NeCTAr and RASoC: Tale of Two Class SoCs for Language Model Interference and Robotics in Intel 16

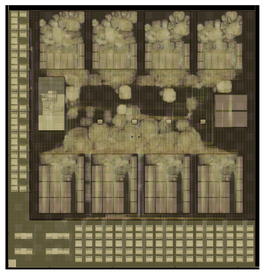
Viansa Schmulbach, Jason Kim, Ethan Gao, Nikhil Jha, Ethan Wu, Oliver Yu, Ben Oliveau, Xiangwei Kong, Brendan Roberts, Connor McMahon, Lixiang Yin, Vamber Yang, Brendan Brenner, George Moujaes, Boyu Hao, Lucy Revina, Kevin Anderson, Bryan Ngo, Yufeng Chi, Hongyi Huang, Reza Sajadiany, Raghav Gupta, Ella Schwarz, Jennifer Zhou, Ken Ho, Jerry Zhao, Anita Flynn, Borivoje Nikolić

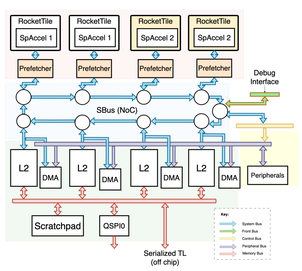
Overview

- Two 16nm heterogeneous system-on-chip designs developed from concept to chip tapeout in <15 weeks by a team of mostly undergraduate students with no prior chip design experience
- Both chips contain: 64b-wide TileLink unidirectional torus NoC, on-chip 64KB scratchpad, 256KB of L2$ in four banks, and an on-chip 10MHz–2.5GHz Intel-provided PLL
- On-chip peripherals include: 52x GPIO, 3x UART, 2x XIP QSPI Flash/PSRAM, 6x PWM, 2x I2C, 1x 4ch. QSPI, and Serial Tilelink

NeCTAr: Near-Cache Transformer Accelerator

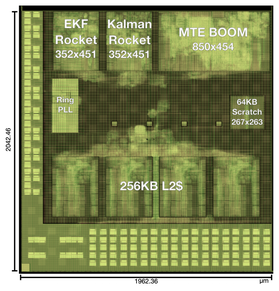
- **RISC-V SoC optimized for ML applications in Intel 16**, accelerators include:
  - Near-memory compute engines with MAC operations co-located with one L2$ bank
  - Two CPU-coupled sparse matrix accelerator versions with L1 and L2 cache access
  - L2 data cache best-offset prefetchers

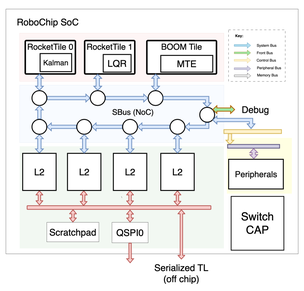




RASoC: Robotics Application System-on-Chip

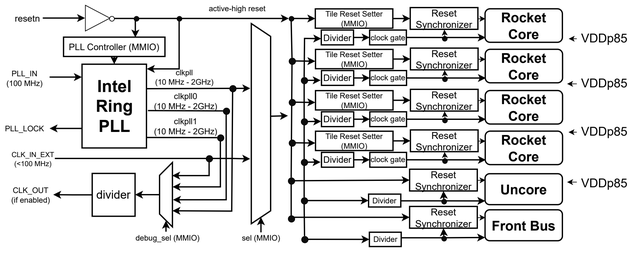
- **RISC-V SoC suitable for controlling robotics systems in Intel 16**
- Integrates custom Kalman filter and LQR on two large Rocket tiles
- Improves software security by adding a new Memory Tagging Extension to a medium BOOM tile
- Includes an internal, digitally controlled switched capacitor voltage regulator





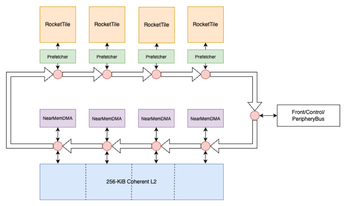
Clocking

- **All new clock system based around Intel PLL IP**
  - Supports 10 MHz to 2.2 GHz. PLL is responsive and able to run near its maximum frequency at 2.13 GHz.
  - PLL configurable through MMIO
- Externally controlled debug clock mux is working as expected
- Clock dividers working as expected

