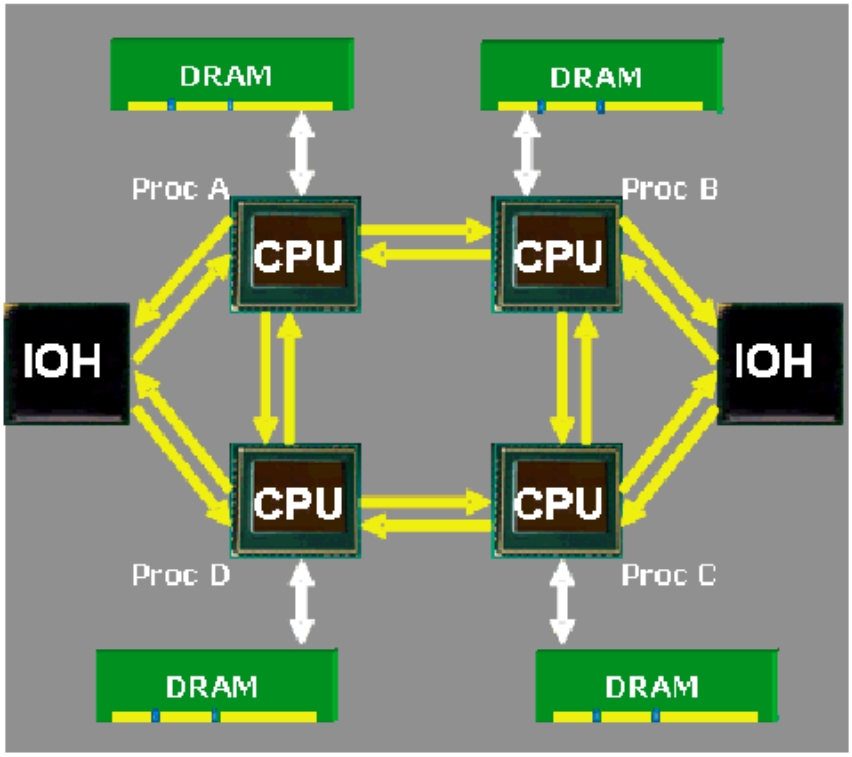

Figura 12. Conectarea Utilizand QPI a mai multor procesoare memorii si sisteme I/O

Sisteme Multiprocesor Intel si AMD

Familia Hammer a fost creata pentru a putea oferi un multiprocesor scalabil, eficient din punctul de vedere al pretului raportat la numarul de procesoare. AMD a mai avut o tentativa in trecut de a crea procesoare pentru sisteme multiprocesor, cu AthlonMP. Desi acesta nu a fost o reusita de piata, datorita lui AMD am putut studia problemele aparute in astfel de sisteme. La Athlon MP, memoria (care era partajata) era botleneck-ul principal. Fiind memorie partajata, toate procesoarele imparteau FSB (Front Side Bus). Cu alte cuvinte viteza cu care procesoarele puteau teoretic accesa memoria era mult mai mare decat viteza cu care putea fi aceasta accesata. Solutia de la Hammer ar fi fost sa ofere fiecarui CPU propria sa conexiune la North Bridge, dar acest lucru ar fi fost foarte scump. Solutia relativ ieftina si care nu are nici penalizari de performanta a fost includerea controller-ului de memorie in procesor, ceea ce s-a si facut. Astfel, fiecare procesor are propria sa legatura de 128 biti cu memoria, avand pana la 5.3 Gbytes/sec.

