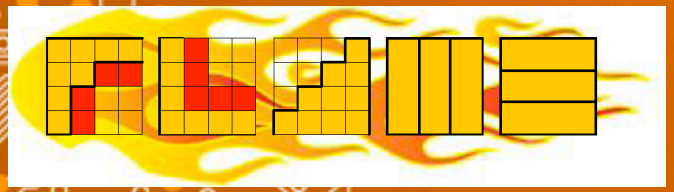
THE UNIVERSITY OF TEXAS AT AUSTIN
Cockrell School of Engineering

Optimizing for the Inner Kernel:
A Low Power, High Performance Core for Matrix Multiplication Kernels

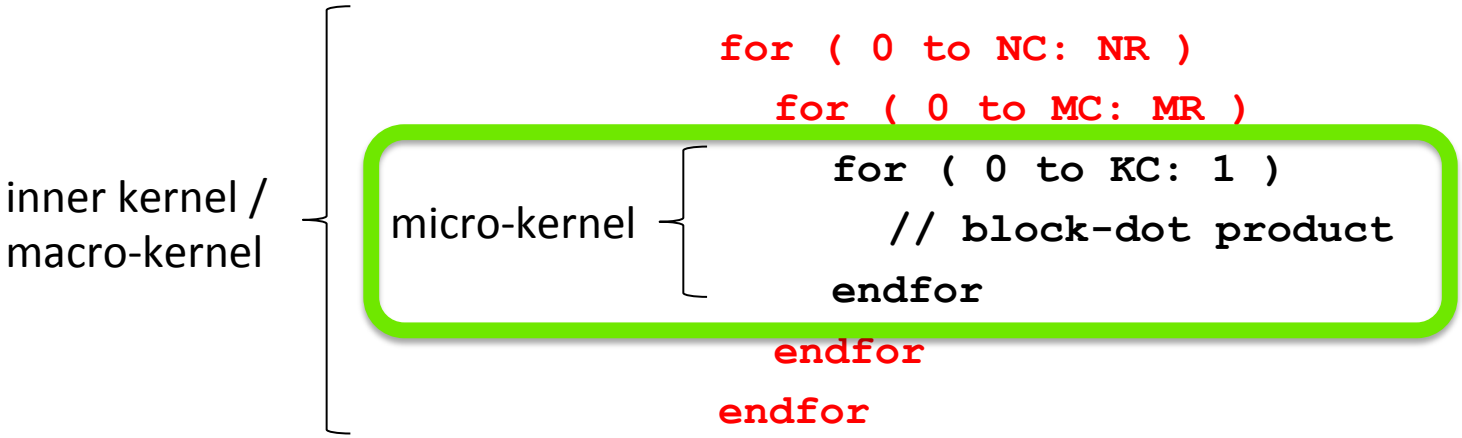
Ardavan Pedram



THE UNIVERSITY OF TEXAS AT AUSTIN
UT ECE
ELECTRICAL & COMPUTER ENGINEERING

The GEMM Micro-kernel

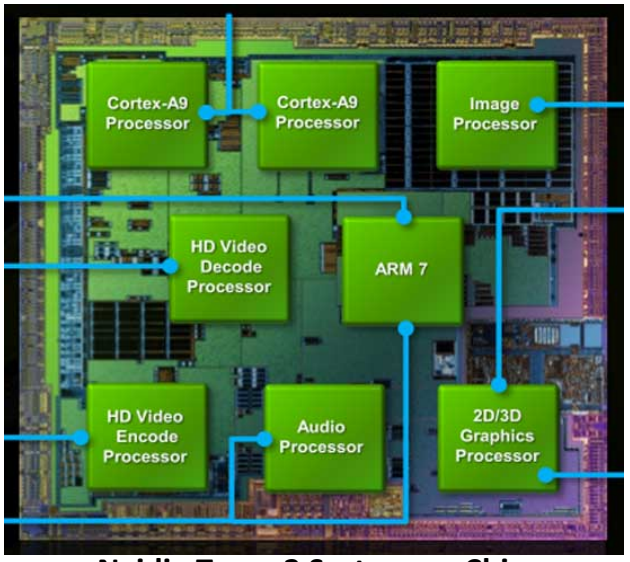
- BLIS exposes outer two loops of the inner kernel
- **Key observation**: Virtually *all* of the differences between level-3 inner kernels reside in the outer two loops



