Anton: A Specialized ASIC for Molecular Dynamics

Martin M. Deneroff,
David E. Shaw, Ron O. Dror, Jeffrey S. Kuskin,
Richard H. Larson, John K. Salmon, and Cliff Young
D. E. Shaw Research
Marty.Deneroff@DEShawResearch.com

Themes in this Talk

- Why build a molecular dynamics simulation engine?
- Anton's overall, systematic design
  - Amdahl's Law in Specialized Computers
  - Putting it All Together
- "Green" Computing?

Molecular Dynamics Simulation

Molecular Dynamics (MD) is a way to simulate and observe the structure and dynamics of molecular systems based on physical principles

- Life Sciences research
- Drug discovery / validation of efficacy, specificity, toxicity → “Rational Drug Design”
- Materials Science

The Molecular Dynamics Computation

1. Divide time into uniform timesteps (~ 1 fs)
2. Compute forces on all particles in system in a timestep

E= *Bonded:* *(Flexible Subsystem)*

$\Sigma\ k_b(r-r_0)^2$ Stretch

$+\Sigma\ k_\theta(\theta-\theta_0)^2$ Bend

$+\Sigma\ A[1+\cos(n\tau-\varphi)]$ Torsion

Non-Bonded: *(High Throughput Interaction Subsystem)*

$+\Sigma\Sigma\ q_iq_j/r_{ij}$ Electrostsatic

$+\Sigma\Sigma\ q_iq_j/r_{ij}$ Van der Waals

3. Based on these forces, compute new velocities and positions for all particles

V= $V_0 + F * \Delta T / M$ *(Flexible Subsystem)*

S= $S_0 + V_0 * \Delta T + F * (\Delta T)^2 / M$

4. Rinse and repeat... About 1 TRILLION times to simulate a millisecond!

Also, do a bunch of ugly housekeeping functions along the way to maintain (as needed for experiment) constant temperature and / or pressure, etc.

Why Accelerate MD?

Most important biological processes require ~ 1 ms or real time to complete.

- Simulation of this length requires O(year) on the most powerful existing computers
- Maximum useful time for performing simulations is O(month) or less

Need to go faster or impractical!

Specialized Supercomputing

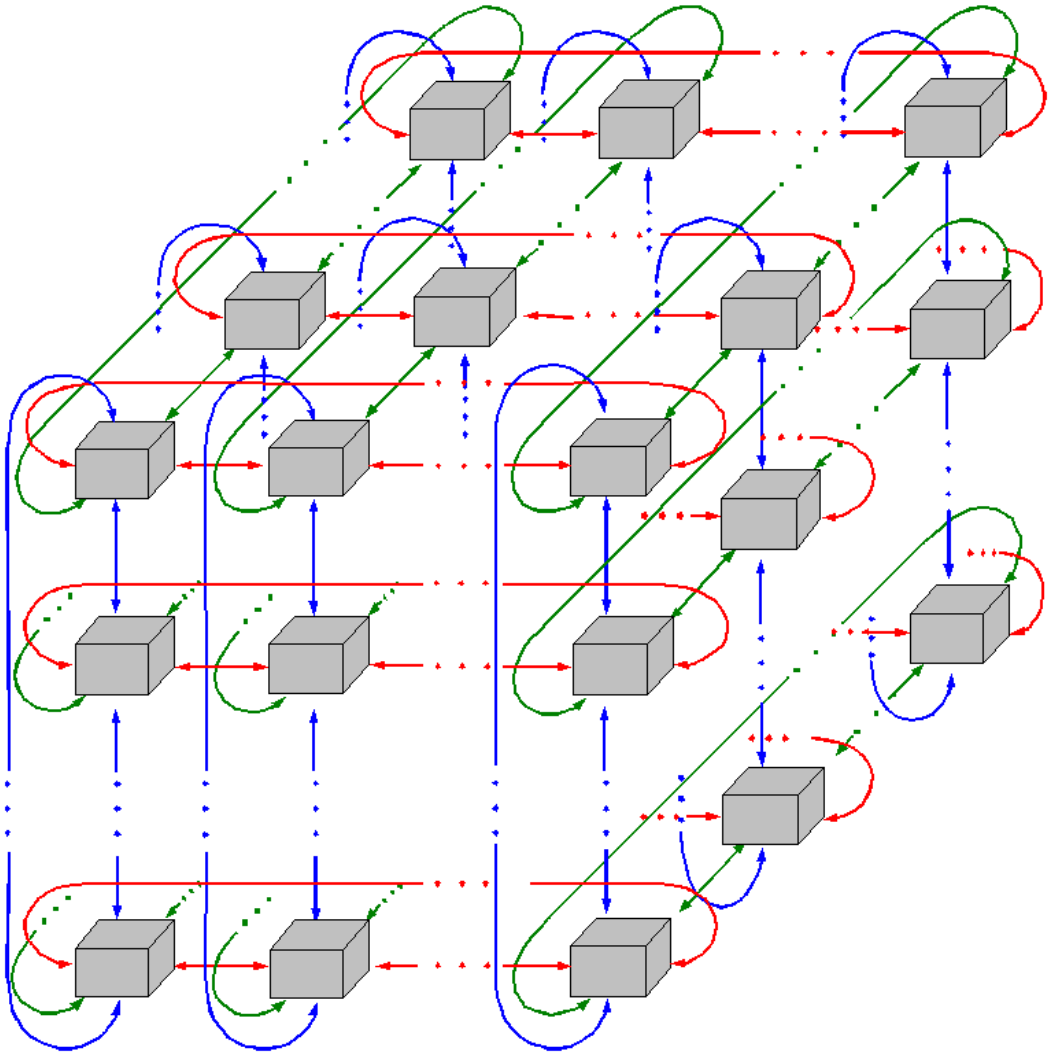
Anton is a specialized computer designed to perform molecular dynamics simulations