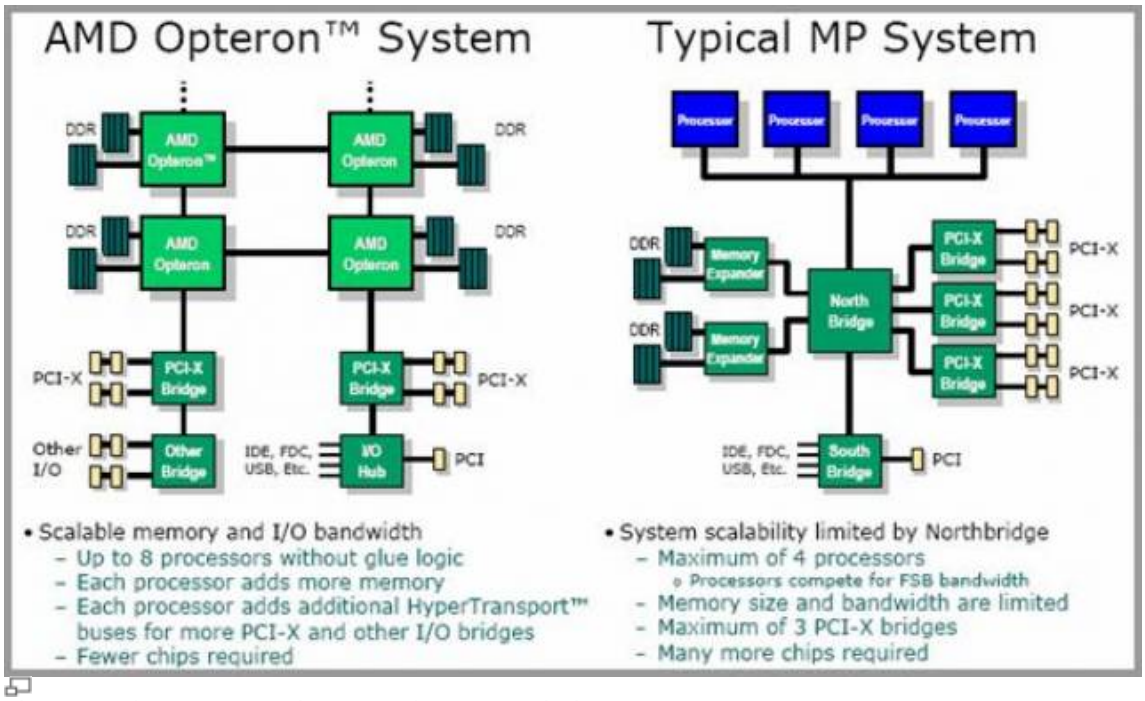

Figura 13. Sistem AMD vs sisteme Muti Procesor Clasice

In plus, datorita hypertransport, fiecare procesor poate accesa memoria celorlate procesoare la viteze de 3.2Gbytes/sec. Datorita acestui fapt, implementarea unui sistem dual-procesor e la fel de “usoara” ca a unuia cu 8 procesoare, deoarece partile componenete sunt scalabile prin utilizarea Hypertransport. AMD numeste acesta abordare “glueless multiprocessing”, deoarece procesoarele sunt legate slab prin Hypertransport. De fapt, e diferenta dintre o cuplare puternica gen circuit-switched versus o cuplare slaba, gen packet-switched, cum se intampla in cazul de fata. Figura de mai jos face o comparatie intre arhitecturile de la Intel (Xeon) si AMD (Athlon/Opteron).

