Tema de casa Sisteme de Operare

Magistrala PCI

**Prof. coordonator:Doctor Inginer Stefan Stancescu**

 Student

Sima Viorel 433A

Cuprins

**Interfata PCI .......................................................................................................2**

**Aplicatii ale magistralei PCI.................................................................................3**

 **O vedere în asamblu a magistralei**

 **Care sunt beneficile si performantele pe care le oferă magistrala PCI**

 **Administrarea**

**Performante PCI.................................................................................................6**

**Maparea intrărilor si iesirilor în memorie pentru linux........................................7**

**Principiu de funcționare a unei de magistrale de tip PCI......................................8**

**Ciclurile pe care le parcurge magistrala în timpul funcționării.............................9**

**Fazele unui transfer de citire de pe magistrarla PCI ............................................10**

**Elapele de functionare atunci cand de face această citire ...................................11**

**Cum se face managementul intre ordonare si postare(scriere)............................12**

**Cele 2 categorii de dispozitive cu diferite cerinte pentru ordonare si postare ......13**

**Bibliografie..........................................................................................................14**

Interfața PCI

 Deși mulți utilizatori de calculator cred că de PCI este o modalitate de interconectare între cabluri electrice, de fapt este un set complet de specificații care definesc modul în care diferite părți ale unui calculator ar trebui să interacționeze.

 Caietul de sarcini PCI acoperă cele mai multe probleme legate de interfețe de calculator. Noi nu sa le acoperim pe toate aici; în această secțiune, ne preocupă în principal de modul în care un driver PCI pote găsi hardware-ul său și să obțină acces la el

 Arhitectura PCI a fost conceput ca un înlocuitor pentru standardul ISA.

Principalele obiective: pentru a obține o performanță mai bună, atunci când are loc transferul de date între calculator și perifericele sale, pentru a fi pe cat este posibil independent de platforma, și pentru a simplifica adăugarea și eliminarea perifericelor din sistem.

 Magistrala PCI atinge o performanță mai bună cu ajutorul unei rate de ceas mai mare decât ISA ,ceasul ruleaza la 25 sau 33 MHz ( rata reală fiind un factor al ceasului de sistem ) , și au fost făcute și implementări pentru 66 - MHz și chiar 133 - MHz, de asemenea, acesta este echipat cu o magistrală de date pe 32 de biți , și cu o extensie de 64 de biți care au fost incluse în caietul de sarcini .

 Independenta de platforma este de multe ori un scop în proiectarea unueimagistrale ale de calculatorului , și este o caracteristică deosebit de importantă a PCI , deoarece lumea PC a fost întotdeauna dominată de standarde de interfață - procesor specifice . PCI este în prezent folosit pe scară largă pe IA - 32 , Alpha , PowerPC , SPARC64 , și IA - 64, și alte platforme .

 Ceea ce este cel mai relevant pentru cel care scrie driverele , cu toate acestea , este suportul PCI pentru auto-detecție pe placa de interfață . Dispozitivele PCI sunt jumperless ( spre deosebire de majoritatea perifericelor mai vechi )și sunt configurate în mod automat în momentul pornirii . Apoi , driverul de dispozitiv trebuie să poată accesa informații de configurare în dispozitiv pentru a finaliza inițializarea .

 Acest lucru trebuie să se întâmple fără a fi nevoie de o cautare foarte amănunțită .

Aplicații ale magistralei locale PCI

 Aceasta magistrala a fost gandita astfel incat sa poată oferii in industrie un standard de inalta preformanta la un cost redus .Chiar daca in momentul de fata tehnologia este foarte schimbatore putem observa faptul ca PCI a ramas cam la fel cu mici modificari deoarece are o aplicabilitate foarte mare la un cost redus .

 Chiar daca prima data a fost gandita doar pentru platfomele de tip windows , acum PCI poate fi folosit si pentu aplicatile mobile prin diferite servere .Cum se stie faptul ca telefoanele mobile functionealza la o tensiune de 3,3 volti iar calculatorele folosesc 5 volti , PCI va face o descriere completa la cum va arata calea dintre aceste 2 tensiuni

 Cum in momentul de fata procesoarele sunt independente se va arăta faptul ca PCI trebuie sa fie optimizat pentru diferitele funcții de intrare sau de iesire , va deschide operatile cu procesorul sau cu memoria si de asemenea interconecteaza multe periferice .De asemenea cum aceasta magistrala este foarte larga atunci cand este folosită pentru un semnal video si astfel vom putea sa avem performante foarte bune dar daca folosim astfel inseamna ca performantele per total vor scadea deoarece banda este finita si in caz ca vrem sa executam mai multe procese de intrare sau iesire vor fi probleme in ceeea ce priveste rapiditatea de executie .

