



ACCELERATORI HARDWARE NELLA COMPUTER VISION

INTRODUZIONE

- La potenza di calcolo delle **CPU** risulta **insufficiente** per gran parte delle applicazioni.
- Utilizzo di **acceleratori hardware**: DSP, FPGA e GPU.
- Differiscono tra loro per **potenza di calcolo, velocità di sviluppo** di algoritmi specifici, **consumo di energia** e **prezzo**.
- Applicazioni differenti richiedono acceleratori hardware differenti.
- Il semplice confronto su **tempi di calcolo** non può essere l'unico metro di valutazione poiché è dipendente dalla bravura del programmatore.

DSP - INTRODUZIONE

- Microprocessori con architettura specializzata per **l'elaborazione di segnali**.
- Texas Instruments è uno dei maggiori produttori di DSP utilizzati per la computer vision.
- Possiamo dividere i suoi DSP in **4 gruppi principali**, in funzione della potenza di calcolo e consumo.

DSP - TIPOLOGIE(1)

- **Ultra-low power DSP:** economici ma con bassa potenza computazionale. Utilizzo limitato ad algoritmi semplici.
- **Power optimised DSP:** per dispositivi portatili o mobili dove il consumo di energia è la caratteristica più importante. Possono elaborare algoritmi con un livello di complessità moderato.
- **DaVinci digital media processors:** per applicazioni multimediali come l'elaborazione di video e immagini e la cattura video. Includono codec hardware per video e immagini (ad esempio, MPEG, H.264 e JPEG) e acceleratori hardware per l'elaborazione video. Includono un processore ARM. Rispetto ai Power optimised DSP possono eseguire compiti più complessi ma con un consumo maggiore.

DSP - TIPOLOGIE(2)

- **Multicore DSP:** Ottimizzati per L'HPC. Sono costituiti da più core DSP (fino a 8 core), che permettono loro di eseguire compiti in parallelo. Parallelismo simile a quello dei CPU multicore nei PC. Le prestazioni computazionali massime dichiarate assumono una completa parallelizzazione degli algoritmi. Nella pratica sono richieste tecniche avanzate di programmazione parallela per ottimizzare la velocità di calcolo di questi DSP.

DSP - SUPPORTO ARITMETICO E LIBRERIE

- **Supporto Fixed-Point vs Floating-Point:** Il tipo di supporto aritmetico è un fattore cruciale nella scelta di un DSP.
 - **Floating-Point:** Facilita l'implementazione degli algoritmi e aumenta la precisione rispetto al fixed-point. Più adatto per applicazioni che richiedono alta precisione nei calcoli.
 - **Fixed-Point:** Esegue operazioni con meno bit, ma richiede che il programmatore riposizioni con attenzione il punto decimale dopo ogni operazione matematica. Solitamente più economici dei DSP a floating-point.
- Molte **librerie** per computer vision sono disponibili e i tempi di sviluppo per un task su DSP single core sono relativamente brevi.

DSP - VANTAGGI

- **Costi Ridotti:** DSP economici rispetto ad altri hardware per dispositivi portatili.
- **Basso Consumo di Energia:** Consumo energetico ridotto, adatto per dispositivi portatili.
- **Gestione Periferiche e Protocolli:** Supportano periferiche, porte standard e protocolli di comunicazione.

DSP - SVANTAGGI

- **Limitati per Complessità Elevata:** DSP multicore non adatti per applicazioni ad alta velocità o alto throughput di dati.
- **Preferibili per Elaborazione Sequenziale:** Non ottimali per algoritmi massicciamente paralleli.
- **Efficienza Ridotta con CPU nei PC:** Uso combinato con CPU nei PC non sempre efficiente.
- **Costo delle Schede DSP:** Nonostante i chip DSP siano economici, le schede hanno costi paragonabili a quelli delle schede GPU.