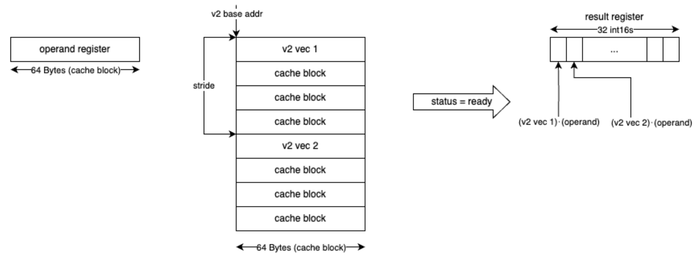


Memory Subsystem

- 256 KiB, 8-way set associative banked L2$ + **larger 64KiB Scratchpad**
- Network on Chip (NoC): unidirectional ring, 64 bit system bus width
- **1-to-1 correspondence of L2$ with DMA engine to reduce SBus traffic**



Near-Memory Compute Engine (NMCE) Architecture



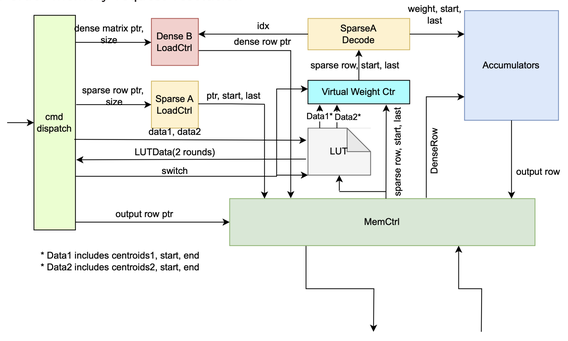
- **Allow for concurrent operations to different addresses and accelerates cache block size memcpy and multiply accumulate.**
- memcpy and MAC accelerator physically placed near L2 cache banks in uncore
- MAC operation computes up to 32 dot products (int16) of 64B input vectors (int8)
- Communicates with core through four memory-mapped control registers: (1) `v1Reg`, a 64B register, (2) `v2addr`, a 64b address, (3) a `stride` in bytes for memory accesses and (4) `count`, the number of requested operations (up to 32)
  - Computes `count` dot products between int8 vectors and writes each saturated int16 result to a 64B register
  - 64B multiply-accumulate is computed and written in a single cycle

Test	Naive	Accel	4 DMA	Speedup
64 Byte Block Memcpy	4	10	N/A	.4x
128 KB Block Memcpy	56405	42756	32162	1.75x
1 MiB Block Memcpy	837970	608005	456652	1.84x
8 element vector product	76	94	N/A	.81x
8x8 Matrix vector product	952	92	110	9.66x

Table 1: NearMem maximum speedup (Accel/4 DMA vs. Naive)

Sparse Accelerator Architecture

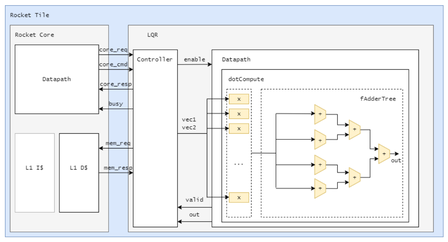
- **Directly accesses L2$ for increased memory bandwidth**, using TileLink protocol for cache coherence and supports virtual address translation
- Uses a reservation station for out-of-order memory request resolution
- **Codebook module for sparse row quantization and pruning**
  - LUT for decompressing 4b sparse weight to 8b sparse weight
- **Signed int support**
  - 17-bit pipeline with signed 8-bit operands
  - Intermediate saturation at ±49k



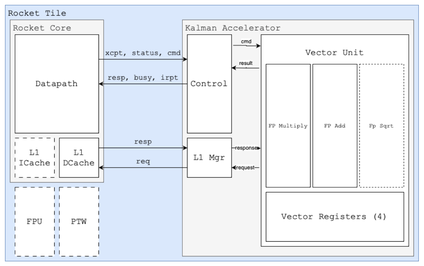
Linear-Quadratic Regulator (LQR) Accelerator

- **Solve LQR state-space equations** implemented by a 32b floating point matrix multiplication accelerator
- ∼2.2× solve speedup with 23% increase in Rocket area

Test	Simulation Speedup
vec[6]	35.15%
vec[31]	30.57%
vec[32]	11.42%
mat[8x10] ∗ vec[10]	54.23%