Era of Heterogeneous Computing

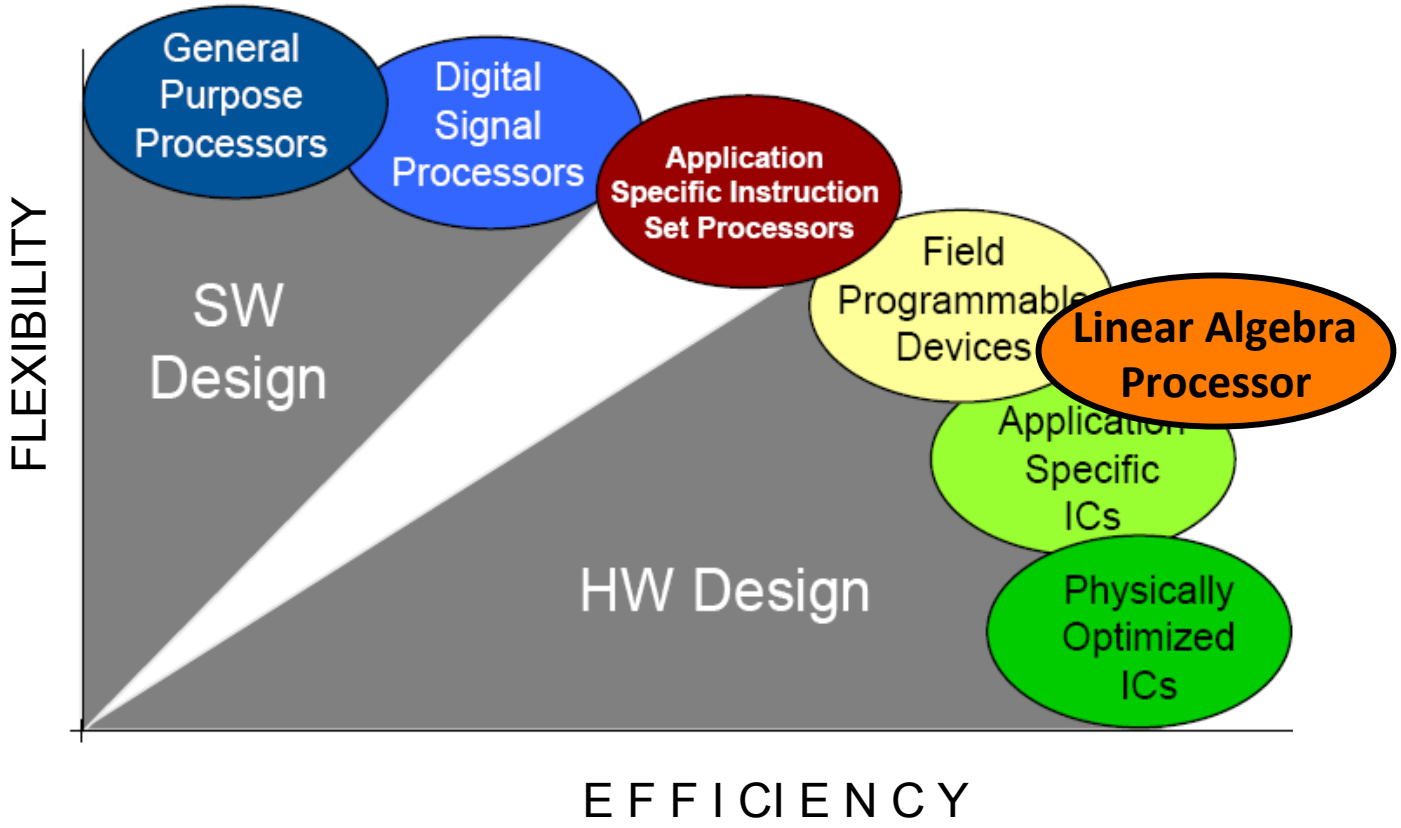
- Physical limits of technology scaling
  - Power wall and dark silicon
    - Only a fraction of a chip may be active
    - Efficiency/optimality vs. flexibility/generality
    - GFLOPS/W (energy per operation)

➢ Opportunity for specialization
  - ➢ Heterogeneous multi-core / Asynchronous CMP
  - ➢ On-chip accelerators