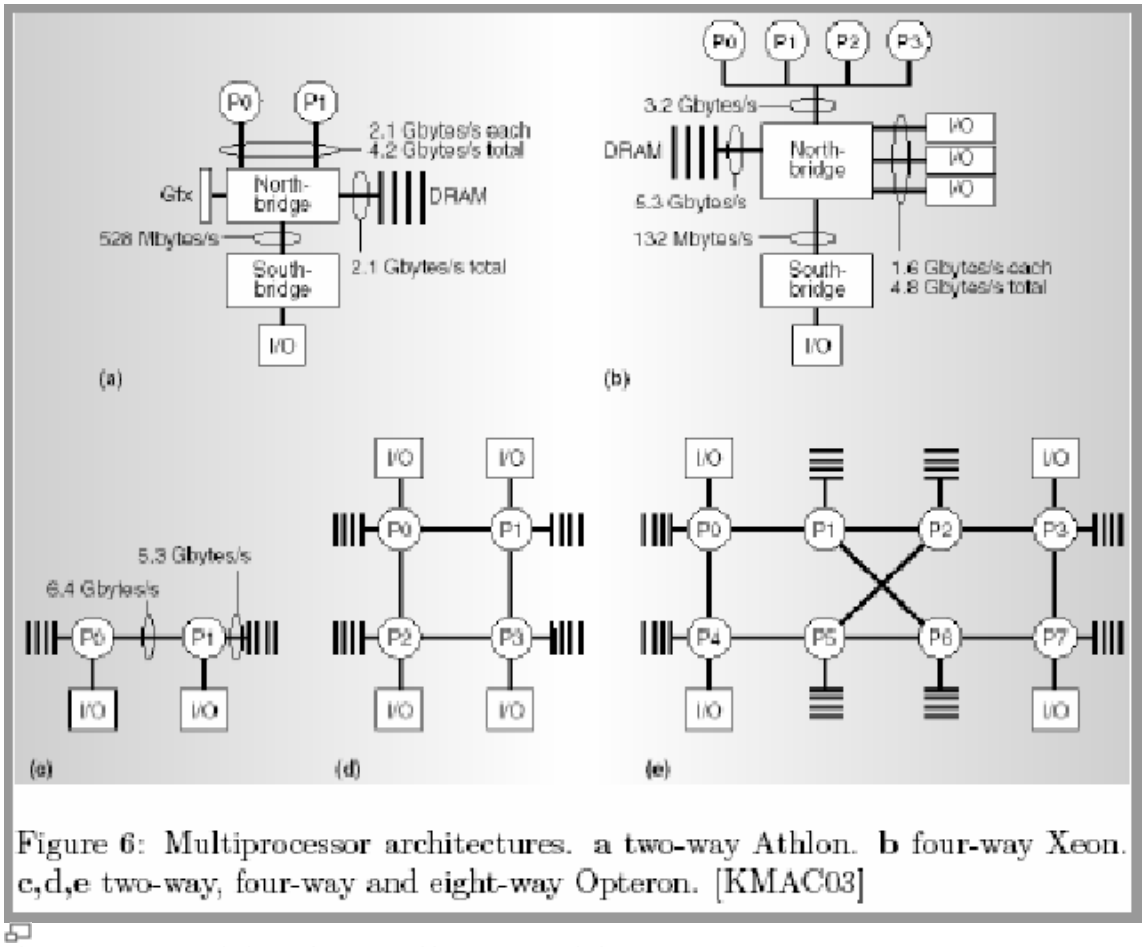

Figure 6: Multiprocessor architectures. **a** two-way Athlon. **b** four-way Xeon. **c,d,e** two-way, four-way and eight-way Opteron. [KMAC03]

Figura 14. Comparatie intre sisteme multiprocesor Intel si AMD

Se observa ca cele doua procesoare Athlon (a) impart acelasi controller de memorie. Desi aceasta abordare nu are repercusiuni asupra performantei, sistemul nu e scalabil, adica pentru un sistem cu 3 procesoare ar trebui creat un controller separat. Este practic un sistem puternic cuplat (circuit-switched). Cum am mai mentionat, sistemele puternic cuplate sunt greu de scalat. In sistemul (b) cu Intel Xeon, procesoarele impart FSB-ul care, dupa cum am arata mai sus, duce la un botleneck semnificativ. In final, la ©(d)(e) avem sisteme cu Hammer Opteron. Acestea , fiind slab cuplate prin HyperTransort sunt usor de scalat la 2, 4 sau 8 procesoare. In plus, fiecare are propria sa legatura la memorie, neaparand botleneckuri, ca in cazul Intel Xeon.

Comparatie intre servere AMD si Intel

Generatia Phenom de microprocesoare de la AMD este competitorul direct al Intel Xeon si Intel Itanium. Cele doua arhitecturi , Intel versus AMD sunt fundamental diferite, dar ofera performante comparabile, in functie de domeniul de aplicatie ales pentru comparatie.