- Performance >100x general purpose computers
- Implemented as an array of identical ASICs connected as a 3D Torus

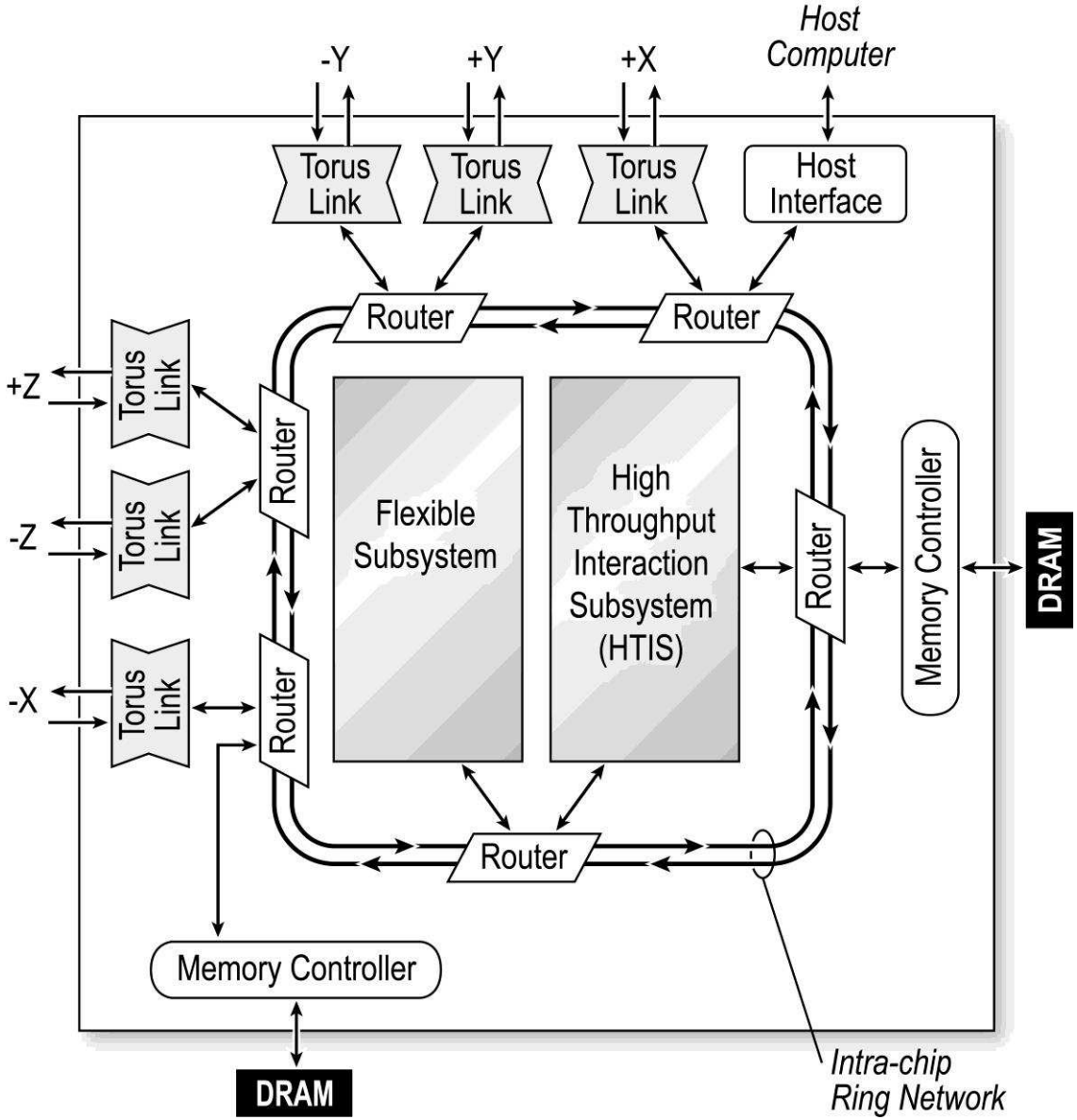
Anton Architecture

- Match Machine structure to logical partitioning of Problem:
  - 3D Spatial Decomposition with Periodic Boundary Conditions
    - > 3D Torus Interconnect
- Co-Design algorithms and hardware to optimize performance within technology constraints:
  - Neutral Territory (NT) method for particle movement tightly integrated with communications mechanisms
  - Gaussian Split Ewald method developed for ease of implementation relative to original Ewald method
- Pairwise interactions, the bulk of conventional MD, can be sped up almost arbitrarily using Direct Product Select-Reduce (DPSR) methods -- then, communications and other parts of MD computations dominate. Anton seeks to balance all parts and (to extent possible) execute all in parallel.

One Anton Segment (512 ASICs)



Anton ASIC Block Diagram



What makes Anton fast?

