SAPIENZA
UNIVERSITÀ DI ROMA

Da definire

Facoltà di Ingegneria dell'informazione, informatica e statistica

Corso di Laurea in Informatica

Candidato

Leonardo Ganzaroli

Matricola 1961846

Relatore

Prof. Luigi Cinque

Correlatore

Dr. Diego Bellani

Anno Accademico 2024/2025

Da definire
Tesi di Laurea. Sapienza – Università di Roma

Questa tesi è stata composta con LaTeX e la classe Sapthesis.

Email dell'autore: ganzaroli.leonardo@gmail.com

Sommario

write your abstract here
Scaletta:

- Introduzione
  - Acceleratori hw, intel 8087, cenni storici
  - Esplosione degli stessi e cause
  - PCI, PCIe (cenni) causa della popolarità
  - Obiettivo, disaccopiamento hw e sw
- teoria PCIe
- Linux, driver
- QEMU, QOM, limiti
- Processo di implementazione
- Conclusioni (riassunto)

Indice

Capitolo 1

Introduzione

Questo capitolo

1.1 Acceleratori hardware

Gli acceleratori hardware sono dei componenti hardware progettati appositamente per svolgere determinate funzioni, essi permettono di svolgere quelle funzioni in modo più efficiente rispetto alla classica combinazione di software e CPU. I possibili campi d'applicazione sono molteplici ma alcuni mostrano chiaramente il vantaggio di delegare i compiti, per esempio:

- In crittografia sono spesso richiesti dei calcoli particolarmente intensivi, evitare di svolgerli nella CPU comporta una notevole riduzione del carico sulla stessa
- L'elaborazione grafica richiede molti calcoli che possono essere ottimizzati svolgendoli in parallelo, data la sua natura sequenziale la CPU non risulta la scelta migliore
- L'elaborazione dei segnali audio deve essere svolta in tempo reale e potrebbe richiedere delle operazioni aggiuntive, la CPU potrebbe non riuscire a gestire il tutto senza ritardi

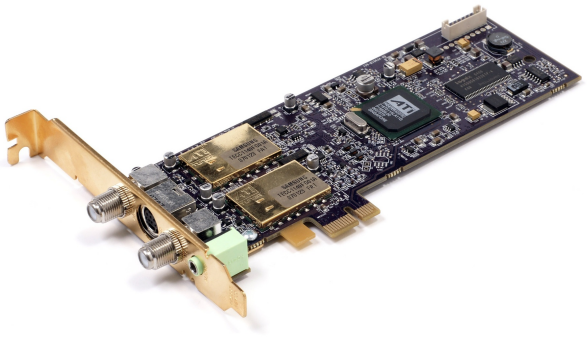

Figura 1.1. TV Wonder HD 650 di ATI Technologies

1.1.1 Evoluzione nel tempo

I primi acceleratori nacquero alla fine degli anni '70 / inizio anni '80 sottoforma di semplici coprocessori matematici atti a svolgere calcoli aritmetici complessi, con l'avanzare del tempo e conseguentemente della tecnologia sono stati sviluppati acceleratori sempre più vari e potenti. Come si vedrà in 1.1.2 gli acceleratori più diffusi in questo periodo sono quelli che si occupano di grafica e quelli relativi all'IA, per dare un'idea dell'evoluzione avvenuta seguono 3 esempi provenienti da anni diversi:

1. **Intel 8087** (1980)

   Si tratta del primo coprocessore matematico progettato per i processori 8086 e 8088, il suo scopo principale era quello di accelerare i calcoli in virgola mobile ma poteva anche calcolare funzioni come quelle trigonometriche ed i logaritmi [1].

2. **Sun Crypto Accelerator 1000** (2002)

   Lo scopo di questo acceleratore è quello di effettuare la computazione di vari algoritmi crittografici usati principalmente nei protocolli di sicurezza in ambito e-commerce. Essendo creata solamente per svolgere calcoli non presenta alcuna interfaccia esterna tranne quella PCI per il collegamento all'host [2].

3. **Pixel Visual Core** (2017)

   Presente negli smartphone Pixel 2 e Pixel 2 XL di Google, è un coprocessore che si occupa dell'elaborazione fotografica. Oltre al miglioramento delle foto permette di eseguire applicazioni di Machine Learning come il riconoscimento dei volti [3].

Con il tempo anche l'architettura generale degli elaboratori ha subito un'evoluzione, l'8087 aggiungeva 68 istruzioni al set dell'8086 ed era collegato in parallelo ad esso, quando arrivava una sua istruzione il coprocessore prendeva il controllo e ad operazione avvenuta mandava un segnale al processore [1].

Risulta evidente che questo metodo comporta uno spreco della CPU, infatti ogni volta che deve essere eseguita anche una semplice operazione come $0.1 + 0.2$ il coprocessore (anche se per un tempo irrisorio) mantiene il controllo e non permette alla CPU di fare altro.

Nei sistemi odierni sarebbe impensabile usare questo metodo, considerando la mole di dati comunemente interessata in un'operazione di un dispositivo si perderebbe una grande quantità di tempo. Per risolvere il problema si usa l'accesso diretto alla memoria, si usa un apposito chip detto *Controllore* che contiene 4 registri:

- Il primo contiene l'indirizzo di partenza in memoria
- Il secondo il numero di byte interessati
- Il terzo l'indirizzo del dispositivo o dello spazio I/O interessato
- Il quarto indica lettura o scrittura

Così facendo si interpella la CPU solo all'inizio per settare i registri del controllore e alla fine tramite un apposito segnale, anche se in questo modo la CPU è libera di svolgere altri compiti non può comunque accedere al BUS perché il controllore ha priorità più alta. Ciò nonostante il guadagno ottenuto supera di gran lunga quest'ultimo problema [4].

1.1.2 Cause della popolarità

1.1.3 Metodi di connessione

1.2 Obiettivo

Capitolo 2

PCIe

2.1 *Fabric*

2.2 Configurazione software

2.3 Richieste al sistema operativo

Capitolo 3

Linux

3.1 Gestione dei driver

Capitolo 4

QEMU

4.1 QOM

4.2 Limitazioni

Capitolo 5

Implementazione

Capitolo 6

Conclusioni

The grasping power of the mirror..

Ringraziamenti

Bibliografia

[1] Intel, *8087 Math Coprocessor Datasheet*, October 1989.

[2] Oracle Corporation, *Sun Crypto Accelerator 1000 Installation and User's Guide, Version 2.0*, October 2005.

[3] Ofer Shacham and Masumi Reynders, "Pixel visual core: Image processing and machine learning on pixel 2." https://blog.google/products/pixel/pixel-visual-core-image-processing-and-machine-learning-pixel-2/, 2017. Accessed: 2025-06-24.

[4] Andrew S. Tanenbaum and Todd Austin, *Architettura dei calcolatori: Un approccio strutturale*. Pearson Education Italia, 5 ed., 2006.