Extended Kalman Filter Accelerator



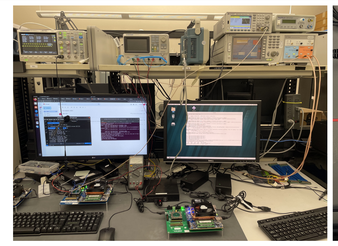
- **Estimate position by fusing information from multiple sensors**
- EKF supports modeling non-linear inputs like IMU based acceleration data
- Implemented as a 32-bit floating point vector engine
- Overall ∼2.6× speedup over pure software solution of EKF on Rocket

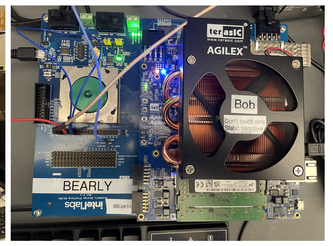
Test	Simulation Speedup
addition	46.41%
multiplication	46.43%
division	23.83%
gps_ekf	38.40%

Agile Chip Design Methodology

- SoCs implemented in Chisel in the Chipyard framework
- Hammer physical design tool used to wrap around commercial EDA tools, extended for new PDK
- Used Chipyard RTL generators, including Rocket core and Constellation NoC generator
- Chipyard support for integrating accelerators
  - RoCC protocol + custom ISA extensions
  - Memory-mapped register generator for DMA accelerators
- Test SoCs generated in less than 4 weeks

Bring-up and Verification

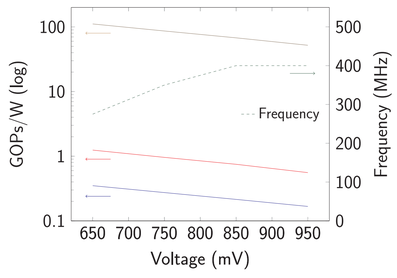




- The 2mm x 2mm SoCs in Intel 16 are two in a 2x2 tile array in an interposer package designed to integrate easily with a custom PCB and FPGA setup for emulation and bring-up
- Bringup Infrastructure consists of three main pillars:
  - **Baremetal IDE:**
    - Collection of build scripts and board support files to allow easy creation of baremetal programs
    - Both existing and new Hardware Abstraction Layer and Programming Interfaces for hardware, eg: DMA Engine, GPIO, etc
  - **Apollo Agilex FPGA**:
    - UART-TSI Host for upload binaries
    - DDR4 SDRAM External Memory Interface with AXI port
    - SPI Flash Emulator
    - SignalTap Logic Analyzer for Peripherals
  - **Heather Lake Bringup PCB:**
    - Intel provided PCB w/ ZIF Socket & FMC+ Connector for FPGA Connection
    - Contains both internal clocks/power supplies and taps
    - Breaks out SwitchCap, I2C, Debug Ports

Measurements and Results

- NeCTAr is evaluated using the FPGA + chip setup shown above
- At the nominal voltage, the maximum clock frequency is 400MHz, and the highest efficiency is 132 GOPs/W
  - Matrix-vector multiplication kernel shmoo plot shown below
- NeCTAr runs a 1.7M ReLU-Llama trained on the TinyStories dataset, performance table shown on the right
  - The ReLU-Llama model is modified to induce activation sparsity to halve weight reads from off-chip
- Bottlenecked by external memory reads over 1-bit memory bus



Matrix Multiplication

Single-core	Multi-core	NMCE
0.350 GOPs/W	1.24 GOPs/W	132 GOPs/W
18.7 MOPs	56.6 MOPs	6.02 GOPs

1.7B LLAMA

Single-core	Multi-core	NMCE
39.0 infs/s/W	40.0 infs/s/W	45.4 infs/s/W
1.19 infs/s	1.25 infs/s	1.28 infs/s

Future Work

- Widened Serial-TL port for accessing off-chip memory
- Support for additional number representations (int16, float)
- Utilize Chipyard updates with new Multi-ChipTop flows for building and simulating chip + FPGA

Acknowledgements

This chip is one of three taped out by the Spring 2023 EE290C/EE194 class, consisting of over 36 students on the BearlyML23 and RoboChip teams. Thank you to:

- Intel University Shuttle Program for donating fabrication and packaging
- NSF CCRI ENS Chipyard Award #2016662
- Scalable Asymmetric Lifecycle Engagement (SCALE)
- Apple for supporting integrated circuit engineering classes at UC Berkeley EECS department through Apple's New Silicon Initiative program.

University of California, Berkeley
Department of Electrical Engineering & Computer Sciences