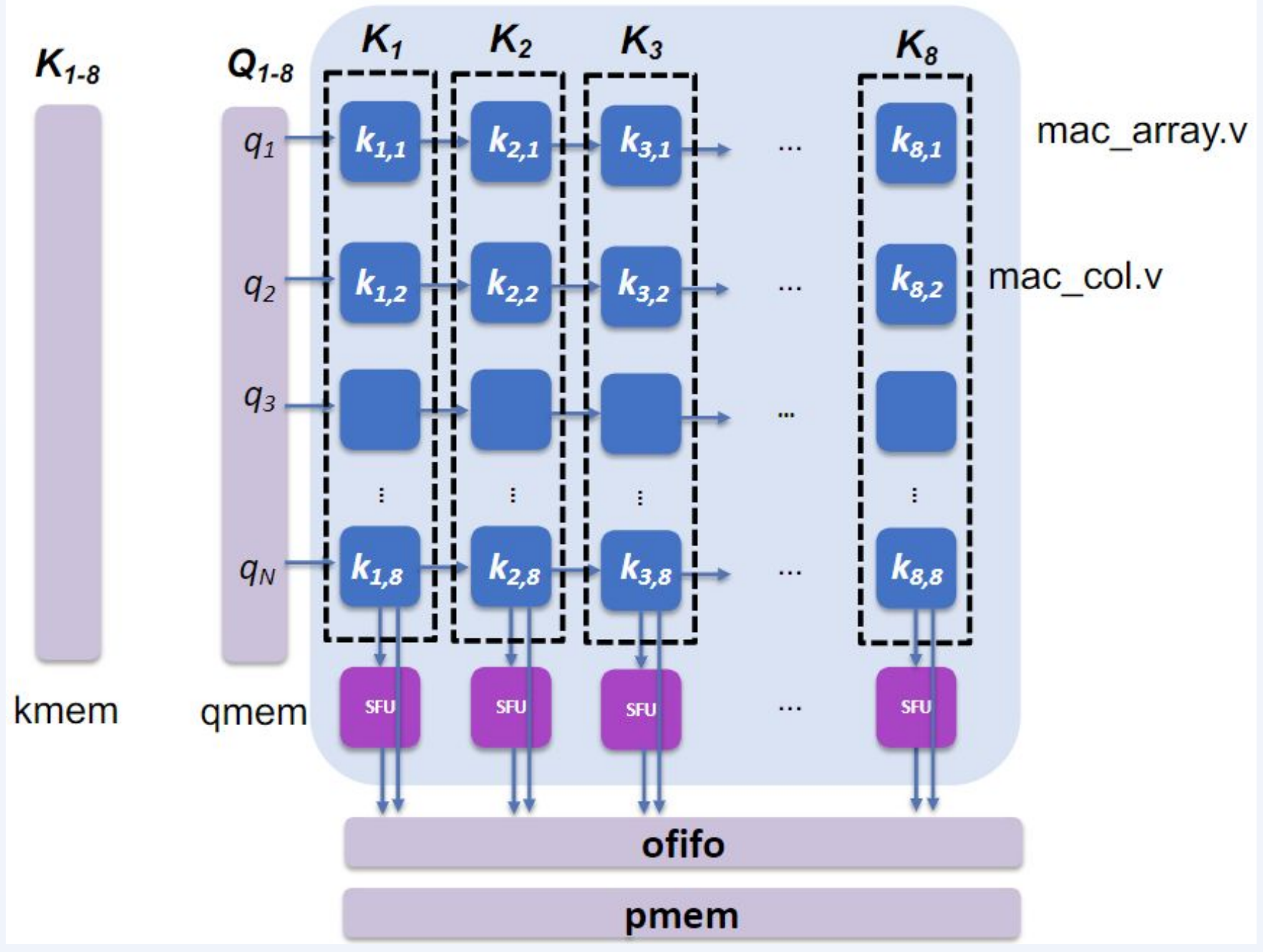
Dual Core Machine Learning Accelerator for Attention Mechanism