- Specialization
  - High-Throughput Interaction Subsystem (HTIS)
    - Highly specialized, largely hardwired
  - Flexible Subsystem
    - Programmable, but still specialized
  - Communication Subsystem
    - Over two orders of magnitude faster than Gigabit Ethernet
    - 50 ns hop latency, 640 Gbit/sec/node
    - Highly optimized end-to-end messaging with no software stack
- Parallelization
- System Integration

Why does Anton need **fast** general-purpose computation?

The GROMACS MD code on a single Xeon core spends:

- 71% of its time in HTIS-related tasks
- 29% of its time in other tasks
  - Amdahl's Law: accelerating only the HTIS tasks limits maximum speedup to <4
- Communication becomes significant fraction of this time when converted to parallel computation

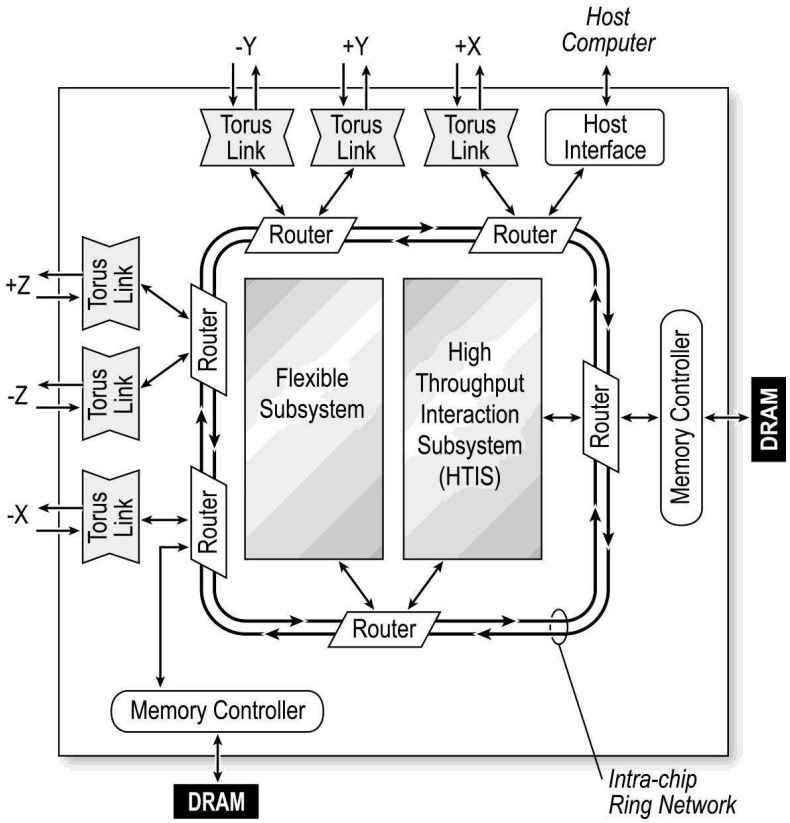
For our purposes, fast means two things:

- Low compute latency
- Low communication latency

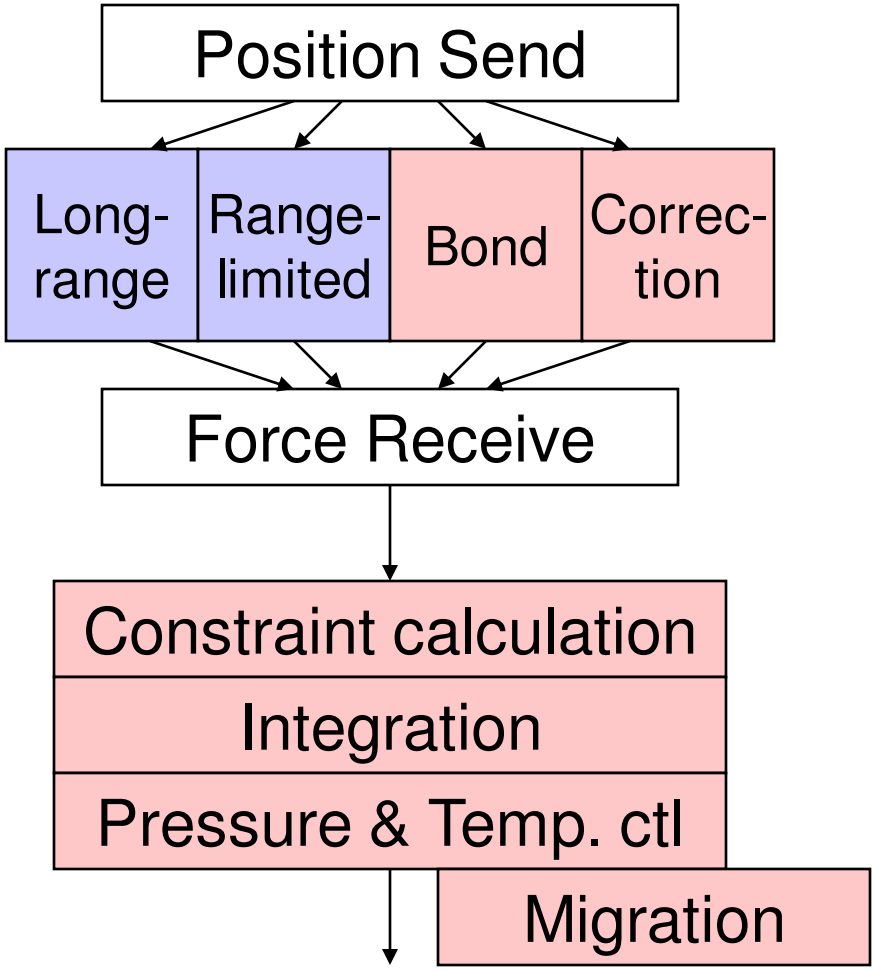
Why does Anton need general-purpose computation?