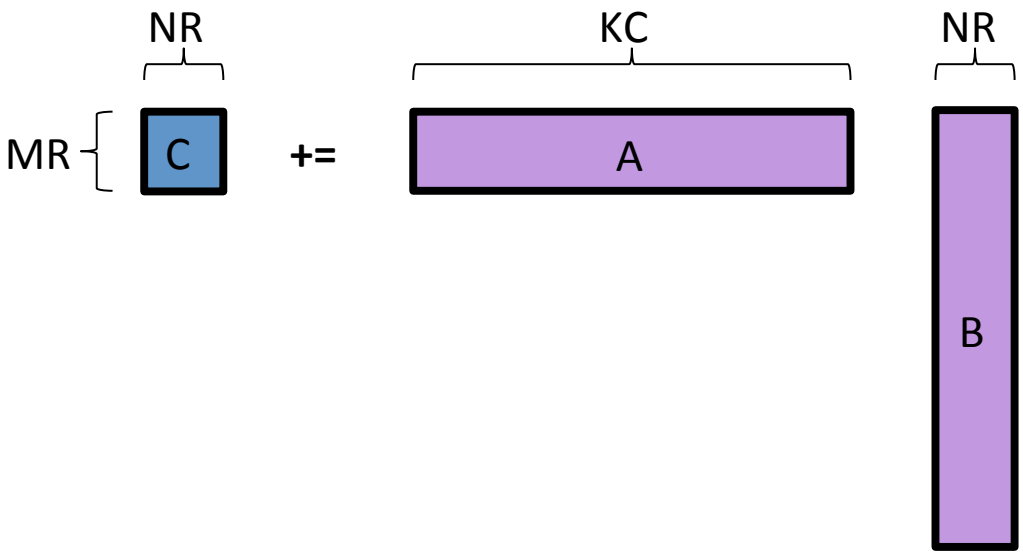
Nvidia Tegra 2 System on Chip

Implementation Spectrum



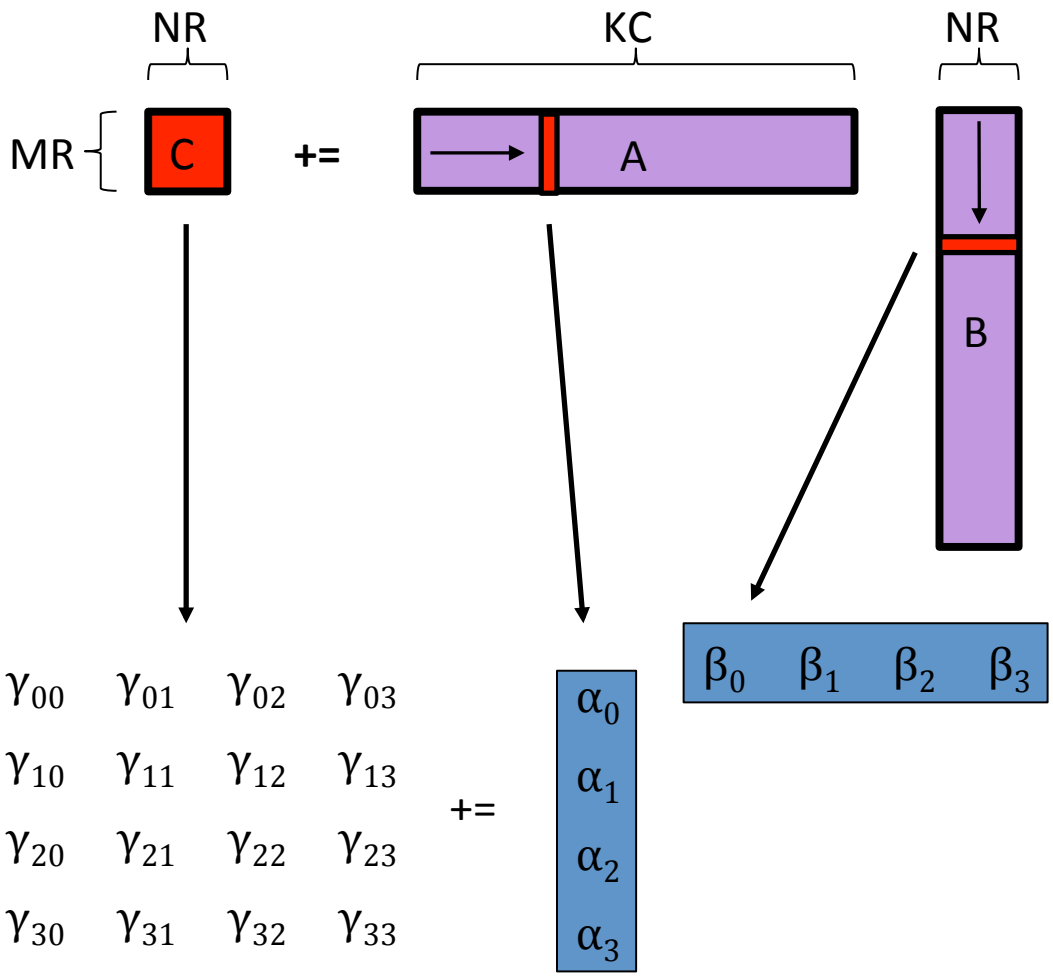
Source: T. Noll, RWTH Aachen, via R. Leupers, "From ASIP to MPSoC", Computer Engineering Colloquium, TU Delft, 2006

The GEMM Micro-kernel




```
for ( 0 to NC: NR )
  for ( 0 to MC: MR )
    for ( 0 to KC: 1 )
      // block-dot product
    endfor
  endfor
endfor
```

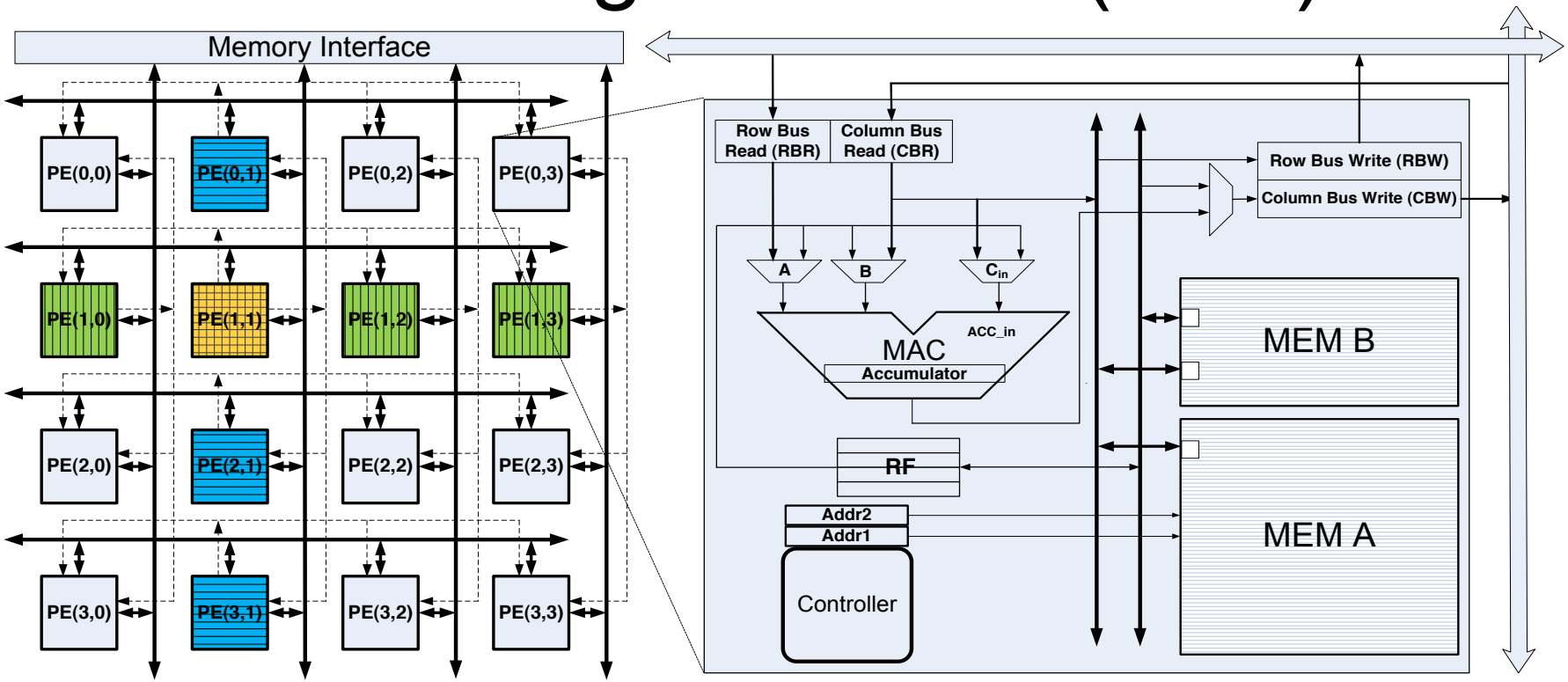
The GEMM Micro-kernel




```
for ( 0 to NC: NR )
   for ( 0 to MC: MR )
      for ( 0 to KC: 1 )
         // block-dot product
      endfor
   endfor
endfor
```