O prima mare diferenta intre cele doua arhitecturi este modul in care cele doua abordeaza compatibilitatea cu 32 de biti. Astfel, de la Hammer incoace, AMD a ales sa extinda setul actual de instructiuni x86 pentru 32biti cu instructiuni pentru 64, in timp ce Intel a renuntat complet la setul x86, trecand in mod radical la IA64. Compatibilitatea la AMD este asigurata automat, noul set de instructiuni fiind doar o extensie e acelui vechi. La Intel, compatibilitatea cu 32 biti se face prin emularea vechilui set. Fiind o emulare, exista penalizari de performanta. Pe de alta parte insa, Intel a reusit sa scape in acest fel de complicatii de arhitectura inutile intr-o lume numai de 64 biti. Intr-o lume de 64 biti, AMD are legacy.

O alta diferenta este legatura dintre performanta, viteza ceasului si gradul de paralelism oferit. Astfel, la inceput Intel s-a concentrat pe marirea frecventei de ceas, si mai putin pe efectuarea de mai multe operatii in paralel. AMD ofera frecvente mai mici, dar a pus mult accent pe paralelism. Astfel performanta e oferita de AMD la frecvente mult mai mici decat Intelul. Un avantaj al acestui fapt este ca AMD elimina astfel problemele de disipare a caldurii. Aceste aspecte sunt esentiale mai ales in domeniul serverelor. In ultimii ani Intel merge pe aceeasi cale, si se axeaza in principal pe cresterea numarului de core-uri si nu a frecventei de procesare, cu aceleasi avantaje mentionate anterior.