- HTIS Functions
  - The HTIS excels at Direct Product Select-Reduce (DPSR) operations.
    - These are the majority of calculations (>70%) in MD simulation
  - Midrange forces (electrostatic + van der Waals between pairs of atoms)
  - Charge spreading
  - Force interpolation

- Flexible Subsystem Functions
  - Bond forces
  - Correction forces
  - FFT and Fourier-space operations
  - MD Integration and Constraints
  - Pressure and Temperature control
  - Migration

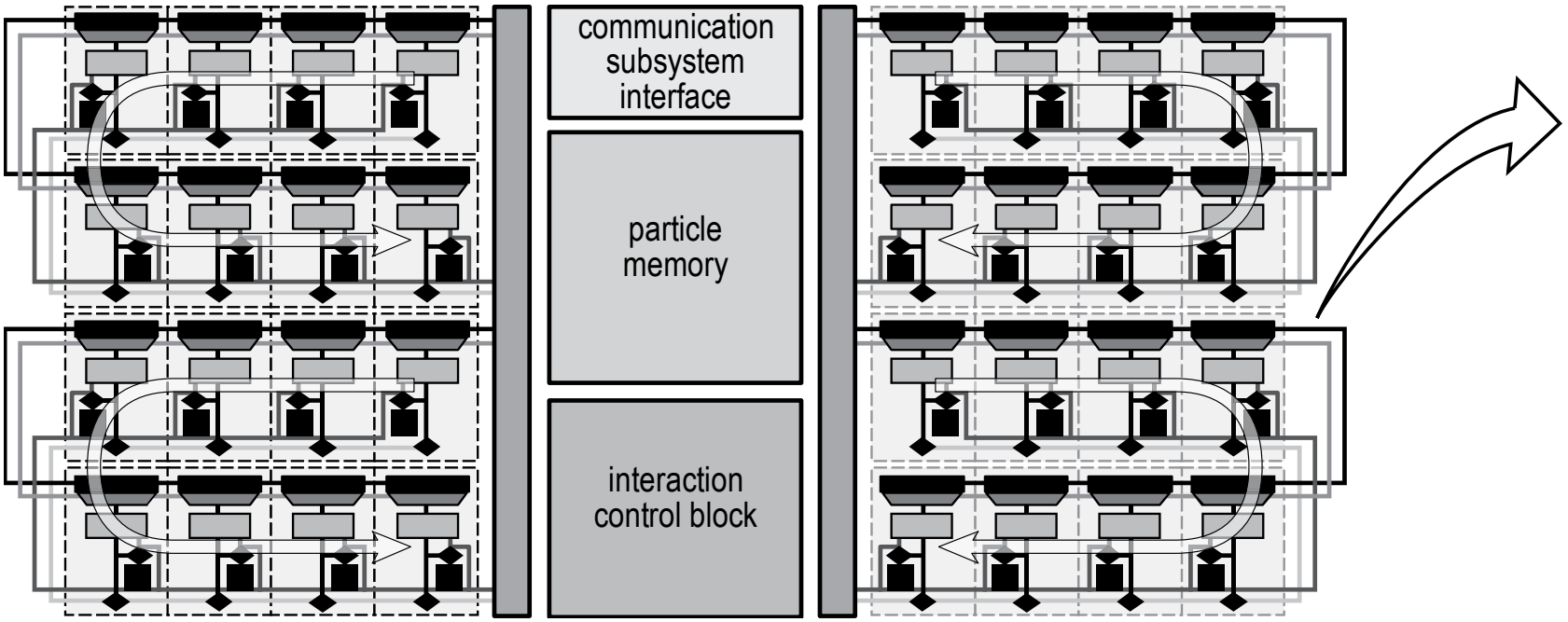


Functional view of an MD Time Step



