Pipeline HDC su Cyclone V: binding, bundling e recovery con parallelismo 256-way

Panoramica

Questo documento descrive una pipeline di Hyperdimensional Computing (HDC) mappata su un datapath Cyclone V, con enfasi su binding, bundling, recovery, accessi alle memorie M10K, controllo tramite FSM e parallelismo a 256 lane. Le operazioni fondamentali dell'HDC includono il binding, che combina ipervettori tramite operazioni elemento per elemento come XOR nei casi binari, il bundling, che aggrega più ipervettori tramite somma o majority vote, e il recovery, che effettua un confronto associativo con prototipi o codebook memorizzati.[cite:8] [cite:9] [cite:22]

L'uso di un FPGA Cyclone V è adatto a questo tipo di pipeline perché il dispositivo integra logica programmabile, DSP ed embedded memory blocks M10K, che possono essere organizzati in banchi per sostenere accessi paralleli e latenze regolari.[cite:18] [cite:30] [cite:33]

Operazioni HDC di base

Binding

Il binding costruisce rappresentazioni composte combinando due o più ipervettori; nei sistemi HDC binari questa operazione è spesso implementata con XOR, mentre in altre formulazioni può corrispondere a un prodotto componente per componente.[cite:8] [cite:28] [cite:34] Questa proprietà rende il risultato statisticamente dissimile dagli ingressi originali ma ancora utile per rappresentare associazioni strutturate, come simbolo-posizione o chiave-valore.[cite:8] [cite:9]

Bundling

Il bundling aggrega più ipervettori in una memoria prototipale comune, tipicamente attraverso somma seguita da thresholding oppure majority vote, preservando la similarità con i contributi principali.[cite:8] [cite:22] [cite:31] In hardware, questa fase richiede accumulatori, logica di riduzione e una politica di aggiornamento che mantenga coerenti i risultati tra lane parallele e memoria intermedia.[cite:22] [cite:31] [cite:34]

Recovery

Il recovery prende un ipervettore query e lo confronta con un insieme di prototipi o vettori memorizzati, usando misure di similarità come overlap bitwise, distanza di Hamming o score equivalenti implementati in modo distribuito.[cite:9] [cite:22] [cite:31] Il risultato finale è l'identificazione del miglior match, che rappresenta la fase di lettura associativa dell'acceleratore HDC.[cite:9] [cite:28]

Mappatura sul datapath Cyclone V

La pipeline è naturalmente esprimibile come datapath FPGA perché ogni dimensione dell'ipervettore può essere trattata come un'unità regolare e ripetibile; questo favorisce una decomposizione in lane parallele con controllo centralizzato.[cite:22] [cite:31] [cite:34] Sul Cyclone V, la combinazione di ALM, registri di pipeline, DSP e blocchi M10K permette di separare la parte computazionale dalla parte di storage, mantenendo una struttura a flusso adatta a throughput elevato.[cite:18] [cite:30] [cite:33]

In un'implementazione di questo tipo, il datapath legge ipervettori elementari o intermedi dalle M10K, li instrada verso le lane di binding, accumula i risultati nella fase di bundling e poi lancia una fase di recovery contro una memoria di prototipi. La scelta di organizzare la computazione in più stadi espliciti riduce il fan-out globale e consente di chiudere il timing con più facilità rispetto a una struttura monolitica.[cite:22] [cite:31] [cite:33]

FSM di controllo

Una FSM dedicata coordina la pipeline passando attraverso stati del tipo LOAD, BIND, BUNDLE, REDUCE, RECOVER e OUTPUT, così da sincronizzare gli accessi di memoria e l'attivazione delle lane.[cite:22] [cite:31] Questo schema è coerente con le architetture HDC ad alte prestazioni, nelle quali il controllo esplicito consente di separare la latenza delle memorie dalla latenza di elaborazione e di riusare gli stessi blocchi in più fasi senza conflitti funzionali.[cite:22] [cite:34]

- LOAD: preleva item memory, positional memory, codebook o stato intermedio dalle M10K e li allinea ai registri d'ingresso delle lane.[cite:30] [cite:33]
- BIND: abilita le 256 lane affinché eseguano la combinazione elemento per elemento sugli ipervettori correnti.[cite:22] [cite:28]
- BUNDLE: aggiorna accumulatori o strutture di voto per generare un ipervettore composito.[cite:22] [cite:31]
- REDUCE: combina risultati parziali tra gruppi di lane o tra passate successive.[cite:31] [cite:34]
- RECOVER: confronta il vettore finale con prototipi memorizzati e calcola il best match. [cite:9] [cite:22]
- OUTPUT: registra la classe, il simbolo o la chiave recuperata e rende disponibile il risultato all'interfaccia esterna.[cite:22] [cite:31]

Parallelismo a 256 lane

Con parallelismo 256-way, il datapath elabora 256 componenti dell'ipervettore per ciclo, oppure 256 sotto-operazioni indipendenti per ciclo, a seconda di come il vettore è stato impacchettato e distribuito sulle lane.[cite:22] [cite:31] Questo tipo di parallelismo è particolarmente adatto all'HDC perché le operazioni base sono quasi sempre indipendenti tra dimensioni prima della fase di riduzione, quindi la scalabilità è elevata fino a quando memoria e routing rimangono sostenibili.[cite:22] [cite:28] [cite:34]

Se la dimensionalità dell'ipervettore è pari a D, il numero di cicli necessari per una passata di binding o similarity è approssimativamente $\lceil D/256 \rceil$, al netto di latenza di pipeline e operazioni di caricamento.[cite:22] [cite:31] Questo significa che l'aumento del numero di lane agisce direttamente sul throughput, ma sposta il collo di bottiglia verso la banda delle memorie M10K e verso la riduzione dei parziali.[cite:22] [cite:30]