 De asemenea Magistrala locală PCI mai ofera si urmatorerele benefici pentru utilizatorii de PCI,de obicei configurariel refistrelor se vor face automat de catre sistem ceea ce va fi foarte plcaut pentru utilizator deoarece acesta nu mai trebuie sa il reconfigureze.

 **O vedere in asamblu a magistralei**

 Această diagramă va arata cum arată de obicei o arhitectura pentru o magistrală locală de PCI.Se vede ca PCI este asemănătore cu o autostrada care conectează toate celelalte drumuri nationale care vin de la diferite componente ale placii de bază

 De obicei pentru o implementare locală a magistrlaei PCI aceasta poate sa suporte 4 conectori ,de asemenea se mai pot sa se fac diferite extinderi dar de obicei această configurație dă cele mai bune rezultate .Ca să se poată face o trecere corectă între 5V si 3.3V vor exisa 3 moduri de conectare 1 pentru 5 volti unul dual pentu 3.3 si 5V si altul pentru 3.3 volti

 **Care sunt beneficile si performantele pe care le oferă magistrala PCI**

Aceasta magistrala a fost gandită ca să poata sa reziste in timp la diferitele schimbari

Performanta crescuta

deoarece este capabila sa concureze cu subsistemul unuei memorii sau al unui procesor

sa poata se sincronizeze magistrala pana a 33MHZ sau pana la 66 de MHZ

Cost scazut

 cum este de mult timp de piata tehnoogia fiind ne schimbata traseele au ajuns apoape de perfectiune si astfel nu este nevoie pentru noi studii in aceasta parte

Este ușor de folosit

este usor de folosit deoarece registrele se pot configura automat

Are o longivitate estinsa deoare s-a adaptat ireprosabil la noua gama de calculatore(telefoanele mobile )

Are o rentabilitate crescuta deoarece dupa cum am spus si mai devreme tipmul pe piata isi spune cuvantul adica daca nu ar fi avut o rentabilitate asa de mare nu ar fi rezistat pe piața

Are o compabilitate foarte crescuta deoarece majoritatea dispozitivelor o folosesc si de asemenea asigura si o integritate foarte bună pentru date

 **Administrarea**

aceasta administrare este facută de catre PCI SIG care trebuie:

să mentina o compatibilitate pentru toate magistralele locale de tip PCI

să mentina magistrala PCI pe cat de simpla cu putințaă posibilă că poata sa fie ușor de implementat

să poata sa asigure o longevitate cat mai mare pentu acest tip de magistrala

Magistrala PCI- performanțe

 Aceasta magistrala mai poate fi denumita si magistrală mezanin deoarece ea va adauga un alt nivel de configuratie de magistrală .Ea va ocoli magistrala care este de obicei standard de intrare/ iesire ,și va folosi o magistrala a sistemului deoarece facand acest lucru se poate creste viteza ceasului .Viteza ceasului e foarte importanta deoarece în urma folosiri unui test de tip brenchmark se poate vede o performanța mut mai crescuta a sistemului și astfel se poate cere un preț mai bun de catre producator care va duce de asemenea la un profabilitate mai buna a produsului .

 De obicei pentru magistrala PCI invormațiile vor avea o viteza de transfer de aproximativ 33 MHz .Lățimea de bandă va fi de aproximativ 132 de MB/s.Dacă magistrala va fi utilizata cu o unitate CPU pe 64 de biți atunci lungimea de bandă se va dubla ceea ce inseamnă ca vitezele de transfer vor ajunge pana la dublul celei din naite adica 264 de MB/s.Dar în ciuda tuturor acestor modificări vitezele de transfer a datelor vor fi mai mici ,putem pune acest lucru pe sema ca în toata aceasta discuție pe carea am avut-o în acest studiu a fost una ideeala care nu tine seama de efecte de gîtuire sau alte efecte care chiar daca vor avea o probabilitate de apariție foarte joasă ele tot vor exista ceea ce inseamna ca nici o data nu fom avea perfectiune .