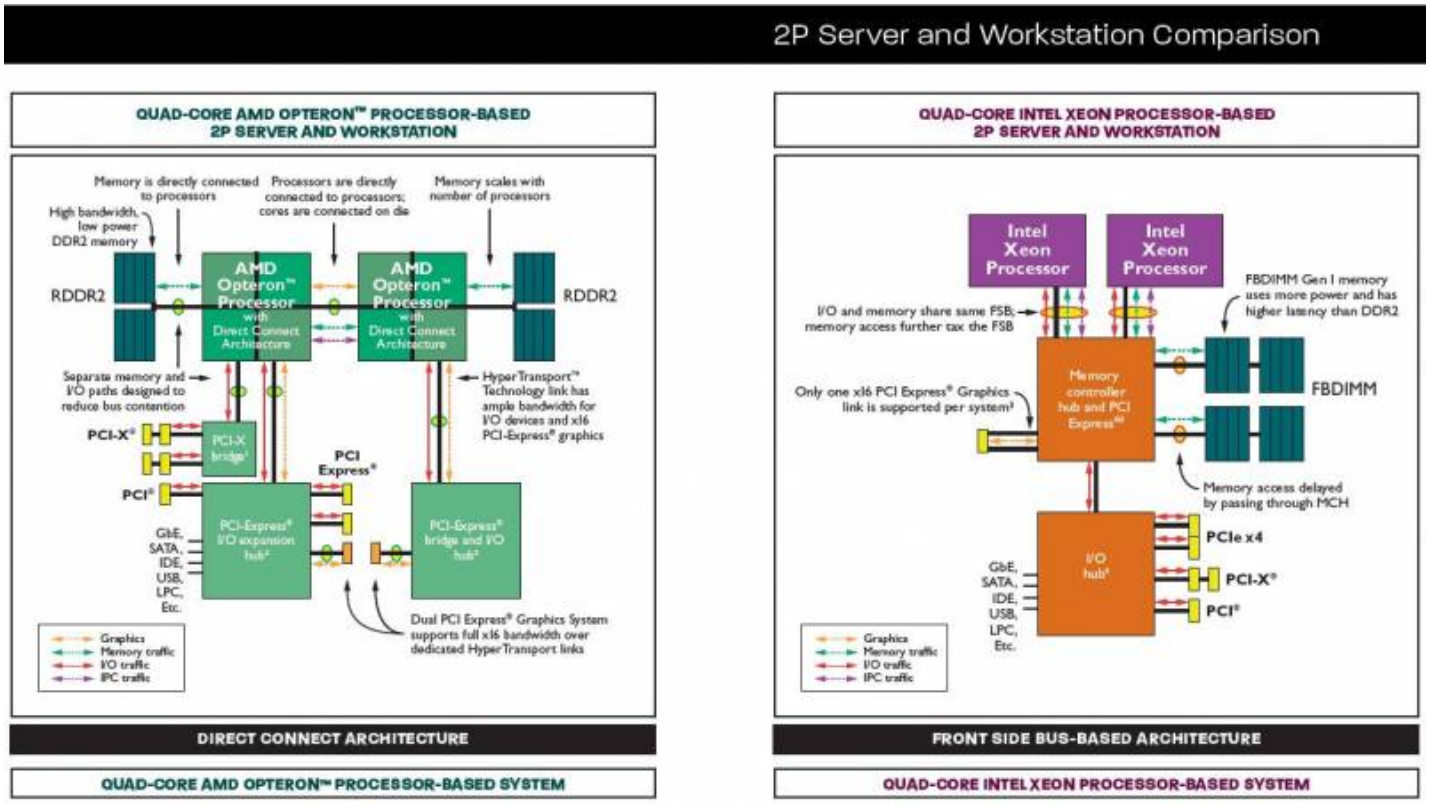

Figura 15. Comparatie intre servere multiprocesor Intel si AMD cu doua procesoare