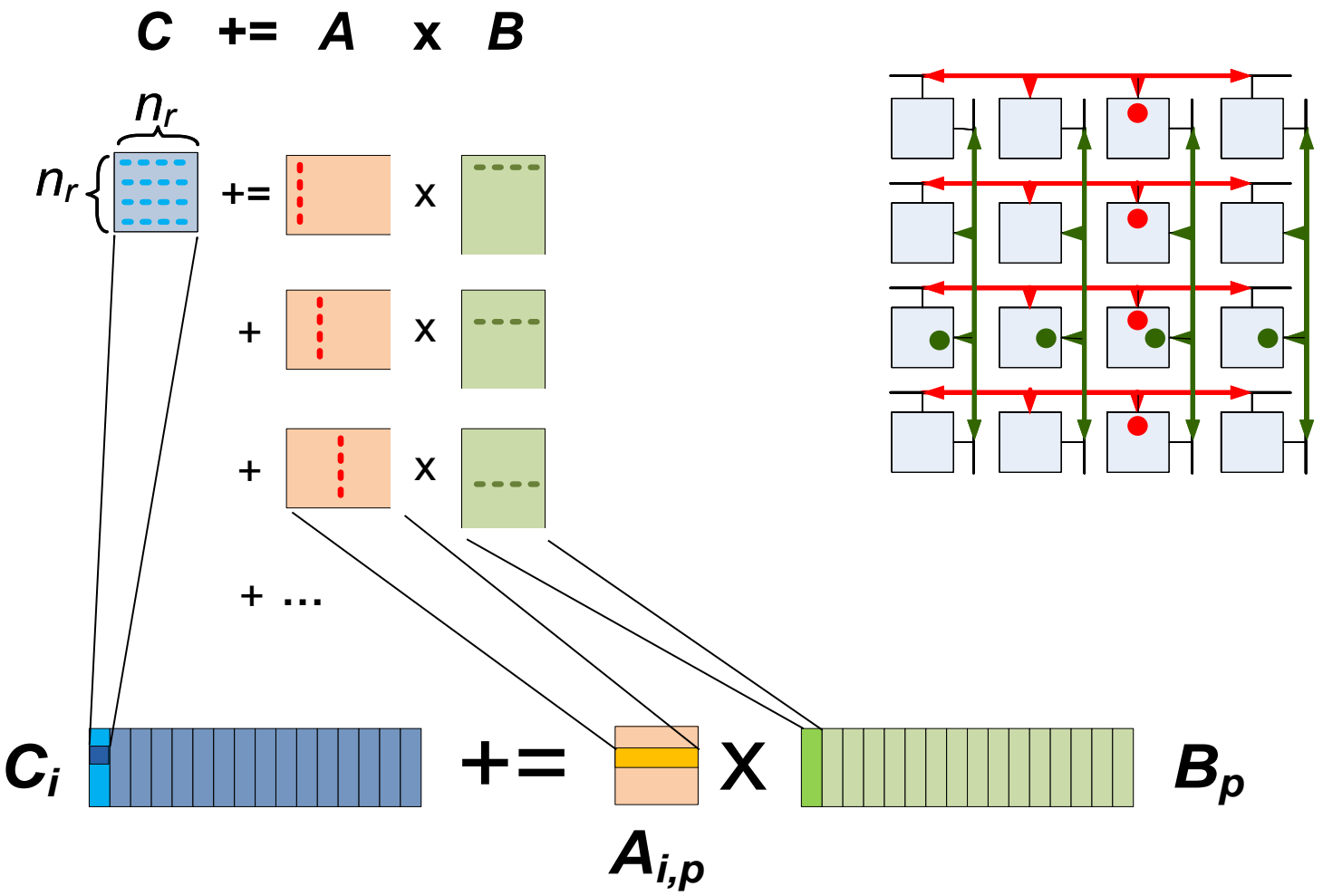
- Typical micro-kernel loop iteration ("block-dot product")
  - Load column of packed A
  - Load row of packed B
  - Compute outer product
  - Update C (kept in registers)

Linear Algebra Core (LAC)



- Scalable 2-D array of $n_r \times n_r$ processing elements (PEs)
  - Specialized floating-point units w/ 1 MAC/cycle throughput
  - Broadcast busses (no need to pipeline up to n_r=16)
  - Distributed memory architecture
  - Distributed, PE-local control