FPGA - INTRODUZIONE

- Le FPGA (Field-Programmable Gate Arrays) contengono array di porte logiche riprogrammabili. A differenza di CPU, DSP e GPU, non hanno un'architettura prestrutturata o un'unità di elaborazione centrale.
- La programmazione degli FPGA richiede l'uso di linguaggi di descrizione hardware (HDL), come VHDL e Verilog, linguaggi a basso livello. Sono disponibili anche linguaggi di programmazione di alto livello (ad esempio, SystemC) e linguaggi domain-specific.
- Xilinx e Altera sono i principali fornitori di FPGA nel campo della computer vision.

FPGA - FEATURE

1. Configurable Logic Blocks (CLBs):

1. Sono le risorse hardware e le porte logiche utilizzate per l'implementazione di algoritmi.
2. Le celle logiche interconnesse formano un circuito logico unificato che rappresenta l'algoritmo.
3. Un gran numero di CLBs consente l'implementazione di algoritmi complessi.
4. Il numero di CLB non è una metrica diretta per il confronto a causa di variazioni nel numero di porte logiche e di bit in ingresso.

2. Slices DSP:

1. Blocchi essenziali per operazioni matematiche di base e compiti di elaborazione del segnale.
2. Prestazioni misurate in operazioni massime al secondo.
3. Bottleneck in applicazioni con alto throughput come quelle real-time.

3. RAM a Blocchi:

1. Costruite con architettura SRAM per lo stoccaggio e il buffering dei dati.
2. Fondamentali per lo stoccaggio di immagini all'interno dell'FPGA.
3. La capacità di stoccaggio è un fattore chiave nelle applicazioni di elaborazione di immagini e visione artificiale.

FPGA - SVILUPPO

- **Tempo di Sviluppo Esteso:** Lo sviluppo su FPGA richiede più tempo rispetto ad altri acceleratori hardware.
- **Sfide di Programmazione:** Richiede programmatori esperti con conoscenza approfondita dell'hardware FPGA. Lo sviluppo di codici robusti è complesso e spesso basato su funzioni di base, il che allunga il tempo di sviluppo.
- **Evoluzione degli HDL:** Si prevede una riduzione dei tempi di sviluppo con l'introduzione di nuove generazioni di linguaggi di descrizione hardware (HDL), come Bluespec System Verilog e Chisel.

FPGA - VANTAGGI

- **Velocità di Elaborazione Elevata:** FPGAs di Xilinx e Altera processano miliardi di operazioni al secondo, superando altri acceleratori.
- **Alto Throughput di Dati:** Ideali per schede di acquisizione dati.
- **Natura Parallela e Riconfigurabile:** Permette la progettazione di hardware FPGA per applicazioni ad alte prestazioni.
- **Efficienza Energetica:** FPGAs mostrano il più alto potere di elaborazione normalizzato per il consumo di energia.
- **Bassi Requisiti di Potenza:** Adatti per dispositivi portatili.
- **Adattabilità per ASICs:** I codici FPGA possono essere adattati per circuiti integrati specifici per applicazioni (ASICs), riducendo i costi di produzione di massa.
- **Comunicazione Semplificata:** L'utilizzo dell'interfaccia PCIe tra board FPGA e il PC host è stato semplificato molto grazie a package open source.

FPGA - SVANTAGGI

- **Lungo Tempo di Sviluppo:** Principale svantaggio degli FPGA.
- **Sfide nello Sviluppo di Codici Efficienti:** Nonostante l'introduzione di nuovi strumenti per semplificare e velocizzare questo processo, è ancora richiesta una conoscenza tecnica approfondita per sviluppare codici robusti su FPGA.

GPU - INTRODUZIONE

- Le GPU sono composte da **numerosi core** di elaborazione, ottimizzate per calcoli matriciali veloci e paralleli (le immagini sono matrici 2D).
- Dispositivi generalmente accessibili, con lo sviluppo trainato dall'industria dei videogiochi. Le GPU sono efficienti in termini di costi per algoritmi massicciamente paralleli e utilizzate in una vasta gamma di applicazioni.
- **NVidia** è il fornitore più noto.