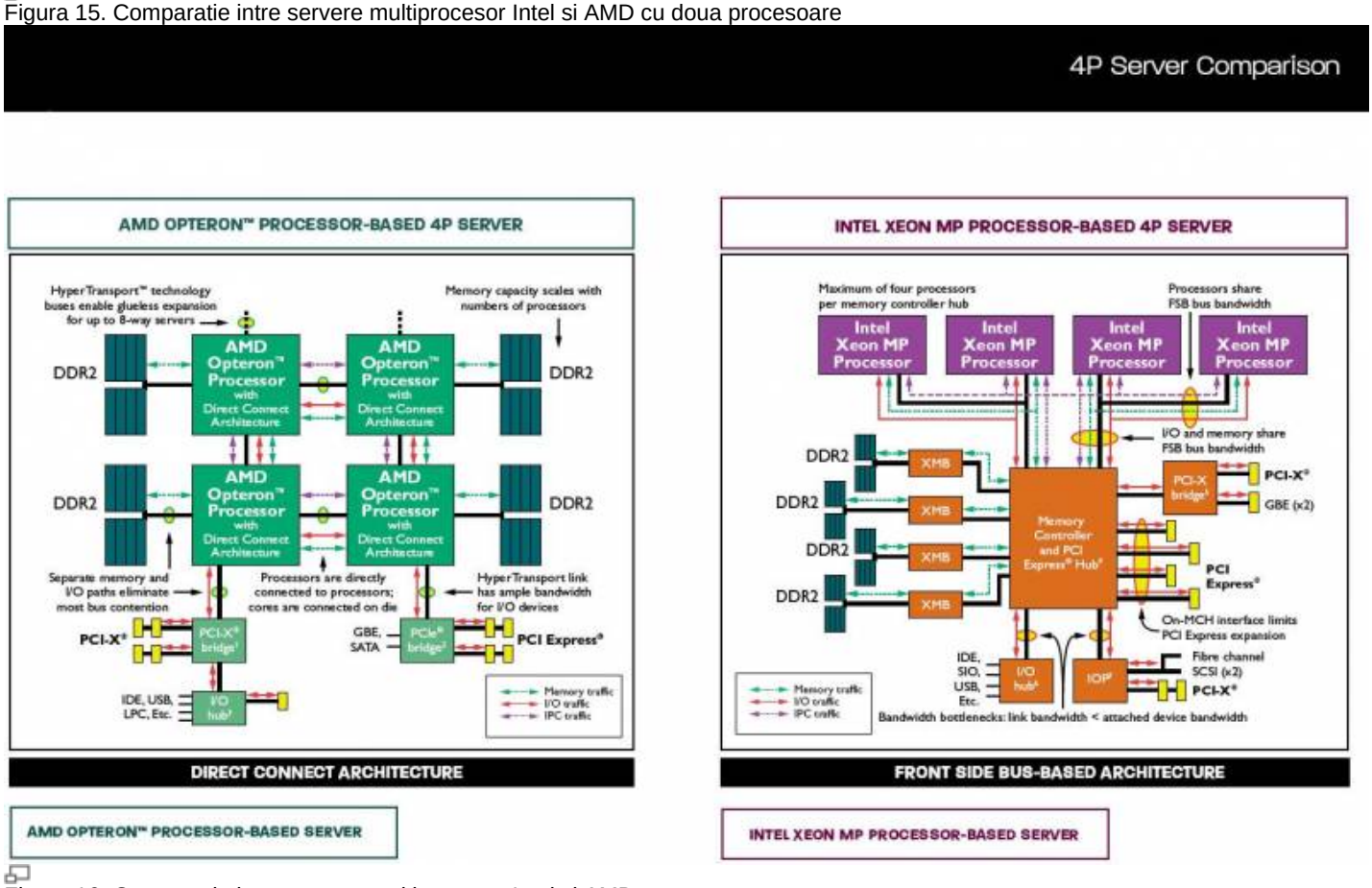

Figura 16. Comparatie intre servere multiprocesor Intel si AMD cu patru procesoare

O alta direrenta e numarul de registre generale; acest aspect are efecte imediate asupra costului de productie. Intel are 128 de registre pentru numere intregi si 128 pentru numere in virgula mobila, in timp ce AMD are numai 16 registre generale. AMD a decis aceasta abordare in urma constatarii ca 80% din cod foloseste maxim 16 registre. Aceasta abordare i-a permis sa reduca costurile.

In sfarsit, o serie de exemple de utilizare ale sistemului de interconectare QuickPath pot fi observate in urmatoarea figura. Aici procesoare din generatia Xeon - modelele Bloomfield si Nehalem - sunt legate de controller-ele placilor de baza si de sistemele de intrare - iesire prin legaturi Intel QuickPath. Acelasi tip de interconectare este utilizat si pentru legatura mai multor chipseturi intre ele.