 Putem să ne găndim cu această ocazie la noi atunci cănd cumpăram un nou calculator cu un anumit spatiu de stocare se observa ca de fiecare data comerciantul nu va zice dimensiunea exacta a HDD ci doar o aproximare DAR acceastă aproximare va fi de ficare dată în plus putem dezvolta aceasta idee dar nu este scopul nostru din acest studiu de caz.

 Putem să ne gandim la cauzele care ar face ca această magistrala sa fie mai rapidă unul dintre aceste ete faptul ca acestă magistrală lucrează in paralel cu magistrala procesorului .Astfel atat timp cînd unitatea CPU procesează date din memoria externa, magistrala PCI va putea sa se ocupe cu transferare de informatii intre alte elemente ale sistemului , prin acest lucru magistrala de tip PCI a inceput sa fie din ce in ce mai populară

**Maparea intrarilor și iesirior în memorie pentru linux**

 Toate placile au si ele propile lor memorii de exemplu placa video are indeajuns[ memorie ca sa poata sa afiseze ecranul , deci cum ar putea un driver sa aceseze această memorie , daca ne gandim asupra CPU putem sa vedem ca acesta tine seama de 3 tipuri de memeorie

memorie care este adresată logic

memorie care este adresata iniat

memorie cu adrese fizice

IA32 va translata aceste adrese logice catre adrese liniare folosind un system de segmentare , iar translatia adreselor liniare catre cele fizice se fae folosind paginarea sistemului .Adresele fizice sunt puse in magistrala de date ca sa poata sa acceseze memoria de tip RAM si ROM .majoritatea CPU no vor folosi o adressare segmentata ci o adresare paginata

Segmentarea de tipu IA32 are avantaje dar de obicei este privit ca o segmentare slaba deoarece sistemele actuale sunt capabile sa optimizeze foarte bine segmentarea.

 Memoria IO este de asemenea un alt spatiu de adresare care este denumit magistrala spatiului de adresare in IA32 magistrala de adrese este o adresare fizică dar nu este la fel pentru toate tipurile de arhtecturi .

 Sa ne indreptam privirea asupra IA32 in continuare , pentru magistrala PCI bufferu placii video este mapat in adresele fizice dincolo de principala memorie fizica , aceste programe nu acceseaza memoria direct ceea ce se face prin acestea este sa mapeze memoria IO in memoria virtuala care este accesibila drivereleor acesta este facuta cu ajutoul une functii Linux speciala

 In concluzie cum driverele nu pot sa ajunga la memoria fizica se fac niste eptape astfel incat sa poata sa comunice cu memoria virtuala si astfel sa poata sa comunice indirect cu cea fizica

**In aceasta schemă se poate vedea principiu de funcționare a unei de magistrale de tip PCI**



In cadrul unei magistrale activitatea se va desfăsura su forma unui transfer intre un master si un slave ,astfel dupa ce master va obține constrolul asupra magistrale va fi determinat pupul de transfer care va urma .

**Ciclurile pentru o magistrală sunt următorele**

1. Aceptarea de intrerupere .Acesta este un ciclu de R care este conceput pentru a fi un control de intreruperi pentr PCI, în aceasta faza linile de adresă nu sunt folosite în faza de adresare iar apoi in faza de date va fi indicat vectorul de intrerupere .

Putem sa ne găndim la urmatorul esxemplu in care vom avea mai multe intreruperi cu diferite prioritati , intreruprile care se vor executa vor fie cele cu indicele cel mai mic ,in urma acestui exemplu se poate executa si o diagrama în care am putea să vedem cum se manifesta aceste intreruperi pentru un procesor .

2.Ciclul special. Acesta dupa cum îi spune numele este unul de care foarte puțini dintre noi au auzit si este folosit pentru trimiterea unui mesaj catre o țintă sau cașre mai multe ținte

3.Citirea I/O. Personal mi se pare ca acest ciclu este unul dintre cele mai imortante cicluri deoarece acesta face legătura (transferul de date ) intre calculator si acel dispozitiv , de asemenea atunci cand se face această comunicare va fi nevoie ca sa exite un spațiu propriu de adresare

4.Scriere I/O

5.Citirea din memorie. Aceste citiri pot sa fie executate în parti(pe blocuri ), contează cum cum controlerul de memorie va gestiona transferurile intre memorie si memoria de tip cache , de obicei cand se citi o linie de adresare cach va fi nevoit ca acest lucru se se faca de aproximativ de 2 ori deoarece dupa cum am spus citirea se va face in blocuri dar aceste blocuri nu pot sa aibă o dimensiune foarte mare.