ECE260B - Group 9

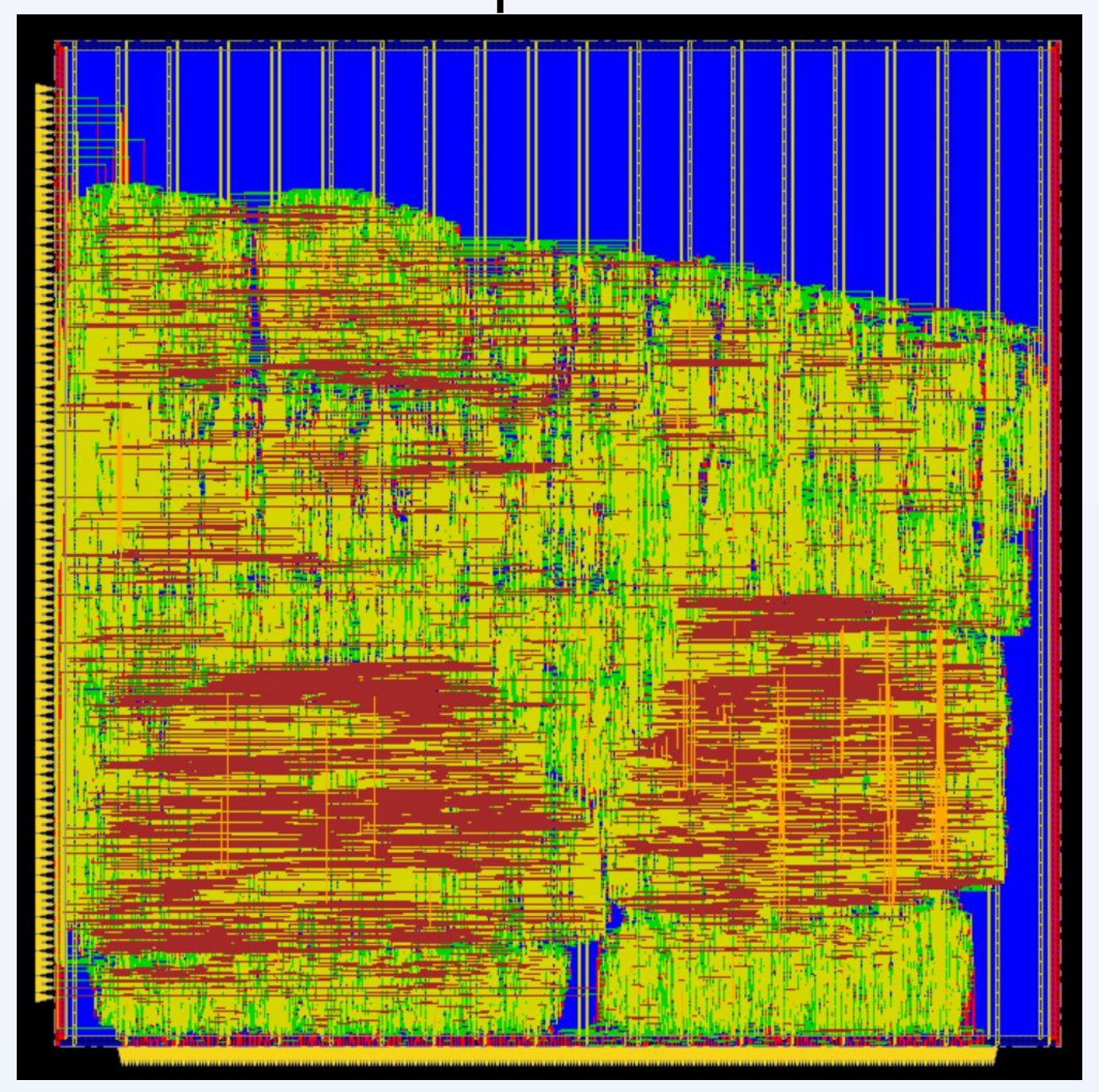
By Zhongkang Fang, Aadhar Sharma, Francis Lu, and John Ruffy

Motivation

The objective is to design a well optimized dual core AI accelerator, in which the architecture processes data in a pipeline of computations performed in parallel. Thus, resulting in a better throughput and lower latency. Due to this architecture's exceptional computational efficiency, it is a commonly used for signal processing, image processing, and complex matrix computations in the field of AI.