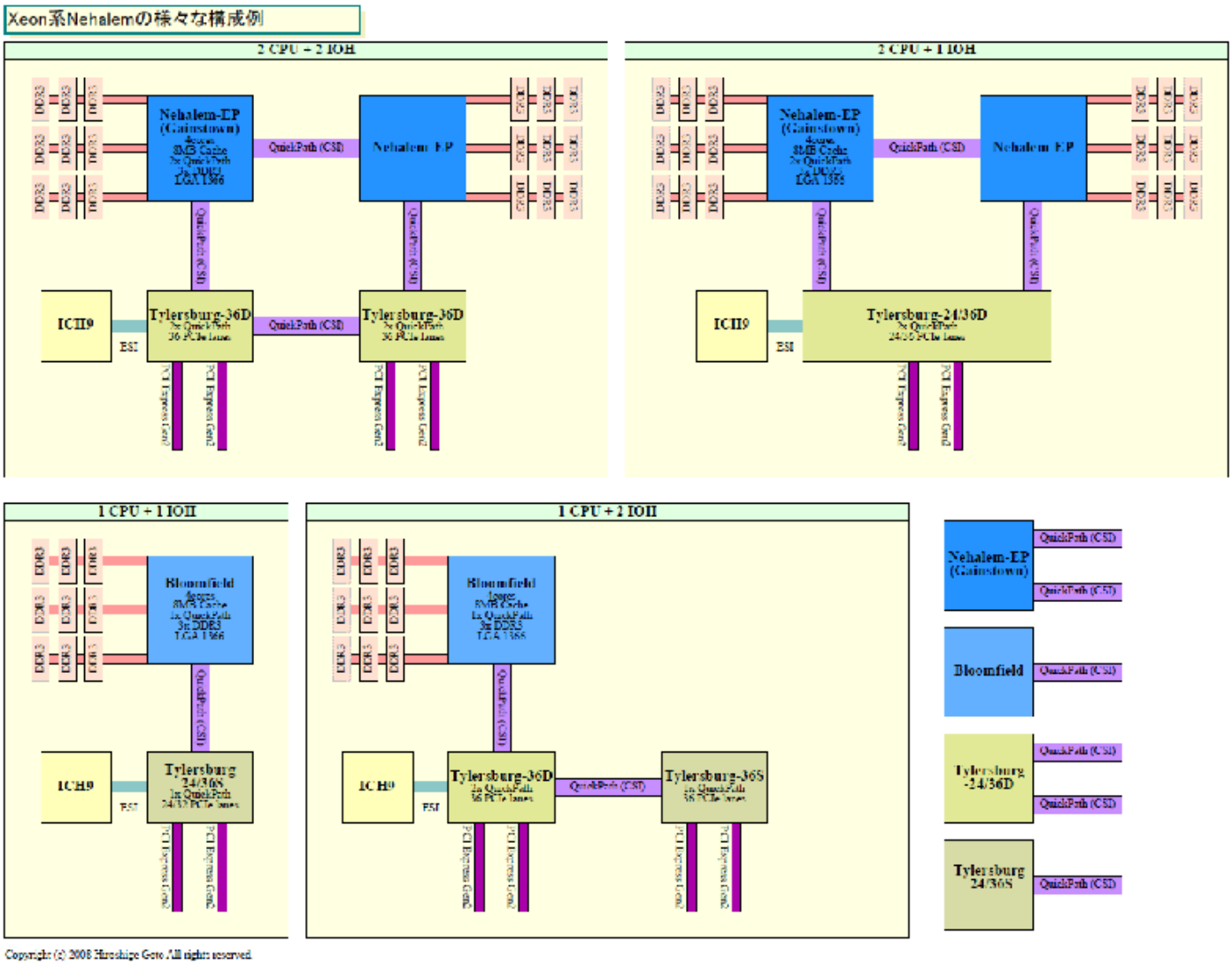

Figura 17. Exemple de utilizare a QPI impreuna cu arhitecturi de procesoare Intel (Bloomfield / Nehalem)

Dupa cum se poate observa, scalabilitatea sistemelor de calcul de-a lungul vremii este dependenta puternic de solutiile de interconectare a elementelor de procesare. Astfel, de la magistrale simple, la magistrale multiple, la solutii complexe si scalabile cum sunt Hypertransport si QuickPath Interconnect, toate s-au dezvoltat in paralel cu modelele de procesoare si chipseturi pe care le interconecteaza.

Exerciții

Se recomanda utilizarea fep.grid.pub.ro sau a sistemelor din cluster (“qlogin -q ibm-dp.q” sau “qlogin -q hp-sl.q”) pentru rezolvarea acestui laborator.

1. (**2p**) Alocați un vector de elemente `struct particle` în următoarele moduri: global, pe stivă și dinamic.
   - Porniți de la `task1a.c`, `task1b.c` și, respectiv, `task1c.c`.
   - Câte elemente poti fi alocate maxim prin fiecare metodă? Utilizati comanda `size` pentru a vedea bss unde se afla stocate variabilele globale.
   - Pentru a creste dimensiunea stack-ului utilizat de sistem puteti folosi `ulimit -s unlimited`. Pentru vizualizare si verificare a limitelor sistemului puteti incerca `ulimit -a`.
   - Pentru a face verificarile codului mai rapide, parcurgeti vectorul cu verificarea vitezei din 5M in 5M - nu pentru fiecare valoare in parte. Exercitiul este despre alocare.
2. (**2p**) Alocați dinamic o matrice de elemente `struct particle`, in mod liniarizat. Populați aleator matricea astfel încât liniile pare să conțină particule care au componentele vitezei pozitive, iar liniile impare să conțină particule care au componentele vitezei negative. Scalați apoi vitezele tuturor particulelor cu 0.5, ignorând structura matricii, prin folosirea unui cast. Introduceti o verificare pentru alocarea dinamica vs. alocarea “clasica” pe linii.
3. (**2p**) Studiați alinierea variabilelor în C.
   - Afișati adresele variabilelor declarate în schelet. Ce observați despre aceste adrese? Ce legatură există între acestea și dimensiunea tipului?