The "Rank-1" Machine



Core-Level Implementation

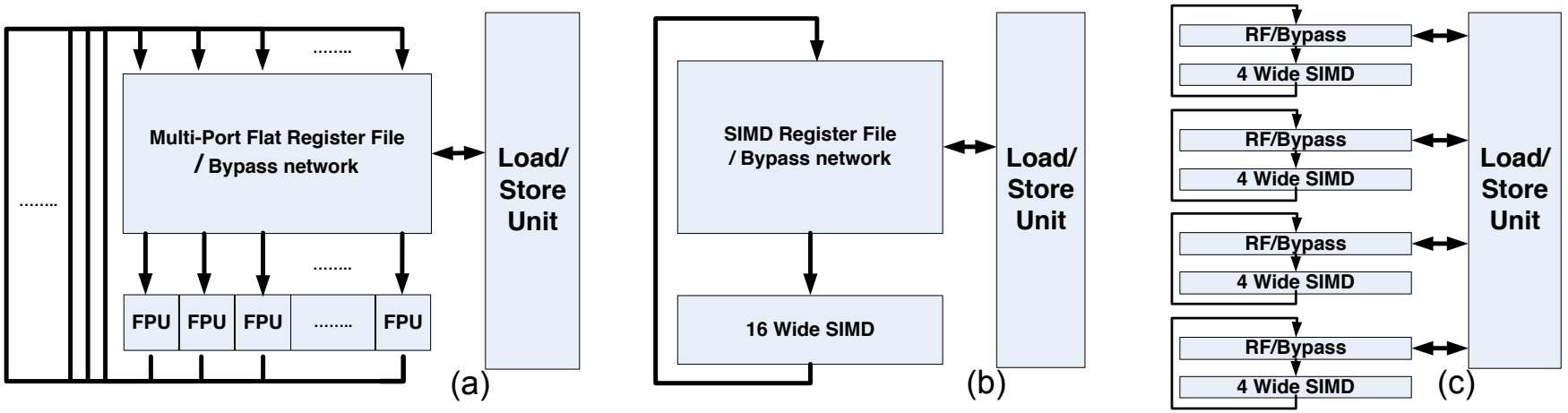
- 4x4 LAC
- 256 kB of local SRAM
- @ 1GHz
- Running GEMM @45nm technology

	$\frac{W}{mm^2}$	$\frac{GFLOPS}{mm^2}$	$\frac{GFLOPS}{W}$	Utilization
Cell SPE (SP)	0.4	6.4	16	83%
NVidia GTX480 SM (SP)	0.5	4.5	8.4	70%
NVidia GTX480 SM (DP)	0.5	2.05	4.1	70%
Intel Core (DP)	0.5	0.4	0.9	95%
ClearSpeed (DP)	0.02	0.3	13	80%
LAC (SP)	**0.2**	**20**	**104**	**95+%**
LAC (DP)	**0.3**	**16**	**47**	**95+%**

Design Trade-off Challenge

- Some Linear algebra data handling requirements
  - Matrix Transpose (SYRK)
  - Vector Transpose (Matrix-vector multiply) (LU)
  - Broadcast (GEMM)
- Common Sources of Inefficiencies in CPUs & GPUs
  - Instruction handling
  - Multi-ported register file
  - Cache overheads: tags and coherency
- What is the best solution for those data handling requirements?

Studied Register File Organizations



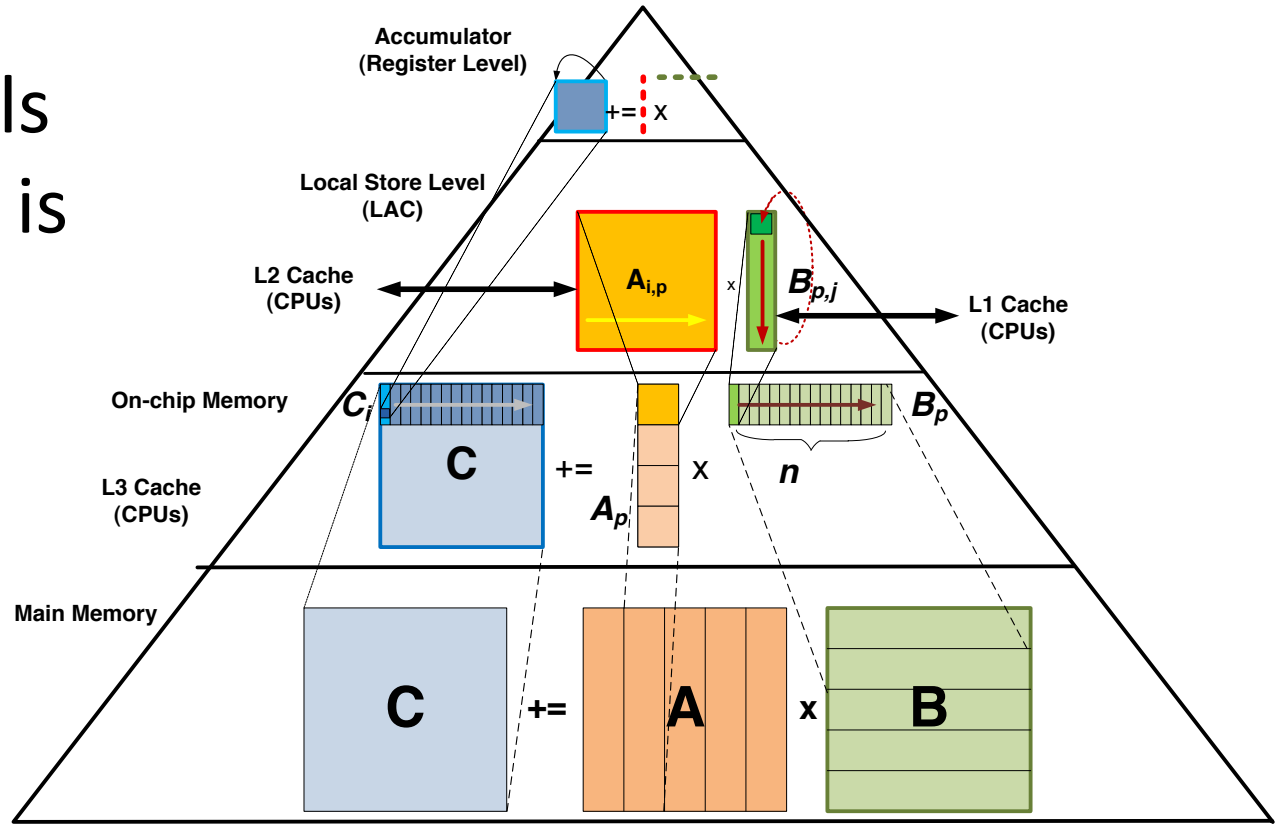
- Organizations for 16 FPUs
  - (a) Flat multiple-ported register file
  - (b) Wide SIMD with wide ports
  - (c) Array of short SIMD units
- No limits
  - load-store unit data movement and bandwidth
  - Scalar pipeline (# instructions that are executed)