*PNR before optimization



Reference.
UCSD ECE260B WI23 Project Description ppt by Mingu Kang

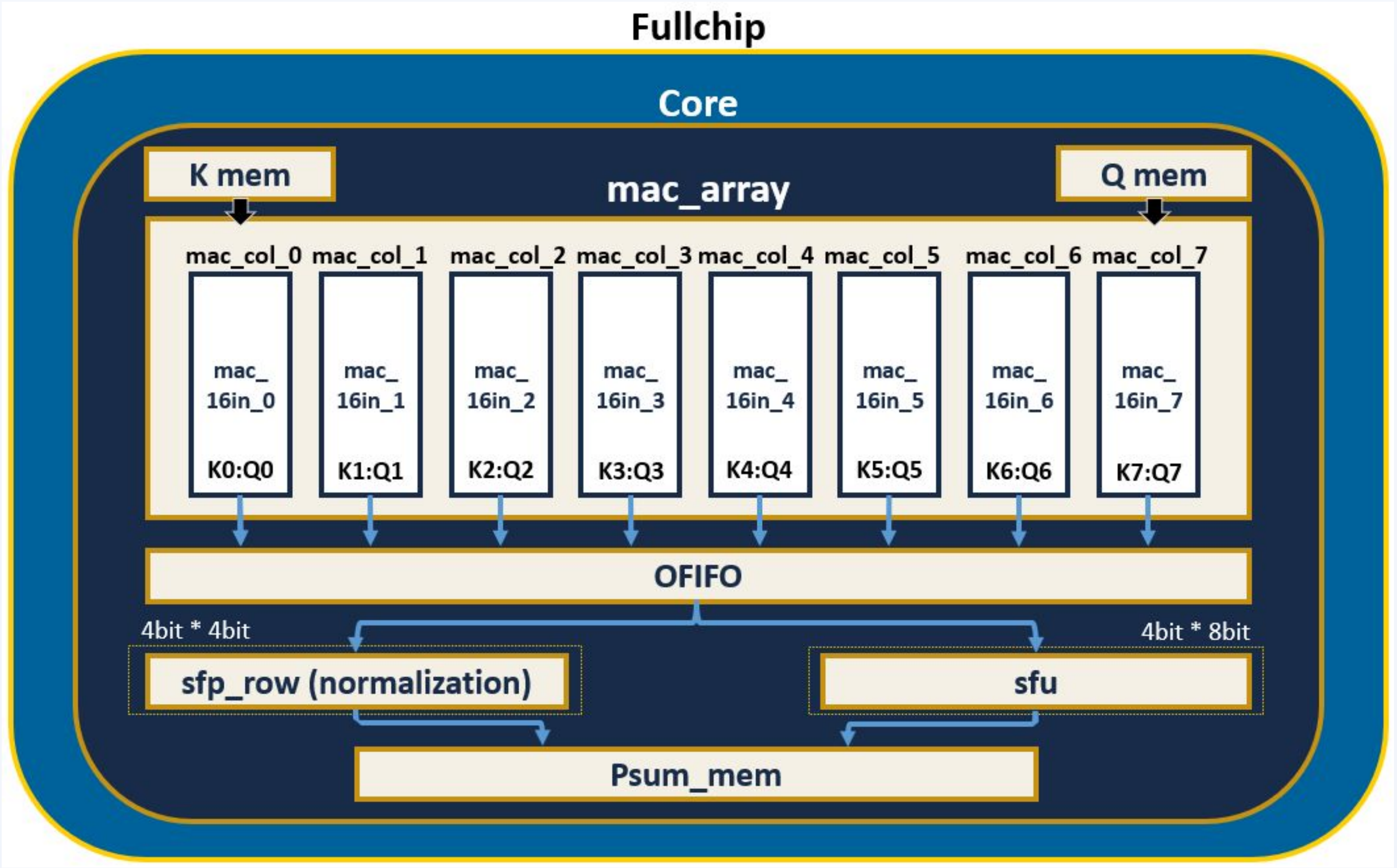
Synthesis Results

	WNS (ns)	Total Power	Total Cell Area (μm^2)
Single Core	-1.682	170.18mW	441861
Dual Core	-1.7	170.28mW	440056
Dual Core Optimized	0	822uW	182870

PNR Results

	WNS(ns)	Leakage Power (mW)	Switching Power (mW)	Total Power (mW)	Total Cell Area (μm^2)	VCD Power Total(mW)
Single Core	1.005	1.05	16.5	66.4	450912	133.734
Dual Core	-3.443	2.6	169	743.8	918147	164.02
Dual Core Optimized	-0.085	2.113	57	199.7	365633	102.5

Alpha 1: Reconfigurable Mode



Fullchip

Core

K mem

mac_array

Q mem

mac_col_0 mac_col_1 mac_col_2 mac_col_3 mac_col_4 mac_col_5 mac_col_6 mac_col_7

mac_16in_0 mac_16in_1 mac_16in_2 mac_16in_3 mac_16in_4 mac_16in_5 mac_16in_6 mac_16in_7