- Calculați dimensiunea structurilor a si b din scheletul de program. Afișati dimensiunea acestora folosind `sizeof`. Explicați cele observate.
- Studiați exemplul de aliniere manuală la un multiplu de 16 sau 32 bytes. In cazul in care lucrati pe o arhitectura de 32 de biti, explicati rezultatele observate.
- Bonus (**1p**) Studiati efectul optimizarilor de compilator -O2 sau -O3 asupra alinierii datelor.

4. (**2p**) Determinați dimensiunea cache-urilor L1 și L2 folosindu-vă de metoda prezentată în curs (slide-urile 72-77). Utilizati variabile char/int8_t pentru acest task.
   - Folosiți makefile-ul pus la dispoziție pentru a genera graficele.
5. (**2p**) Determinați dimensiunea unei linii de cache folosindu-vă de metoda prezentată în curs (slide-urile 67-71).
   - Folosiți makefile-ul pus la dispoziție pentru a genera graficele. Utilizati variabile char/int8_t pentru acest task.
   - Generați grafice pentru mai multe dimensiuni ale vectorului parcurs astfel încât să depășească mărimea cache-ului L1, L2, respectiv, L3.
   - Mecanismele hardware avansate implementate în arhitecturile de procesoare actuale generează comportamente complexe care nu corespund nepărat modelului simplu prezentat în curs. Aceste mecanisme pot chiar masca dimensiunea reală a linei de cache, fiind astfel necesară testarea cu diferite valori ale vectorului parcurs pentru a putea trage o concluzie informată.
   - Studiati si explicati de ce codurile (identice) si makefile-urile (diferite) de la taskurile 4 si 5 duc la graficele obtinute in cadrul laboratorului.
   - Bonus (**1p**): Combinati intr-un singur grafic “relevant” rezultatele prezentate in taskurile 4 si 5.

Tips

Pentru verificarea rezultatelor obtinute pentru dimensiunea cache-ului puteti folosi si rezultatele obtinute cu:

- ```
  [student@localhost ~]$ getconf -a
  [student@localhost ~]$ cat /sys/devices/system/cpu/cpu1/cache/index0/coherency_line_size
  [student@localhost ~]$ cat /proc/cpuinfo
  ```
- ```
  [student@localhost ~]$ sudo dmidecode | less
  ```

Cautati dupa “Cache” si veti afla informatiile despre memoria cache instalata

- ```
  [student@localhost ~]$ valgrind --tool=cachegrind ./program_test
  ```

Acesta va simula rularea programului si va afisa informatii legate de accesul la L1, L2 cache si miss rate.

Un tutorial interesant si util pentru utilizarea gdb poate fi gasit in pagina echipei de SO, aici: http://ocw.cs.pub.ro/courses/so/laboratoare/resurse/gdb [http://ocw.cs.pub.ro/courses/so/laboratoare/resurse/gdb]

Resurse

- Responsabilii acestui laborator: Emil Sluşanschi, Cosmin Samoila [mailto:emil.slusanschi@cs.pub.ro]
- PDF laborator
- Schelet laborator
- Soluție laborator

Referințe

- Alinierea structurilor in C pe x86 [http://en.wikipedia.org/wiki/Data_structure_alignment#Typical_alignment_of_C_structs_on_x86]
- amd_cpu_roadmap.pdf
- amd-quad-core_opteron_2proc_server-ws_comparison.pdf
- amd-4proc_server_comparison.pdf
- amd_designing_for_n_cores.pdf
- introduction_to_amd64_2005.pdf
- coleamd.pdf
- multi-core_codingphenom.pdf
- opteronoverview.pdf
- 318082.pdf
- opteron_mpf.pdf
  ```

- opteron_data_sheet.pdf
- xeon-5600-brief.pdf
- xeon-5600-vol-1-datasheet.pdf
- xeon-5600-vol-2-datasheet.pdf
- nv_ds_tesla_c2050_c2070.pdf
- tesla-kseries-overview-lr.pdf
- Sequential Consistency vs TSO [http://blog.regehr.org/archives/898]
- The Lost Art of C Structure Packing [http://www.catb.org/esr/structure-packing/]

asc/lab4/index.txt · Last modified: 2018/03/13 15:50 by radu_petru.daia