Data Handling Trade-off in Core

- GEMM[ASAP2011]
  - Regular data movement
  - Regular access pattern
  - Easily mapped on most SIMD Cores
- SYRK[ASAP2012],[SBAC-PAD2012]
  - Matrix Transpose needed
  - Data manipulation for Transpose is a challenge

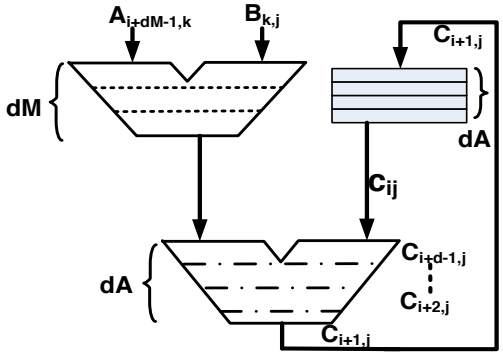
Memory Hierarchy of GEMM

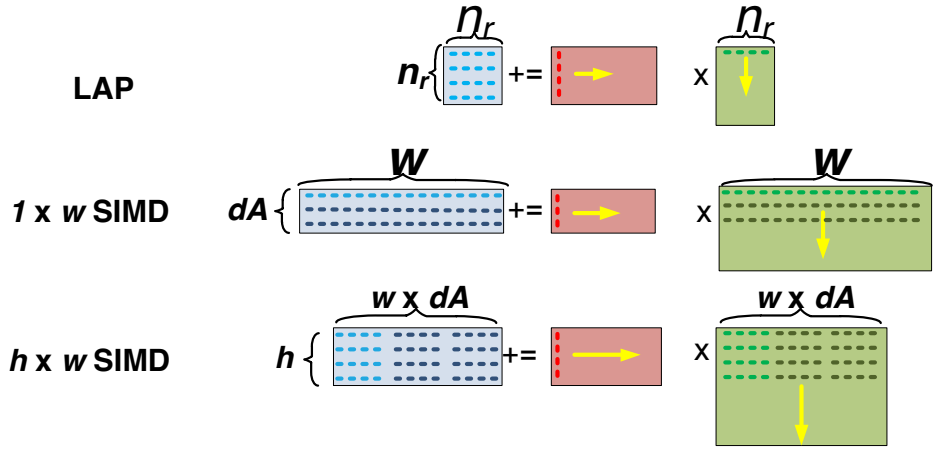
- Blocking in higher levels of memory is almost the same