6.Citire o linie din memorie . De obicei cand se face acest lucru aceasta citire va trebui sa se faca intre 3-12 cicluri de citire

7. Citire multiplă din memorie . Se vor citi mai mult de 3 linii de cache in bloc vor exista cel puțin 12 cicluri de citire

8.Scriere în memorie

9. Sciere in memorie si invalidare .Atunci cand se face acest lucru va trebui sa se ținaă cont în primul rînd de modul Write Back pentru scrierea din cache în memorie

10.Citire configurație .

Pentru fiecare dispozitiv PCI se va include un set de 256 de registre interne , care vor fi foloste pentru faza de configurare . Astfel in acest timp dispozitivul va putea să citească registrele de configurare

11.Scriere configurație . Acum dupa ce in pasul anterior masterul a citit registrele el poate sa le scrie in acest moment

12.Ciclu de adresare dual.

Acesta este folosit ca sa arate ca va apărea o adresare pe de biți

Pentru a putea transfera date pe o magistrală de tip PCI va fi nevoie de o fază de adresare si de una sau de mai multe faze de date

**In aceasta imagine se pot vedea fazele unui transfer de citire de pe magistrarla PCI**



Linile care sunt implicate în transfer sunt următorele

CLK-ceasul sitemului cel mai important semnal deoarece acesta da tipmul

ex pentru un microprocesor in functie de acest semnala se stabileste baza de timp dupa care ne vom putea da seama ca se va intpla atunci cand vom altera celelalte semnale

AD-este semnalul care ste folosit pentru a face linile de multiplexare pentru adrese și date

C/BE- acesta este folosit pentru a arata ca cei octeți poartă date semnificativve

FRAME- este folosit pentru a indica inceputul transferului o data cu inceperea fazei finale de date acesta este dezactivat

IRDY-va indica faptul ca masterul este gata să accepte date

TRDY -la citire pe AD se vor prezenta date valide iar la sciere va indica ca slave este gata să acepte date

DEVESEL - semnal care este activat de catre slave atunci cand va fi recunoscută adresa

**In următoarele rînduri vom prezenta care sunt elapele de functionare**

1.Dacă masterul va avea control asupra magistralei atunci linia va ramane activă pana cand inițiatorul va termina ultima fază de date

2.Apoi slave va recunoaste adresea de pe linile AD

3.Masterul va lasa magistrala ca sa fie folosită de slave si inițiatorul va schimba informații pe liniile AD

4.Slave va activa sau dezactiva semnalul de DESEVEL ca să arate ca a recunoscut sau nu adresa, apoi va pune datele pe linile ADsi va activa TRDY

5.Slave va avea nevoie de un timp ca să pregătească urmatorul octet de date

6. Se va dezactiva TRDY , astfel inițiatorul nu va citi datele de ala inceputul tactului 5 ci de la 6 , astfel se adaugă o stare de tip WAIT

7.După tactul 6, slave va pune pe magistrală al 3-lea octet de date ,apoi initiatorul va dezactiva IRDY

8.Apoi inițiatorul va dezactiva semnalul de FRAME si se va activa semnalul de IRDY

9.Initiatorul va dezactiva IRDY și va permite magistralei să treacă in starea inițială.Slave a detectat dezactivare FRAME si va dezactiva si TRDY si DVESEL.

**Cum se face managementul intre ordonare si postare(scriere)**

 Ca se poată să se facă o ordonare in cadrul PCI trebuie sa se satisfacă 3 lucruri

In primul rand rezultatele scrise de la unul dintre mastari oriunde in sistem este observat de alt master oriunde in sistem si acest lucru se va face in ordinea originala

 Daca masterii sunt in diferite parti ale sistemului adica nu sunt legati intre ei nu este nevoie deoarece vor avea diferiti consumatori deci nu va trebui sa existe un algoritm pentru ordonare

 Pentru aces caz regulile lasa ca parit din memorie care au fost scrise sa poata sa fie rearanjate deoarece nu afectează sistemul .In al 2 lea caz ei lasa ca unele tranzactii sa fie scrise si astfel se va inbunătății performanta iar in al 3 lea rand trebuie să conditioneaza condiția de magistralei deadlock.