Organizzazione delle M10K

I blocchi M10K del Cyclone V costituiscono il livello principale di memoria on-chip per item memory, positional encodings, prototipi, accumulatori intermedi e buffer di output.[cite:16] [cite:30] [cite:33] Poiché ogni M10K ha limiti di larghezza di porta e numero di accessi per ciclo, una pipeline a 256 lane richiede una forte bancarizzazione della memoria per evitare conflitti strutturali.[cite:30] [cite:33]

In pratica, i dati vengono suddivisi tra più banche M10K, spesso allineate con gruppi di lane, così che ogni ciclo possa servire un insieme di richieste deterministiche. Questa strategia rende possibile sostenere il flusso LOAD–BIND–BUNDLE–RECOVER senza serializzare gli accessi su un singolo blocco memoria.[cite:22] [cite:30] [cite:33]

Gli accessi M10K possono essere distinti in tre categorie principali:

Tipo di accesso	Funzione	Impatto architetturale
Lettura di codebook/item memory	Fornisce gli ipervettori base per binding e recovery [cite:8][cite:30]	Richiede layout regolare e distribuzione per lane [cite:30][cite:33]
Lettura/scrittura di accumulatori	Supporta bundling e riduzione [cite:22][cite:31]	Introduce dipendenze temporali e necessità di registrazione [cite:22][cite:31]
Lettura di prototipi	Alimenta la similarity search nel recovery [cite:9][cite:22]	Può diventare il collo di bottiglia principale [cite:22][cite:30]

Descrizione completa della pipeline

Fase 1: caricamento

La pipeline inizia con il caricamento dei vettori sorgente, delle strutture di stato e degli eventuali prototipi necessari per lo step corrente. In questa fase, la FSM produce indirizzi deterministici verso le M10K e popola i registri d'ingresso delle lane, così che il binding possa partire senza stall non necessari.[cite:30] [cite:33]

Fase 2: binding parallelo

Una volta caricati gli operandi, le 256 lane eseguono il binding in parallelo. Se la rappresentazione è binaria, ogni lane implementa una logica XOR su un sottoinsieme delle dimensioni; se la rappresentazione è bipolare o multivalore, la lane applica la regola corrispondente prevista dal formalismo HDC adottato.[cite:8] [cite:28] [cite:34]

Questa fase è regolare e facilmente pipelineable, perché ogni lane è quasi indipendente dalle altre. Il risultato è un insieme di parziali bound che può essere inoltrato direttamente a registri

di uscita o a memorie intermedie per la fase di aggregazione.[cite:22] [cite:31]

Fase 3: bundling

Nel bundling, i risultati bound vengono accumulati per costruire una rappresentazione composita. A livello hardware, ciò significa sommare o votare i contributi lane per lane e poi combinarli in una forma finale che preservi la similarità con i pattern dominanti.[cite:8] [cite:22] [cite:31]

Poiché 256 lane generano una grande quantità di stato parziale, la pipeline deve usare accumulatori locali, eventuali gruppi di riduzione e scritture sincronizzate verso M10K. Questa è la fase in cui l'architettura decide buona parte del compromesso fra area, frequenza e throughput.[cite:22] [cite:31] [cite:34]

Fase 4: riduzione

La riduzione raccoglie i risultati parziali della bundling e li organizza in un vettore finale o in una serie di prototipi aggiornati. In architetture ad alto parallelismo, la riduzione avviene spesso come albero o come sequenza di riduzioni parziali, per limitare il fan-in in un singolo stadio.[cite:22] [cite:31] [cite:34]

Fase 5: recovery

Il vettore finale viene poi usato come query per la memoria associativa. Ogni lane partecipa al calcolo della similarità contro vettori di riferimento, producendo contributi parziali che vengono a loro volta ridotti per ottenere uno score finale per ogni candidato.[cite:9] [cite:22] [cite:31]

Il recovery conclude la pipeline selezionando il miglior match, cioè il prototipo più vicino alla query secondo la metrica implementata. Questa fase rappresenta il corrispondente hardware dell'inferenza o del richiamo associativo in HDC.[cite:9] [cite:28]

Considerazioni per una pubblicazione

Dal punto di vista accademico, i messaggi principali da mettere in evidenza sono quattro. Primo, HDC è intrinsecamente adatto al parallelismo massivo perché le operazioni elemento per elemento dominano una parte rilevante della computazione.[cite:22] [cite:31] Secondo, una mappatura su Cyclone V con memorie M10K bancarizzate permette di tradurre questo parallelismo in throughput effettivo, a patto di controllare bene il traffico di memoria.[cite:30] [cite:33] Terzo, la FSM esplicita è un vantaggio architetturale, non solo un dettaglio implementativo, perché consente una schedulazione riproducibile delle fasi della pipeline.[cite:22] [cite:34] Quarto, il parallelismo 256-way rende naturale la scomposizione dell'ipervettore in fold, ma impone una progettazione attenta della riduzione e dei percorsi di lettura dei prototipi.[cite:22] [cite:31]

Testo pronto per figura

Una formulazione adatta alla didascalia di una figura può essere la seguente: “Architettura HDC su Cyclone V con pipeline a 256 lane per binding, bundling e recovery. Il datapath usa M10K bancarizzate per item memory, prototipi e buffer intermedi, mentre una FSM coordina gli stati LOAD, BIND, BUNDLE, REDUCE, RECOVER e OUTPUT per sostenere throughput elevato con accessi di memoria deterministici.” [cite:22] [cite:30] [cite:33]