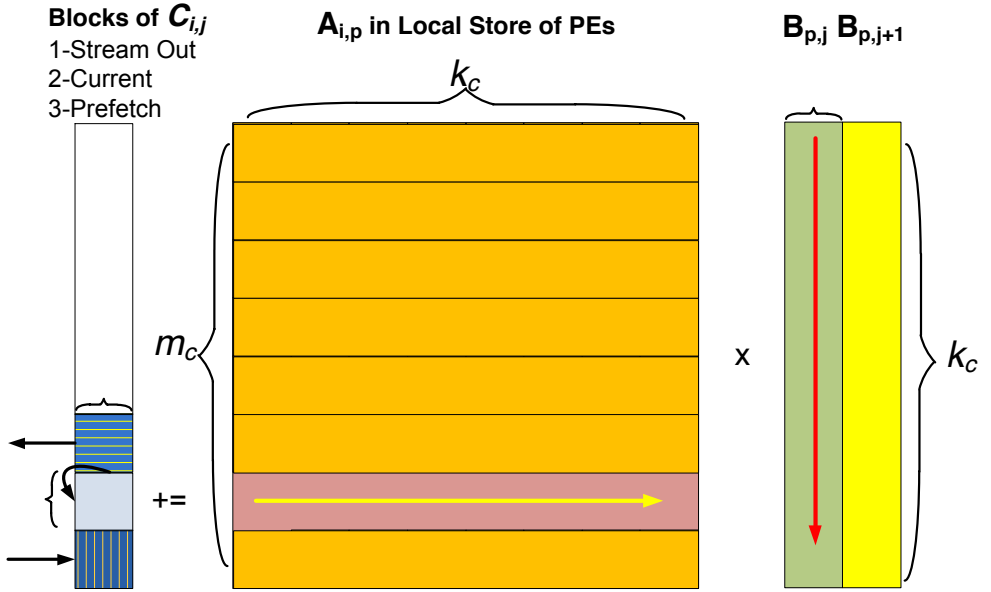


GEMM Register-level Operations

- Accumulator delay multiplies the number of registers needed in core

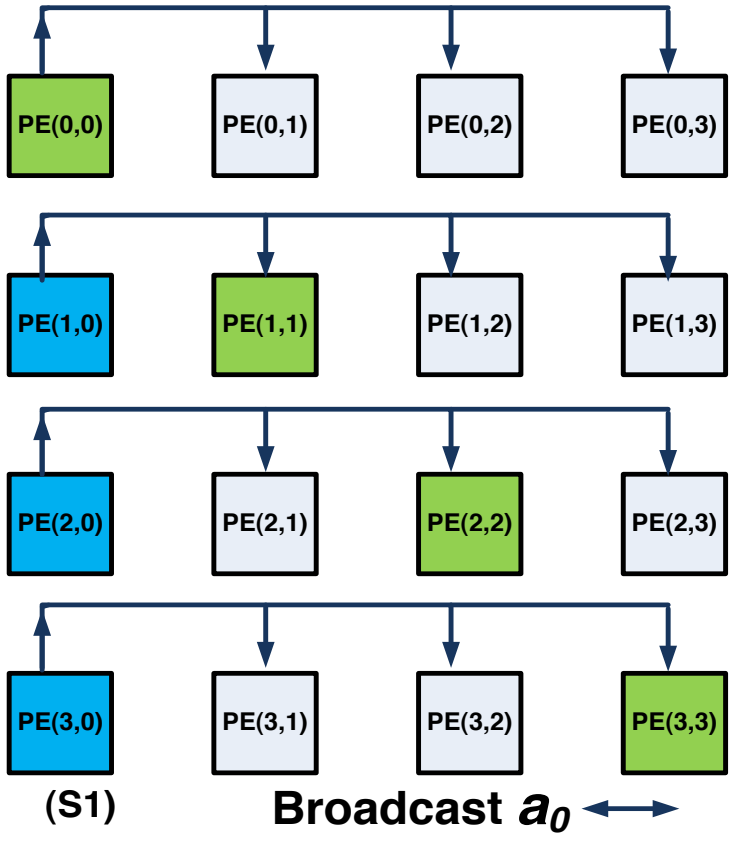






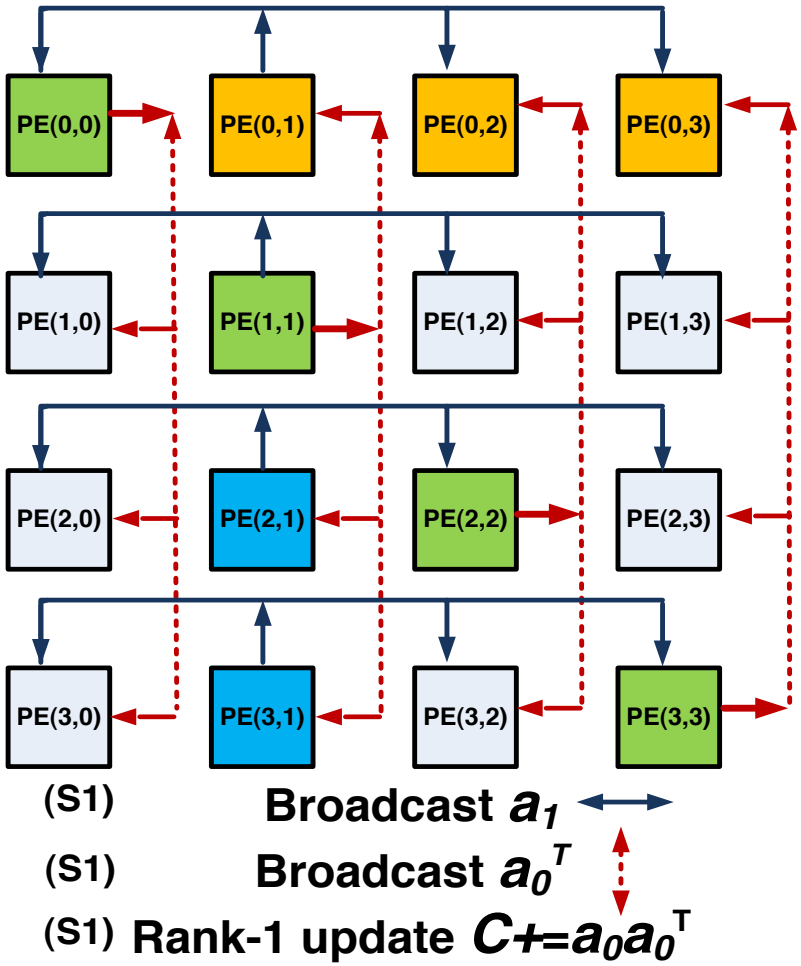
Non-Blocked n_r x n_r SYRK on LAC

- LAC takes advantage of the broadcast buses
- Overlap computation with communication with transposition
- Least operation count in part of bigger SYRK problem
- Simple control
  - 3-stage state machine in PEs



Non-Blocked n_r x n_r SYRK on LAC

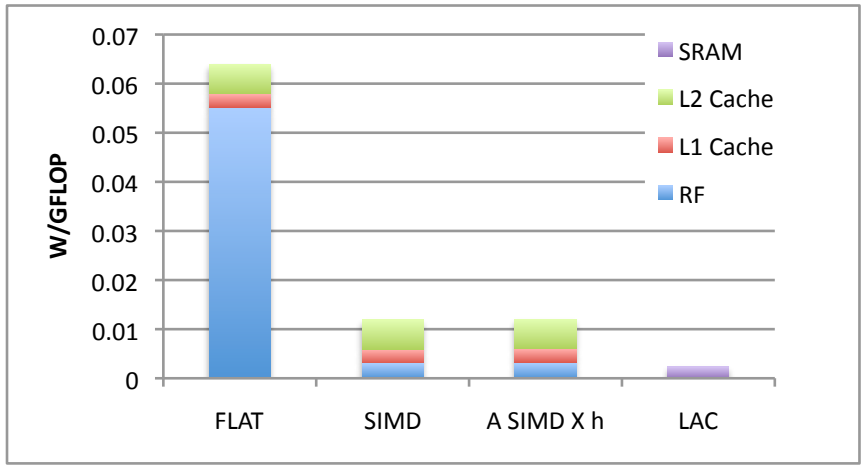
- LAC takes advantage of the broadcast buses
- Overlap computation with communication with transposition
- Least operation count in part of bigger SYRK problem
- Simple control
  - 3-stage state machine in PEs