 O alta parte a tranzactiei se afla in faptul ca trebuie să se stie in ce orfine au fost făcute aceste tranzacții, iar unul dintre indicatorii acestuia este sa vedem timpul la care s-a oprit această tranzacție .Daca tranzactia a fost terminata cu mesajul cheie Retry(incearca din nou) inseamna ca informația nu a fost transmisă corespunzator si inseamna ca este nevoie de o reincepere a transmisiei ,daca mesajul este Abort inseamna ca a aparut o eroare majora este de recomandat sa se opresca si să se faca soluționarea acestei probleme

 In caz ca in cadrul unui master sa zicem masterul 1 va foloso rezultatele din masterul 2 nu trebuie activat masterul 1 pana nu s-au terminat toate operatile din masterul 2 in caz contrar va duce la blocarea magistralei de asemenea atunci cand apare mesajul de Retray nu inseamna neaparat ca va incepe cu aceleasi ci poate sa treaca la urmatoarea instructiune fara sa deranjeze calculatorul

 Aceste tranzactii pot fi impărțite in 2 feluri bazândune pe felul in care ere sunt folositede catre agentul intermedial pe nume brige .(dupa cum ii spune si numele acest agen formeaza un pod).Cele 2 grupuri adica a fost postat sau nu a fost inca postat .Acele tranzactii care sunt postate inseamnă ca si-au terminat traba inca din timpul dispozitivului din care proveneau pe cand cele care nu s-au postat inca inseamna ca ele vor fi postate atunci când vor ajunge la destinație .in tot acest timp masterul poate sa faca alte lucruri pana asteaptaă ca aceste postări sa fie făcute

 Agentii acestia precum bridge fac ca caleaacestor posturi sa nu fie intreruptă iar timpul de execuție să fie căt mai mic cu putință

 In cazul celor ne postate masterul nu pote sa faca alt fel de muncă pana ce aceste tranzacții nu au fost făcute

**Cele 2 categorii de dispozitive cu diferite cerinte pentru ordonare si postare**

Tranzactia ,ordonarea si postarea penstu dispozitivele simple

 Un dispozitiv simplu este orice dispozitiv care actioneaza ca o magistrală si nuare nevoie ca datele sale să fie postate la magistrala logica,in general aceste dispozitive nu se conectează direct la CPU ci sunt implementate ca si dispozitive simple

 Tinta si masterul pentru interfața de tip PCI sunt compet independente. Un dispozitiv simpu nu poate sa facă completari la tranzactii ca fiind o parte din aceea tranzactie decat daca tranzactia s-a terminat

 De asemene dispozitivele simpe pot să opreasc o tranzactie cu ajutorul mesajului de Retray si astfel dupa ce este oprită tranzactia sa se rezolve ceea ce trebuia sa se rezolve.

Tranzactile ordonarile si postările pentru poduri

 Un pod este orice dispozitiv care implementeeaza postari interne si depasiri are memoriei in trimpul tranzactiei,tranzitiile de screre trebuie sa fie executate de catere un disozitiv ca fiind un master pe magistrala de PCI

 De obicei the bridges(podurile) alatureaza 2 macistrala asa cum sunt cele 2 magistrale PCI , o magistrală gazdă si o magistrala PCI ,sau o magistrală PCI si o magistrală locală CPU.

 Briges le este permis sa posteze actiuni de scriere a menorii si se pot deplase pe orcare directie pe acest bridge.urmatorele reguli de ordonare duc la rezultate care permit unuia dintere masteri sa vizualizeze tranzactile facute de ceilalti masteri in ordinea corecta sau pote si prin citirea informatiei care se afla pe bridge

 De asemenea ele garanteaza ca aceea magistrala nu se comporta ca un deadlock atunci cand bridge incearca sa goleasca bufferele sale postate

 Memoria postata isi va termina aceast parcurgere a bridge daca a ajuns in aceeasi ordine cum era si pe magistrala din care a originat

 Daca tranzactia are o singura directe atunci nu trebuie sa tina seama decat de "semnele de circulatie "

 Bufferele de memorie scirse in ambele directii trebuie sa fie golite pana la terminarea tranzitiilor

 Un bridge nu va putea sa accepte ca sa accepte o tranzactie care are ca tinta o arta dranzac'ie deoarece s-ar forma un loop infinit

Bibliografie

<http://www.ics.uci.edu/~harris/ics216/pci/PCI_22.pdf>

<http://student.cosy.sbg.ac.at/~tstuetz/out/ldd/ch6_ldd.pdf>

http://atm.neuro.pub.ro/

<http://lwn.net/images/pdf/LDD3/ch12.pdf>