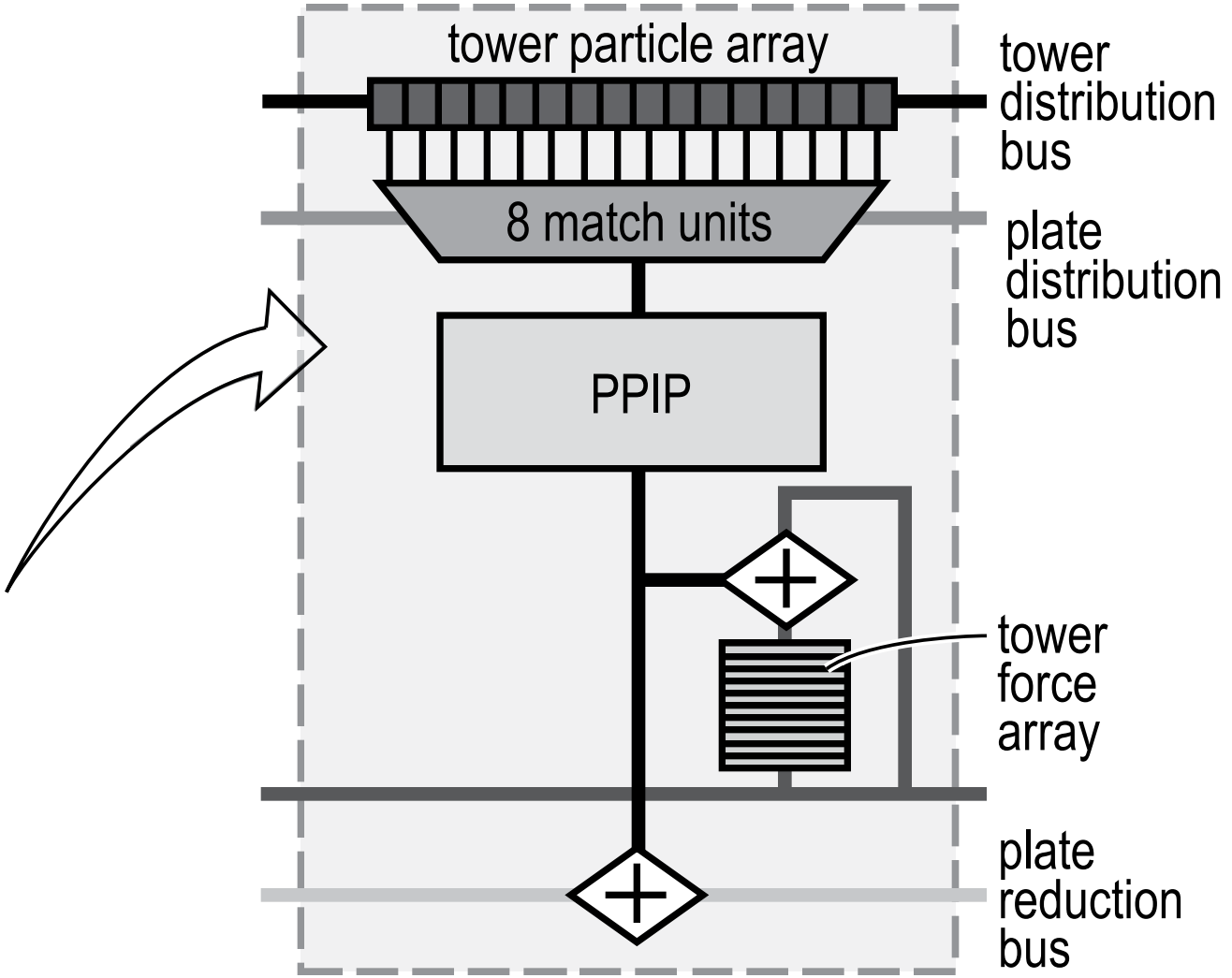
- Force Phase
  - Long-range forces
  - Range-limited forces
  - Bond forces
  - Correction forces
- MD Integration Phase
  - Constraint calculation
  - Integration
  - Pressure and Temperature control
  - [Migration]

Particle-Point Interactions In the HTIS



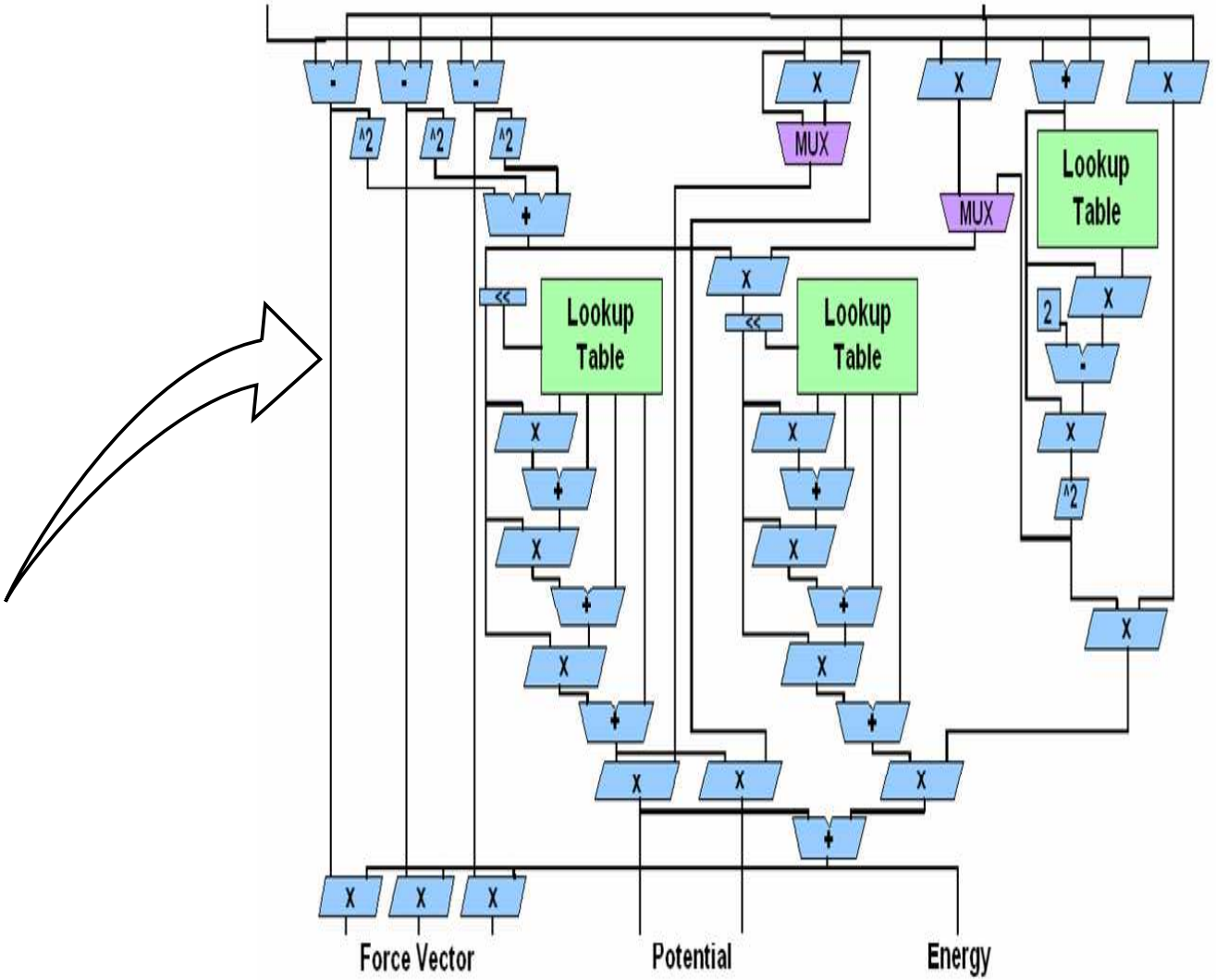
- 32 PPIMs, organized as 4 chains of 8 PPIMs each
- Achieves fantastic computational density, throughput
  - Calculates one non-bonded force/clock/PPIM
  - Equivalent calculation ~50 clocks on modern x86