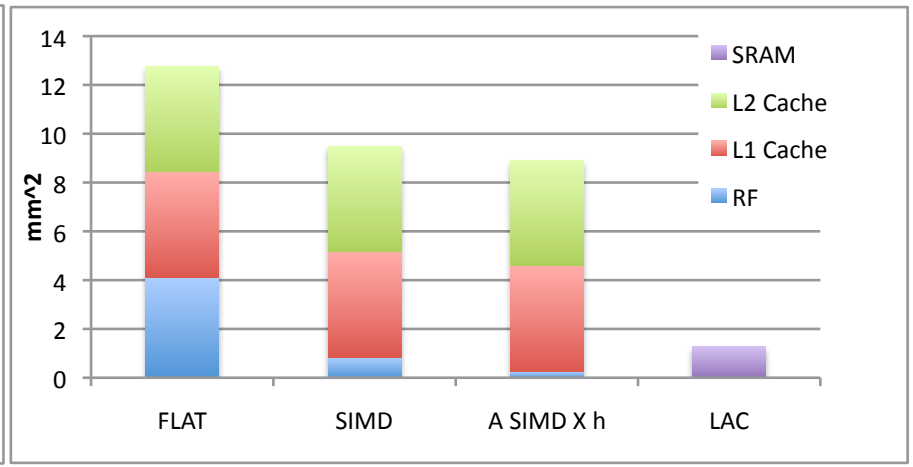


Power and Performance Analysis

- Analytical formulae
  - Utilization
  - Bandwidth
  - Size of local stores
  - Size of register file
  - #ports of memory levels
- Component selections
  - Storage model [CACTI 6.5]
    - Pure SRAM Model for Register file
    - Cache Models for L1 & L2
- Configuration
  - 1 GHz
  - 16 FPUs
  - LAC
    - 256+32 KB aggregate Local store
  - Register file organizations
    - 256 KB L2, 64B line, 8 way
    - 32 KB L1, 64B line, 8way
    - Register file Size variable
      - Aimed for peak performance

GEMM Power and Area

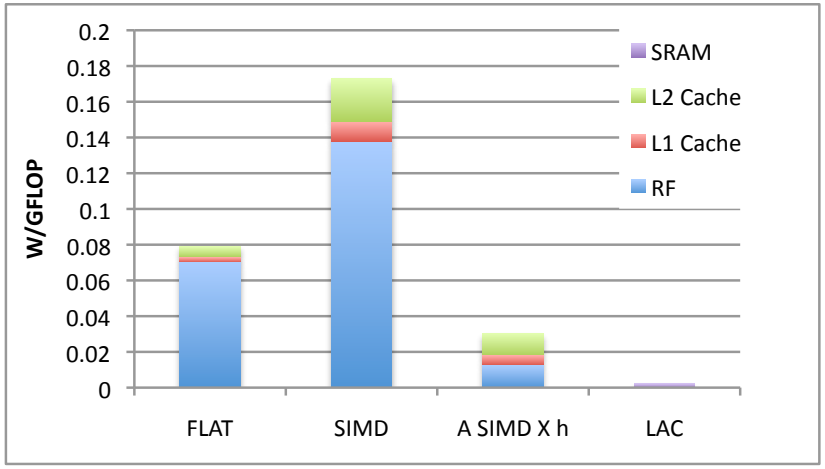


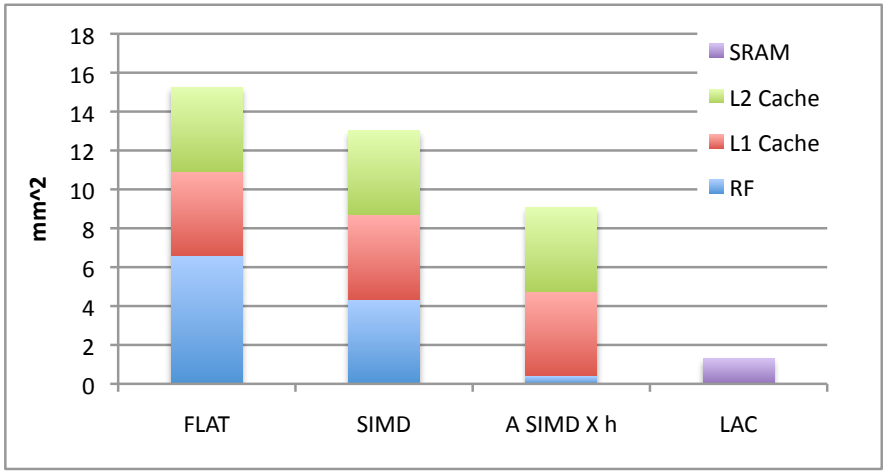


- SIMD architectures
  - Effectively reduce the power consumption
- LAC is the best not using caches
- The flexibility of flat register files is of no use for GEMM.

- 4×4-wide array of SIMD requires less area.
- Most of the area is used by Caches

SYRK Power and Area





- Wide SIMD takes many cycles to transpose
- 4×4-wide SIMD has best efficiency
  - Less power than FLAT
  - Higher utilization then Wide SIMD

- 4×4-wide array of SIMD requires less area.
- Most of the area is used by Caches

Conclusion

- Architectures Studied
  - Linear Algebra Core (LAC)
  - FLAT register file
  - 1D wide SIMD
  - 4×4-wide array of SIMD
- Algorithm/Architecture co-design
  - GEMM (emphasize on data bandwidth to-from core)
  - SYRK (emphasize on data manipulation in core)
- Analytical formulae for different layers of memory hierarchy
- Power and efficiency estimation

Linear Algebra Core

- Results @ 1GHz
  - DP: 32 GFLOPS, 47 GFLOPS/W
  - 0.6 Watts
  - 2.8 mm^2 in 45nm
  - 4 GB/s external BW
  - Orders of magnitude improvement

➢ On-going and future work
  - Generalization: full Level-3 BLAS in hardware
  - Implementation: synthesis and tape-out

Questions ?