K0:Q0 K1:Q1 K2:Q2 K3:Q3 K4:Q4 K5:Q5 K6:Q6 K7:Q7

OFIFO

4bit * 4bit

4bit * 8bit

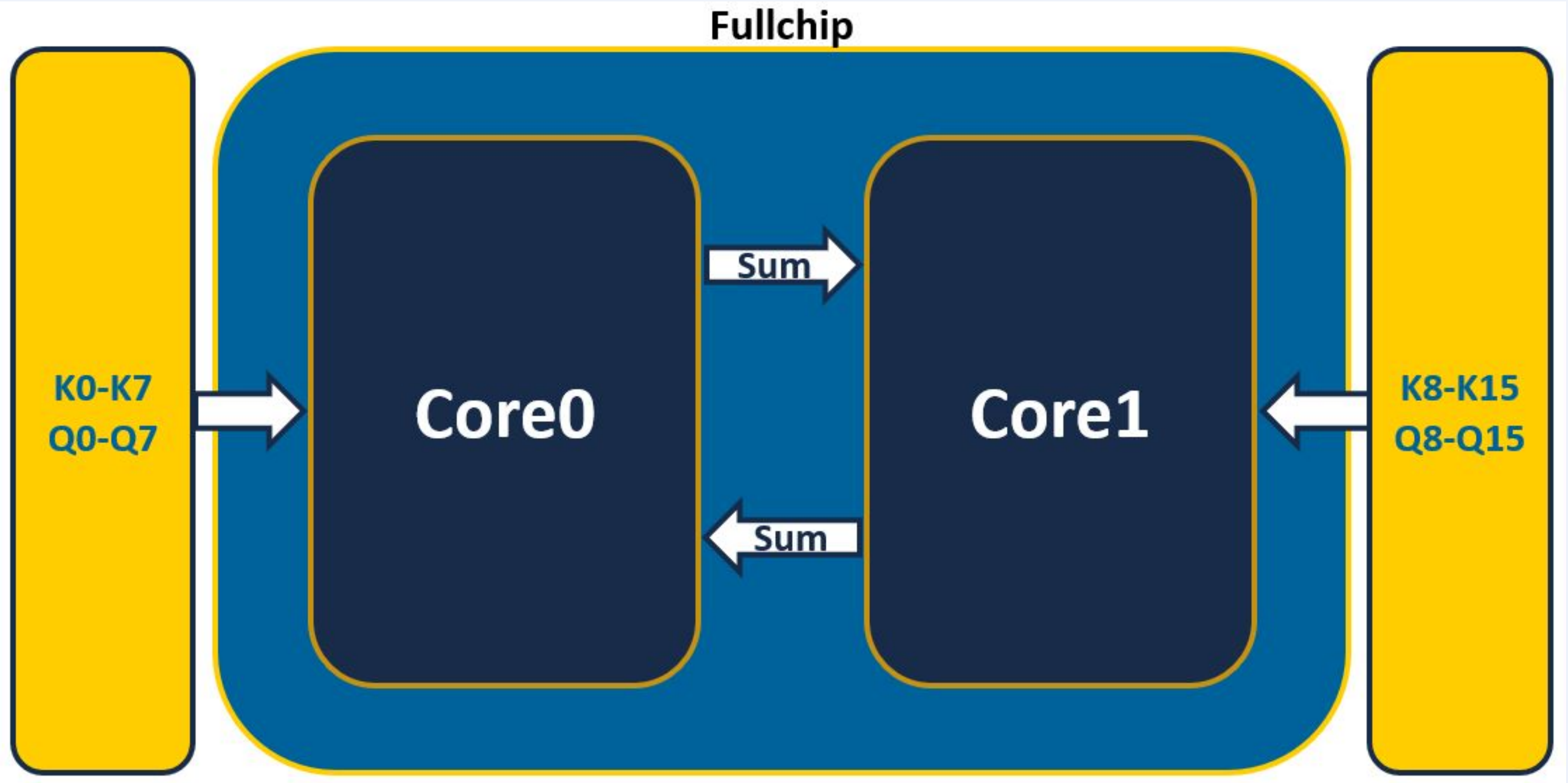
sfp_row (normalization)

sfu

Psum_mem

- **Reconfigurability**
  - ○ Hardware configurability
    - ■ 4bit * 4bit (via sfp_row → normalization)
    - ■ 4bit * 8bit (via special function unit sfu)
  - ○ Bit precision support
    - ■ Signed and unsigned base

Dual Core



Fullchip

K0-K7
Q0-Q7

Core0

Sum

Sum

Core1

K8-K15
Q8-Q15

- **Parallel processing:**
  - ○ Utilize 2 single cores for parallel computation
    - ■ Process 16 vectors (8 vectors per core)
  - ○ FIFO Synchronizer
    - ■ Sum exchange btw two cores → normalization
  - ○ Similar instruction signals for both cores