Pairwise Point Interaction Pipeline (PPIP)



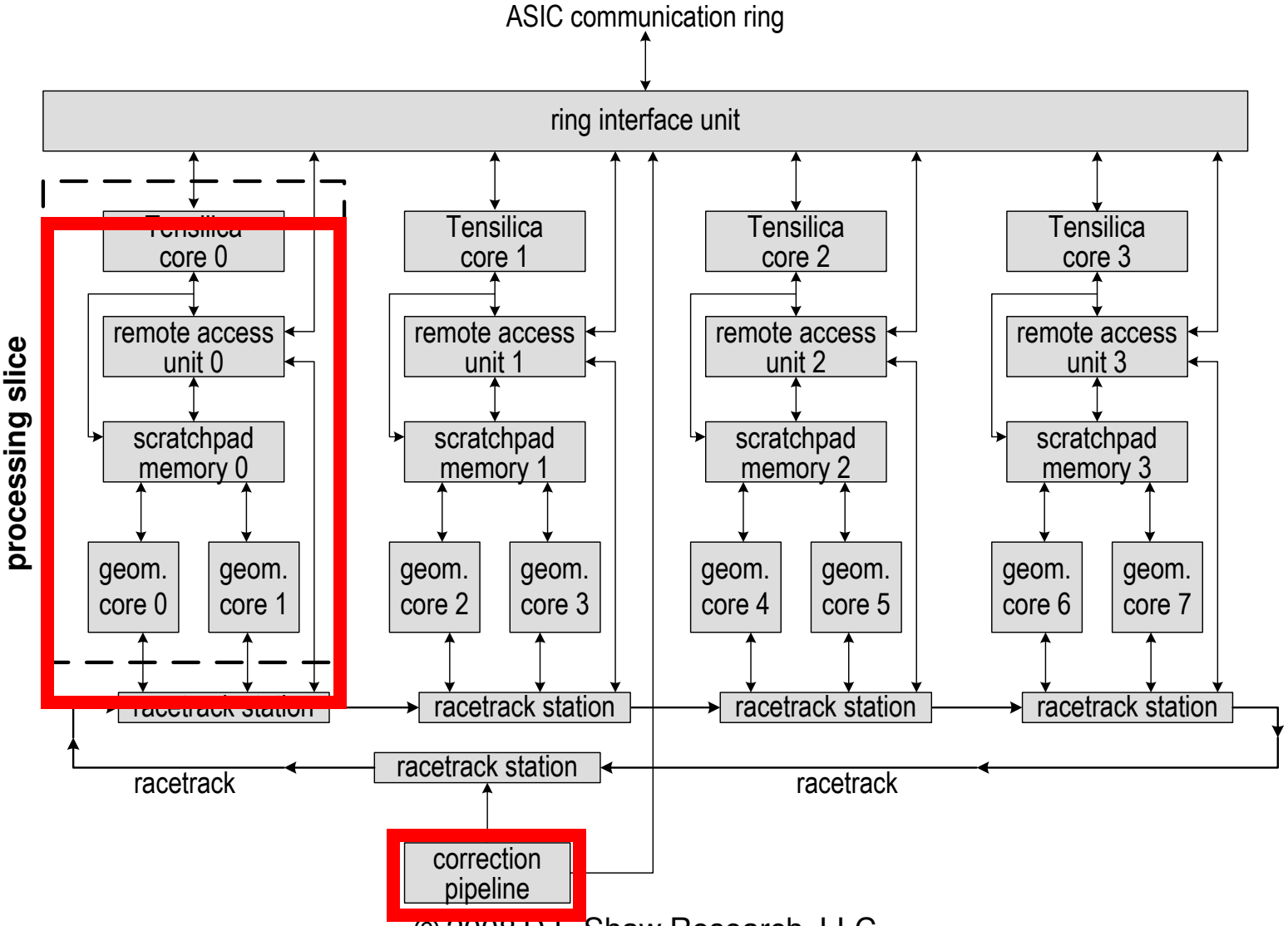
- Describe how PPIP and PPIM interact

Detail of One PPIM

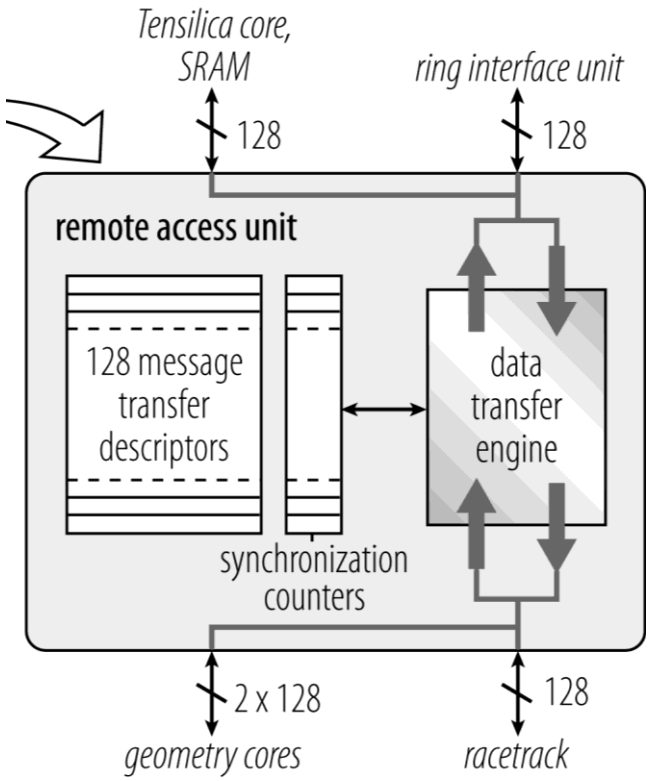


- How many SSE instructions would it take to implement one PPIM (compare to x86 SSE code)

Anton's Flexible Subsystem



Specialization: Remote Access Unit



- Closely-coupled DMA engine
  - 128 transfer descriptors; one operation to launch, change, or sync
  - Each descriptor corresponds to a hardware memmove operation
- Offloads communication from cores
- Supports “push” style of communication
