



Domande - Sistemi di Elaborazione Accelerata

Ingegneria Informatica M

Matteo Fontolan



Disclaimer & Info

Questo documento é una raccolta **personale** basata sulle lezioni e/o sui materiali didattici forniti dai docenti e/o raccolti negli anni dagli studenti.

! Disclaimer

L'autore non si assume alcuna responsabilità per eventuali errori, omissioni o inesattezze contenuti in questo documento. Il materiale è fornito «così com'è» a solo scopo di supporto allo studio.

Se vuoi **aggiungere** materiale utile o **segnalarmi** errori o correzioni, fallo [qui](#).

Spero che queste risorse ti siano utili.
Buono studio e in bocca al lupo! 🍀

Se ti va di offrirmi una cioccolata 🍫 puoi farlo [qui](#).

Bento

Indice

1. MODULO 1: Architetture CPU, Memoria, Numeri Reali e FPGA	6
1.1. Cache e Memoria CPU	6
1.2. Rappresentazione dei Numeri Reali (Focus E5M2)	6
1.3. FPGA e Accelerazione Hardware	6
1.4. Ottimizzazione Reti Neurali	6
1.5. Altro (SIMD e Sensing)	6
2. MODULO 2: CUDA (Modelli e Ottimizzazione)	6
2.1. Modelli Fondamentali	6
2.2. Modello di Esecuzione e Scheduling	6
2.3. Modello di Memoria e Accessi	7
2.4. Performance e Metriche	7
3. MODULO 1: Architetture, Sensing e Numeri Reali	9
3.1. Approfondimento Cache e Memoria CPU	9
3.2. Rappresentazione e Calcolo (E5M2 e Oltre)	9
3.3. 3D Sensing e Acquisizione	9
3.4. FPGA	9
4. MODULO 2: CUDA (Dettagli Tecnici e Performance)	9
4.1. Architettura SM e Scheduling	9
4.2. Modello SIMT e Warp	10
4.3. Memoria e Trasferimenti (Advanced)	10
4.4. Metriche e Profiling	10



Domande Sistemi di Elaborazione Accelerata

Le seguenti domande si basano sulle domande raccolte dagli studenti che hanno sostenuto l'esame nelle scorse sessioni. Non è garantito che tutte le domande elencate siano effettivamente presenti nell'esame, ma rappresentano un buon punto di partenza per lo studio. Non è altresì garantita la correttezza delle risposte.

Indice

1. MODULO 1: Architetture CPU, Memoria, Numeri Reali e FPGA	6
1.1. Cache e Memoria CPU	6
1.2. Rappresentazione dei Numeri Reali (Focus E5M2)	6
1.3. FPGA e Accelerazione Hardware	6
1.4. Ottimizzazione Reti Neurali	6
1.5. Altro (SIMD e Sensing)	6
2. MODULO 2: CUDA (Modelli e Ottimizzazione)	6
2.1. Modelli Fondamentali	6
2.2. Modello di Esecuzione e Scheduling	6
2.3. Modello di Memoria e Accessi	7
2.4. Performance e Metriche	7
3. MODULO 1: Architetture, Sensing e Numeri Reali	9
3.1. Approfondimento Cache e Memoria CPU	9
3.2. Rappresentazione e Calcolo (E5M2 e Oltre)	9
3.3. 3D Sensing e Acquisizione	9
3.4. FPGA	9
4. MODULO 2: CUDA (Dettagli Tecnici e Performance)	9
4.1. Architettura SM e Scheduling	9
4.2. Modello SIMT e Warp	10
4.3. Memoria e Trasferimenti (Advanced)	10
4.4. Metriche e Profiling	10



1. MODULO 1: Architetture CPU, Memoria, Numeri Reali e FPGA

1.1. Cache e Memoria CPU

- **Architettura della Cache:** Descrivere i vari tipi di cache (Direct Mapped, Set-Associative, Fully Associative), i loro vantaggi/svantaggi e l'architettura interna (latch).
- **Set-Associative Cache:** Spiegare il funzionamento con l'ausilio di uno schema/disegno.
- **Line Fill:** Perché il «line fill» è conveniente, come viene implementato e qual è il focus costruttivo delle memorie che lo rende vantaggioso.

1.2. Rappresentazione dei Numeri Reali (Focus E5M2)

- **Standard E5M2:**
 - Definizione del formato e conversioni (da binario **E5M2** a decimale e viceversa).
 - Identificazione e definizione dei numeri subnormali.
 - Calcolo del più grande numero subnormale e del più piccolo numero normale rappresentabile.
 - Valore dell'esponente nei casi limite.
- **Operazioni e Precisione:**
 - Problematiche relative a somme e moltiplicazioni in virgola mobile.
 - Significato e utilità dei bit di arrotondamento (guard, round, sticky bit).
 - Vantaggi delle operazioni **FMA** (Fused Multiply-Add) e **Multiply-Accumulate** rispetto alle operazioni separate.