GPU - FEATURES(1)

- **Microarchitetture NVidia:** Diverse microarchitetture. Ogni generazione introduce nuove funzionalità e miglioramenti.
 - Esempi di Miglioramenti: Kepler aumenta la potenza computazionale, Maxwell ottimizza l'efficienza energetica e lo scheduler, Pascal accelera il clock, e Volta incrementa la larghezza di banda della memoria.
- **Memoria GPU:** Varia in tipo e tecnologia ed ha differenti velocità e larghezza di banda. Bisogna inoltre considerare la quantità di memoria richiesta dalle specifiche applicazioni per lo storage delle immagini.

GPU - FEATURES(2)

- **Core CUDA:** I core CUDA o stream processors sono le unità di elaborazione più piccole delle GPU NVidia, variano in numero a seconda della microarchitettura.
 - **Esempi di Architettura SM:** Fermi ha 32 core per SM, Kepler 192, Maxwell 128, e Pascal 64.
- **Compute Capability:** misura che definisce le caratteristiche e le capacità di una GPU basata su CUDA. Serve a determinare quali funzioni CUDA e quali versioni di software possono essere eseguite su una specifica GPU.
- **Potenza di Elaborazione:** Misurata in FLOPs, indica il numero massimo di operazioni in virgola mobile (precisione singola o doppia) che una GPU può eseguire al secondo.
- **Interfaccia Bus:** le GPU moderne utilizzano un'interfaccia PCIe per la comunicazione dei dati con i PC. La generazione dell'interfaccia PCIe e il numero di data lanes utilizzabili dalla GPU determinano la velocità massima di trasferimento dei dati.

GPU - LIBRERIE E SVILUPPO

- **Librerie CUDA:** CUBLAS per calcoli di algebra lineare, CUFFT per FFT e NPP per funzioni di base di immagini, video e elaborazione dei segnali.
- **Implementazioni OpenCV e PCL:** OpenCV e Point Cloud Library hanno funzioni abilitate CUDA per accelerare l'elaborazione rispetto alle equivalenti basate su CPU.
- **Tempo di Sviluppo Ridotto:** Sviluppare su GPU è generalmente più rapido rispetto a FPGA e DSP, grazie a strumenti come CUDA e NVidia Nsight che facilitano lo sviluppo e il debug.

GPU - VANTAGGI

- **Costo-Efficienza:** le GPU sono più economiche rispetto a FPGA, offrendo un ottimo rapporto prezzo-potenza di elaborazione.
- **Design:** specifico per l'elaborazione di immagini e video.
- **Sviluppo e Debugging:** Più rapido e semplice rispetto a FPGA.
- **Interfaccia PCIe:** Facilità di uso dell'interfaccia PCIe tra schede GPU e PC host.
- **Avanzamento Tecnologico:** Continuo miglioramento delle tecnologie GPU senza un significativo aumento dei costi.

GPU - SVANTAGGI

- **Consumo di Energia:** Maggiore rispetto a FPGA della stessa classe, non ideale per sistemi sensibili al consumo energetico.
- **Parallelismo Dati Massivo:** Diminuzione delle prestazioni in presenza di attese di dati.
- **Collo di Bottiglia nella Velocità:** Il tempo di trasferimento dati tra PC e GPU può limitare la velocità di elaborazione.
- **Gestione della Memoria Complessa:** Sfide nella scelta e gestione di diversi tipi di memoria per applicazioni ad alte prestazioni.
- **Calcoli a Doppia Precisione:** Teoricamente più lenti rispetto a quelli a singola precisione.
- **Sviluppo di Funzioni Low-Level:** Richiede talvolta l'uso di codici in linguaggio assembly, con minore flessibilità rispetto a FPGA.