  - Performs sends, counts acks, and counts received packets
  - Almost no reads or synchronizations
  - Push communications can be seen as “static dataflow”

Specialization: Geometry Core and ISA

The Geometry Core is a cusom design optimized for geometric computations using three and four element vectors.

- Parallel techniques used in the GC:
  - Dual-issue: instruction-level parallelism
  - 4-way SIMD: data-level parallelism
  - MAC operations: pipeline parallelism
  - 8 cores per chip: multicore parallelism

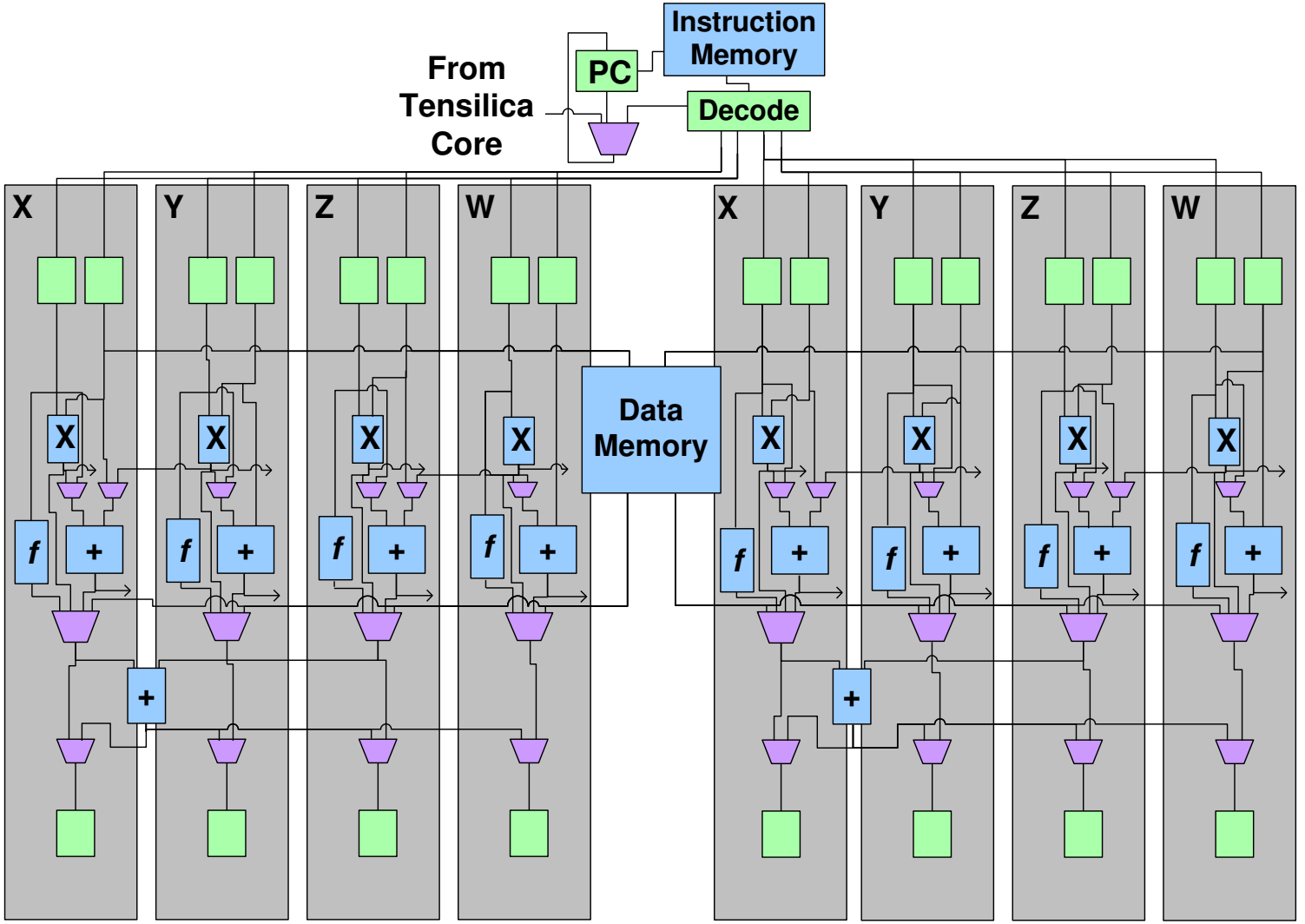
- Mixed 48 vector/192 scalar register file