1.3. FPGA e Accelerazione Hardware

- **Architettura FPGA:** Descrivere la composizione dei blocchi logici (**CLB**) e il funzionamento della memoria interna (**Block RAM**).

1.4. Ottimizzazione Reti Neurali

- **Riduzione Memory Footprint:** Tecniche per ridurre l'occupazione di memoria dei dati in una rete neurale.
- **Palettizzazione:** Spiegare perché l'uso di meno centroidi riduce il peso della rappresentazione (relazione tra numero di centroidi, dimensione degli indici e tipo di dato).

1.5. Altro (SIMD e Sensing)

- **SIMD:** Definizione e principi del parallelismo SIMD.
- **3D Sensing:** Principi base di acquisizione immagini e sensing 3D.

2. MODULO 2: CUDA (Modelli e Ottimizzazione)

2.1. Modelli Fondamentali

- **Modelli CUDA:** Spiegare nel dettaglio il modello di programmazione (host/device), il modello di esecuzione (grid/block/thread) e il modello di memoria.
- **SIMD vs SIMT:** Differenze concettuali e pratiche tra Single Instruction Multiple Data e Single Instruction Multiple Threads.
- **Architettura SM:** Descrivere l'architettura di un Streaming Multiprocessor (**SM**).

2.2. Modello di Esecuzione e Scheduling

- **Warp Scheduling:**
 - Perché il cambio di contesto (context switch) tra i warp è a costo zero?
 - Relazione tra latenza e scheduling.
- **Divergenza e Sincronizzazione:**
 - Concetto di **Branch Efficiency** (definizione e formula).



- **Independent Thread Scheduling** (ITS) e gestione della divergenza.
- Tipi di sincronizzazione in CUDA (`__syncthreads`, barriere, memoria atomica).
- **Parallelismo Dinamico**: In cosa consiste e come funziona la sincronizzazione in questo contesto.

2.3. Modello di Memoria e Accessi

- **Shared Memory**: Architettura, pattern d'accesso, suddivisione in bank e gestione dei **bank conflict** (con esempi).
- **Global Memory e Coalescing**:
 - Definizione di accesso allineato e accesso coalescente.
 - Importanza del coalescing per le performance.
- **Tipi di Memoria Host-Device**:
 - Differenze tra memoria **Pinned** (vantaggi/svantaggi), **Zero-copy** e **Unified Memory**.
 - Funzionamento effettivo a basso livello (paginazione della memoria, mapping).
- **Strutture Dati**:
 - Memorizzazione di una matrice (Row-major vs Column-major).
 - Formule per la mappatura degli indici globali (1D e 2D).
 - Caso studio: Ottimizzazione della trasposizione di una matrice.

2.4. Performance e Metriche

- **Occupancy**:
 - Definizione di occupancy teorica ed effettiva.
 - Formula della occupancy.
 - Cause e conseguenze di una occupancy troppo bassa o troppo alta.
 - Relazione tra occupancy e «latency hiding».
 - Concetto di «Waves per SM».
- **Roofline Model**:
 - Spiegazione del grafico (unità di misura sugli assi X e Y).
 - Definizione di **Intensità Aritmetica**.
 - Differenza tra scenari «Memory Bound» e «Compute Bound» (cause e possibili soluzioni).

Note per lo studio:

- Le domande riguardanti le formule (occupancy, indici, branch efficiency) richiedono i passaggi matematici.
- Le domande sul formato **E5M2** richiedono esempi di conversione bit-a-bit.
- Il focus sulla cache CPU deve distinguere chiaramente tra il funzionamento hardware (latch) e la logica di gestione (mapping).



Approfondimenti Sistemi di Elaborazione Accelerata

Domande aggiuntive basate sul materiale didattico

Le seguenti domande si basano sulla teoria e sono state aggiunte allo scopo di aiutare lo studio (integrando eventualmente alle precedenti). Non è garantito che tutte le domande elencate siano effettivamente presenti nell'esame, ma rappresentano un buon punto di partenza per lo studio. Non è altresì garantita la correttezza delle risposte.

Indice

1. MODULO 1: Architetture CPU, Memoria, Numeri Reali e FPGA	6
1.1. Cache e Memoria CPU	6
1.2. Rappresentazione dei Numeri Reali (Focus E5M2)	6
1.3. FPGA e Accelerazione Hardware	6
1.4. Ottimizzazione Reti Neurali	6
1.5. Altro (SIMD e Sensing)	6
2. MODULO 2: CUDA (Modelli e Ottimizzazione)	6
2.1. Modelli Fondamentali	6
2.2. Modello di Esecuzione e Scheduling	6
2.3. Modello di Memoria e Accessi	7
2.4. Performance e Metriche	7
3. MODULO 1: Architetture, Sensing e Numeri Reali	9
3.1. Approfondimento Cache e Memoria CPU	9
3.2. Rappresentazione e Calcolo (E5M2 e Oltre)	9
3.3. 3D Sensing e Acquisizione	9
3.4. FPGA	9
4. MODULO 2: CUDA (Dettagli Tecnici e Performance)	9
4.1. Architettura SM e Scheduling	9
4.2. Modello SIMT e Warp	10
4.3. Memoria e Trasferimenti (Advanced)	10
4.4. Metriche e Profiling	10



