Tactile Sensing Computation ASIC with Quantized Machine Learning Model on 180nm Silicon

Students: Kearnan Bishop, Qiancheng Li, Maohua Nie, Luke Valerio, Rachel Walland

Advisors: Ang Li, Yiyue Luo, Devin Murphy

Tactile Sensing Compute Chip: Introduction to and Operation of System

The compute chip is a deep learning accelerator (DLA) designed for use alongside a separate control chip in a BSG ring structure. Instructions and data are communicated to the compute chip from the preceding control (or compute) chip using the BSG link protocol, and results are streamed back over the same ring. Model weight matrices are stored in an external QSPI flash memory and are loaded through an on-chip Wishbone master. The control chip is the system master, handling all kickoffs, configuring the compute chip via BSG link, sequencing the reset via FSM, and driving the DLA through ring packets.

COMPUTE CHIP INFORMATION

- **Total area: 2.25 mm²**
- **Core utilization:** 85.28**%**
- **Setup slack: +0.294 ns**
- **Hold slack: +0.102ns**
- **Theoretical max frequency: 50.7 MHz (target 50 MHz)**

MODEL DESCRIPTION

- Single-layer, unidirectional GRU: processes sequences in single forward pass, enables real-time inference
- Quantization-Aware Training (QAT): simulates low-precision (int8) arithmetic during training, weights adapt to quantization effects
- Suitable for edge devices and low-latency applications

CHIP ARCHITECTURE

A basic summary of the major modules in the chip is as follows:

- **dla.sv** is the top-level chip module. It wires together the control path (ctrl.sv) and the datapath as well as memories – two vector memories for input memories A and B, and a number of weight memories, dot product modules, special function units, and accumulation memories according to the number of dot product lanes.
- **ctrl.sv** corresponds to the control path, written entirely inside of a finite state machine. It contains an instruction* buffer (single-write, single read SRAM) an embedded register file, and a wishbone master for communication with off-chip QSPI.

***Instructions are 40 bits, with the below format:**

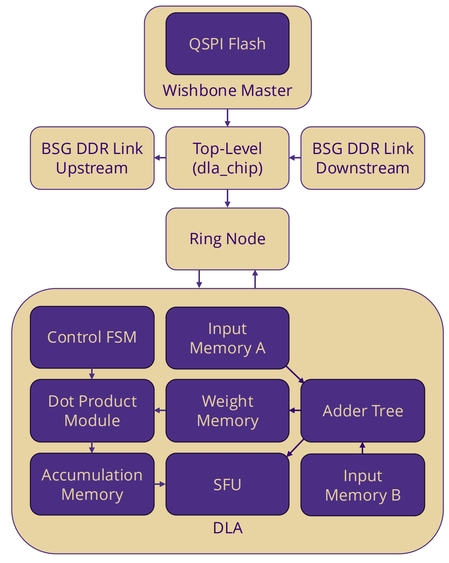
[39:35] Opcode
[34] mode-0
[33] mode-1
[32:30] reg-id-0
[29:27] reg-id-1
[26:0] per-opcode payload

- **dp.sv** is the dot product lane.
- **at.sv** is an adder tree, pipelined according to a number of stages set as a parameter in the system.
- **sfu.sv** is the special function unit, a per-dot product combinational activation function unit. It has zero-cycle latency, writing its result to the accumulation memory in the same cycle it receives the data. This module is also where requantization occurs if it is necessary.
- **Ring_node.sv, ring_ctrl.sv,** and **ring_pkg** are examples of wrapper modules that help with forwarding in the chip; for multiple DLA chips (see Future Applications), ring_node.sv handles forwarding to the correct DLA chip for computation.

VERIFICATION

The following sequence is used for testing the chip's functionality:

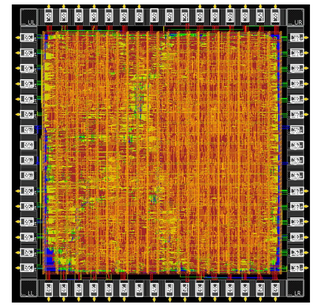
- Initialization: Weights are loaded into the SPI flash memory. Compiled control firmware is transferred to the SERV instruction RAM, and runtime data is transferred to the SERV data RAM. Input frames and expected values are stored in the appropriate testbench arrays.
- Firmware boot phase: This is triggered when the core calibration is complete. The chip is programmed via BSG link, and 207 instruction packets are sent to the chip's instruction buffer. Upstream DMA is armed for 8-bit packed transfer.
- Per-frame loop: For every input frame, the result buffer is reset, the DMA is kicked to stream ADC bytes out of the ring, and the chip program is kicked off for launch. Ping pong buffering is used for read and write.
- Execution: Once kicked off, the chip runs independently. ADC bytes arrive in input memory A, the PC starts, and the chip executes its loaded program. The chip issues instructions to fetch weights from the QSPI flash as needed, and the compute pipeline runs. The results land in the accumulation memory as int8 logits.
- Result check: the testbench validates the output by reading the logits out of the result buffer and flagging any mismatches.



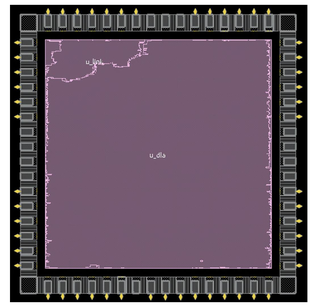
A block diagram of the compute chip architecture.

CHIP LAYOUT

The below figures are snapshots of the current state of the compute chip's physical design. LVS is clean and there are minimal DRC errors.



Physical layout view



Amoeba layout view

POST-SILICON VERIFICATION

Verification will take the form of physical testing through an on-chip scan chain and comparing the results of the virtual testbench with an equivalent loaded through trace files onto the chip.

FUTURE APPLICATIONS

- The control chip and compute chip architectures are designed such that in the future, 2 or 3 DLA chips may function in a pipeline. Each chip would hold its own layer's weights in its QSPI flash, yielding a faster reload.
- The architectural ceiling for the setup is as many as 63 DLA chips; however, practical limitations lower this ceiling and further refinement must be completed for more than 2-3 chips to be possible.

SPONSORS