Alpha 2: Optimization

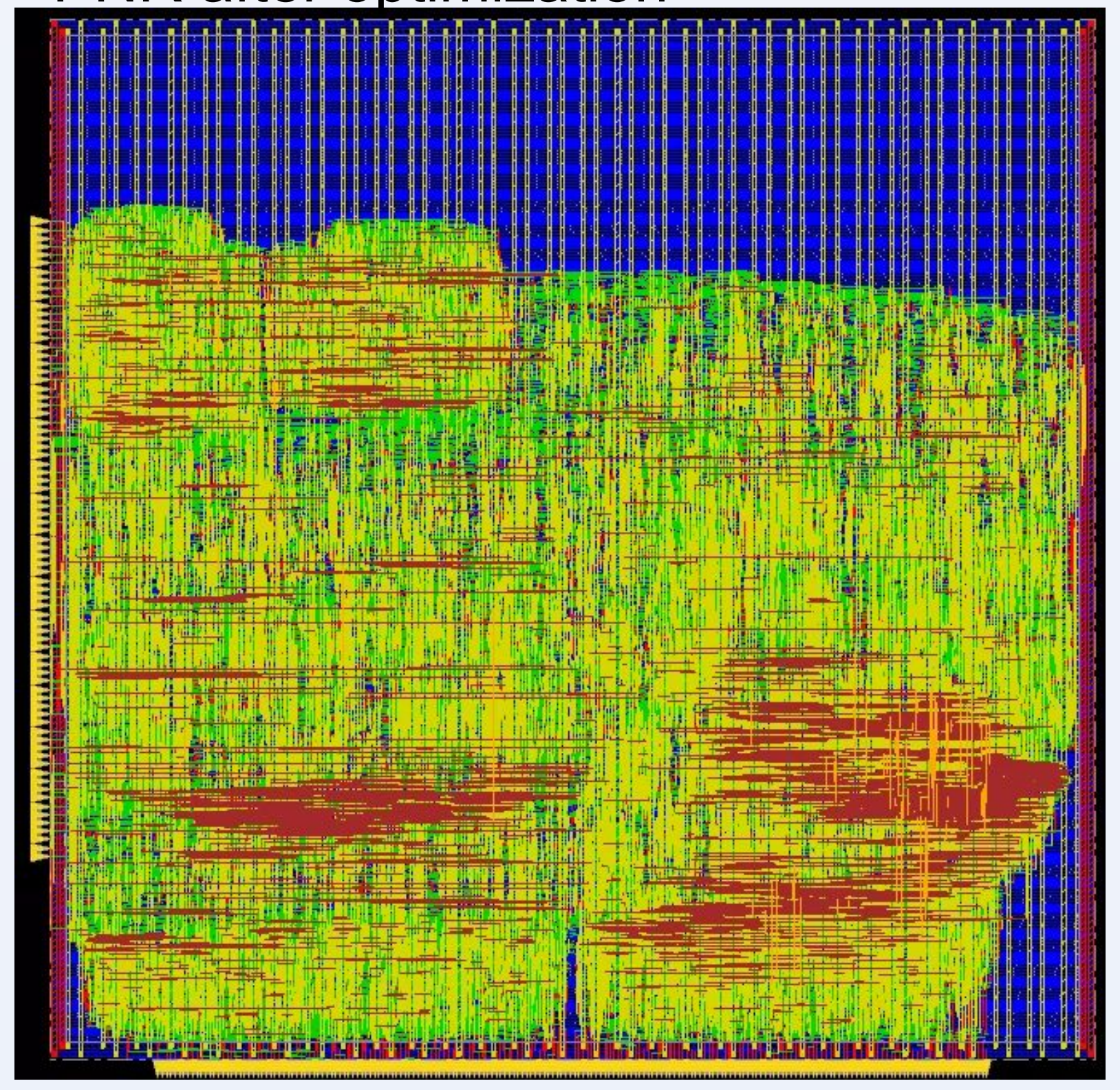
Goal: *Improve performance of dual core computation in terms of power & timing*

- **Applied Techniques/Methods**
  - ○ Lowering of bit precision to 4 & set PR = 8
  - ○ Synthesis → Flatten all
  - ○ Pipelining
    - ■ 1.) sfp_row
      - ● Reading from fifo & sum
    - ■ 2.) mac_col
      - ● Product & sum
  - ○ Multicycle paths

→*Result*

- **Improvements**
  - ○ **Timing → 0 WNS**
  - ○ **Power → Decreased by ~37%**
  - ○ **Cell Area → Decreased by ~60%**

*PNR after optimization



Future Improvements

- **Improvements**
  - ○ Clock gating for more power reduction
  - ○ Multicycle paths in PNR
  - ○ Increasing the total # of cores
    - ■ More parallelism
  - ○ SDF (VCD)

www.PosterPresentations.com