3. MODULO 1: Architetture, Sensing e Numeri Reali

3.1. Approfondimento Cache e Memoria CPU

- **SRAM vs DRAM:** Spiegare la differenza costruttiva (latch vs condensatore) e come questa influenzi latenza, densità e necessità di refresh.
- **Il problema del «Memory Wall»:** Analizzare il divario prestazionale storico tra frequenza della CPU e velocità della RAM.
- **Le 3 «C» dei Cache Miss:** Definire e distinguere tra Miss **Compulsory**, **Capacity** e **Conflict**.
- **Ottimizzazione del Codice:** Perché l'ordine dei loop (es. IJK vs IKJ) cambia drasticamente le performance in una moltiplicazione tra matrici? In cosa consiste il **Loop Tiling**?

3.2. Rappresentazione e Calcolo (E5M2 e Oltre)

- **ULP (Unit in the Last Place):** Definire il concetto di ULP e spiegare come impatta sull'accuratezza al variare dell'esponente.
- **Rounding Modes:** Spiegare la politica «Round to Even» (Ties to Even) e perché è preferita per evitare bias statistici.
- **Format Trade-offs:** Confrontare FP16, BF16 e TF32. Perché l'intelligenza artificiale preferisce avere più range dinamico (esponente) rispetto alla precisione (mantissa)?
- **Saturazione:** Perché nel calcolo SIMD è spesso preferibile usare l'aritmetica con saturazione rispetto a quella con overflow/wrap-around?

3.3. 3D Sensing e Acquisizione

- **Rolling vs Global Shutter:** Quali artefatti visivi produce un sensore Rolling Shutter in presenza di oggetti in movimento veloce?
- **Event Cameras:** In cosa differiscono radicalmente dalle camere tradizionali e quali sono i vantaggi in termini di latenza e dynamic range?
- **Principi di Profondità:**
 - **Stereo Vision:** Come si passa dalla disparità alla profondità ($Z = \frac{b \cdot f}{d}$)?
 - **Time of Flight (ToF):** Come funziona il calcolo della distanza basato sulla velocità della luce?
- **deBayering:** Descrivere l'operazione di demosaicizzazione. Perché una camera monocromatica è preferibile per la metrologia rispetto a una camera a colori?

3.4. FPGA

- **LUT come Memoria:** Spiegare come un blocco LUT possa essere configurato per fungere da Distributed RAM.
- **Workflow HLS:** Descrivere il passaggio da codice C/C++ a bitstream. Cosa sono i `#pragma` in questo contesto?

4. MODULO 2: CUDA (Dettagli Tecnici e Performance)

4.1. Architettura SM e Scheduling

- **SM Sub-partitioning:** Come sono suddivisi internamente gli SM (es. SMSP) e come vengono assegnati i warp agli scheduler?
- **Independent Thread Scheduling (ITS):** Cosa è cambiato dall'architettura Volta in poi riguardo al Program Counter? Come risolve il problema dei deadlock nei branch divergenti?
- **Costo dei Registri:** Come influisce il numero di registri usati per thread sulla «scelta dei blocchi residenti» (Resource Partitioning)?



4.2. Modello SIMT e Warp

- **Little's Law:** Applicare la legge di Little per stimare il numero di warp necessari a nascondere la latenza aritmetica vs latenza di memoria ($\text{Warp} = \text{Latenza} \times \text{Throughput}$).
- **Branch Efficiency:** Se un warp diverge in 2 rami che eseguono lo stesso numero di istruzioni, qual è l'efficienza massima teorica?
- **Warp Shuffle/Sync:** Quando è davvero necessaria l'istruzione `__syncwarp()` nelle architetture post-Volta?

4.3. Memoria e Trasferimenti (Advanced)

- **PCIe Bottleneck:** Perché aggregare piccoli trasferimenti in un unico grande buffer migliora la bandwidth effettiva?
- **Unified Memory (UM):** Spiegare il ruolo del **Page Migration Engine**. In cosa differisce il comportamento pre-Pascal e post-Pascal riguardo alla mutua esclusione CPU-GPU?
- **Pinned Memory:** Spiegare il concetto di «copia di staging» del driver CUDA quando si usa memoria paginabile.
- **Bank Conflicts:** Se 32 thread accedono allo stesso indirizzo in Shared Memory, si verifica un bank conflict? (Distinzione tra **Conflict** e **Broadcast**).

4.4. Metriche e Profiling

- **Nsight Systems vs Nsight Compute:** Quale strumento useresti per analizzare la timeline dei trasferimenti e quale per analizzare i conflitti nei banchi della shared memory?
- **Arithmetic Intensity:** Come si calcola per un kernel di somma vettoriale? Perché è quasi sempre memory-bound?
- **Occupancy:** Perché un'occupancy del 100% non garantisce necessariamente le performance massime?