Proposal: Reducing DRAM Traffic and Enabling Cross-Engine Pipelining in Neural Video Accelerators via Group-Synchronized Forwarding

Team Members

- **Name:** Snehal Sharma(snehal.sharma@iiitb.ac.in), Nipun Verma(nipun.verma@iiitb.ac.in)
- **Status:** Student
- **Institute:** IIIT Banglaore

Abstract

Neural Video Compression (NVC) offers superior coding efficiency but imposes severe bandwidth constraints on hardware. The baseline "VCNPU" architecture relies on a "Scatter-Gather" dataflow where intermediate Motion Synthesis results ($\hat{O}_t$) are written to off-chip DRAM and read back by the Deformable Prediction Module (DPM). This round-trip consumes approximately 18 GB/s of bandwidth for 1080p video at 60 FPS, creating a "memory wall" and preventing parallel execution.

This project proposes a micro-architectural enhancement: the **Group-Synchronized Tile Forwarding** engine. By exploiting the specific 4-row "Group" processing granularity of the underlying accelerator, we replace the DRAM round-trip with an on-chip, multi-banked SRAM FIFO. This enables the DPM to consume motion vectors immediately upon generation. Our design includes a "Split-Prefetcher" to decouple motion vector availability from Reference Frame (F_{t-1}) fetches. Analytical modeling confirms this reduces off-chip traffic by ≈ 300 MB/frame and enables deterministic overlap of the SFTM and DPM engines. *Link to VCNPU Paper*

Project Description & Impact

1. Technical Motivation: The "Memory Wall" in NVC

The decoding process involves two distinct, bandwidth-heavy stages:

1. **Motion Synthesis (SFTM):** The Sparse Fast Transform Module generates motion feature maps ($\hat{O}_t$).
2. **Deformable Compensation (DPM):** The DfConv Processing Module uses $\hat{O}_t$ to compute offsets and sample from a Reference Frame (F_{t-1}).

In the baseline architecture, $\hat{O}_t$ is fully serialized to external memory via a "Scatter" interface before the DPM can "Gather" it.
The Quantitative Cost: Based on the standard configuration of $N = 36$ channels and 16-bit precision, a single 1080p frame (1920×1080) generates:

$$1920 \times 1080 \times 36 \text{ ch} \times 2 \text{ Bytes} \approx 149.3 \text{ MB}$$

A mandatory Write $\rightarrow$ Read round-trip doubles this to ≈ 298.6 MB/frame. At 60 FPS, this wastes **17.9 GB/s** of bandwidth on transient data.

2. Proposed Architecture

We propose a cohesive dataflow modification comprising three components:

A. On-Chip Group-Alignment FIFO

To eliminate the DRAM traffic, we introduce a dedicated SRAM buffer between the SFTM and DPM.

- **Sizing Strategy:** The SFTM produces output in "Groups" of 4 rows ($m_d = 4$). To ensure deadlock-free execution under jitter, we provision a conservative buffer depth of 2 Groups (8 rows).
- **Memory Footprint:**

  $$1920 \text{ cols} \times 8 \text{ rows} \times 36 \text{ ch} \times 2 \text{ B} \approx 1.1 \text{ MB}$$

  While 1.1 MB is significant, it is feasible in 28nm technology and serves as a justifiable trade-off for saving 18 GB/s of DRAM bandwidth. We will also explore an *Aggressive Tiling Mode* to reduce this to $\approx$ 32 KB by restricting the active wavefront to narrow column tiles, provided flow control allows.
- **Banking:** The FIFO uses multi-banked SRAM to prevent conflicts between the SFTM's write pattern and the DPM's variable-offset read pattern.

B. Split-Prefetcher Logic

A critical risk is that the DPM requires two inputs: the motion vector $\hat{O}_t$ (now on-chip) and the Reference Frame F_{t-1} (still off-chip). We propose a **Split-Prefetcher**:

- It monitors the "Group Done" signals from the SFTM.
- Upon completion of Group N, it immediately issues DRAM Read requests for the corresponding spatial region of F_{t-1}.
- This ensures that reference pixels are available exactly when the DPM reads the motion vector from the FIFO, masking DRAM latency.

C. Credit-Based Flow Control

To enable safe pipelining, we replace the serialized execution model with a handshake FSM.

- **Mechanism:** The FIFO maintains a "Credit" count of valid Groups. The DPM is triggered only when Credits > 0.
- **Latency Masking:** By allowing the SFTM to run ahead of the DPM, we maximize hardware utilization.

3. Impact and Validation

- **Energy Efficiency:** DRAM access consumes orders of magnitude more energy per bit (approx. 100s of pJ) compared to local SRAM access. Eliminating 17.9 GB/s of traffic will significantly lower the system's thermal design power (TDP).
- **Performance:** The pipeline overlap allows the total frame time to approach $\max(T_{SFTM}, T_{DPM})$ rather than $T_{SFTM} + T_{DPM}$, potentially yielding a 10–20% throughput increase.
- **Risk Mitigation:** A "Bypass Mode" will be implemented to route data via the original Scatter/Gather path if the FIFO overflows or for debugging purposes.

4. 4-Week Execution Plan

- **Week 1: Micro-Architecture & Profiling.** Define the exact banking map for the 1.1 MB FIFO to match the SFTM's 4-row Group output. Develop the Split-Prefetcher specification.
- **Week 2: RTL Implementation.** Implement the SRAM FIFO wrapper and the Credit-Based Flow Control FSM in Verilog. Design the arbiter for the Split-Prefetcher.
- **Week 3: Integration & Verification.** Integrate the module between SFTM and DPM stubs. Run test vectors to verify that Group N in the DPM receives numerically identical data to the baseline DRAM path.
- **Week 4: Synthesis & Trade-off Analysis.** Synthesize the design (target 400 MHz) to measure the area cost of the 1.1 MB SRAM. Compare this area penalty against the energy savings from the bandwidth reduction.

VLSI Design Tools available at Home Institute

- **Front-end Design:** Verilog HDL, Xilinx Vivado.
- **Simulation:** Cadence Xcelium / Mentor Graphics QuestaSim.
- **Synthesis & P&R:** Synopsys Design Compiler, Cadence Genus/Innovus.
- **Physical Verification:** Cadence Virtuoso.