  Reduce operation count by:
  - Putting result where it will be used next
  - Seldom spill to memory

- Specialized operations
  - Dot product, determinants, general vector permute,…
  - Biggest benefit: reduces operation count

- A Tensilica core is associated with each pair of GCs and primarily supervises data movement

Geometry Core (one of 8; 64 pipelined lanes/chip)



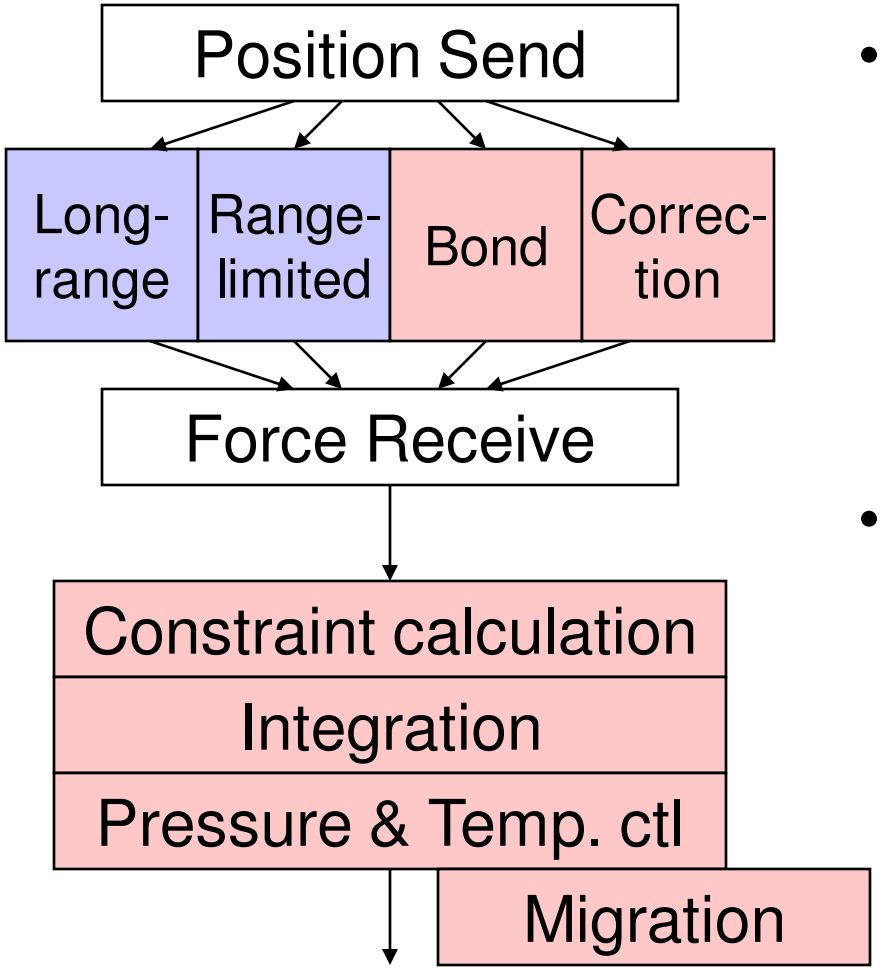
What makes Anton fast?

- Specialization

- Parallelization
  - 512 ASICs, each an SoC by itself
    - 2,560 Tensilica Cores (general-purpose control)
    - 4,096 Geometry Cores (general-purpose compute)
    - 16,896 PPIMs (special-purpose compute)

- System Integration

Functional view of an MD Time Step



- Force Phase
  - Long-range forces
  - Range-limited forces
  - Bond forces
  - Correction forces
- MD Integration Phase
  - Constraint calculation
  - Integration
  - Pressure and Temperature control
  - [Migration]

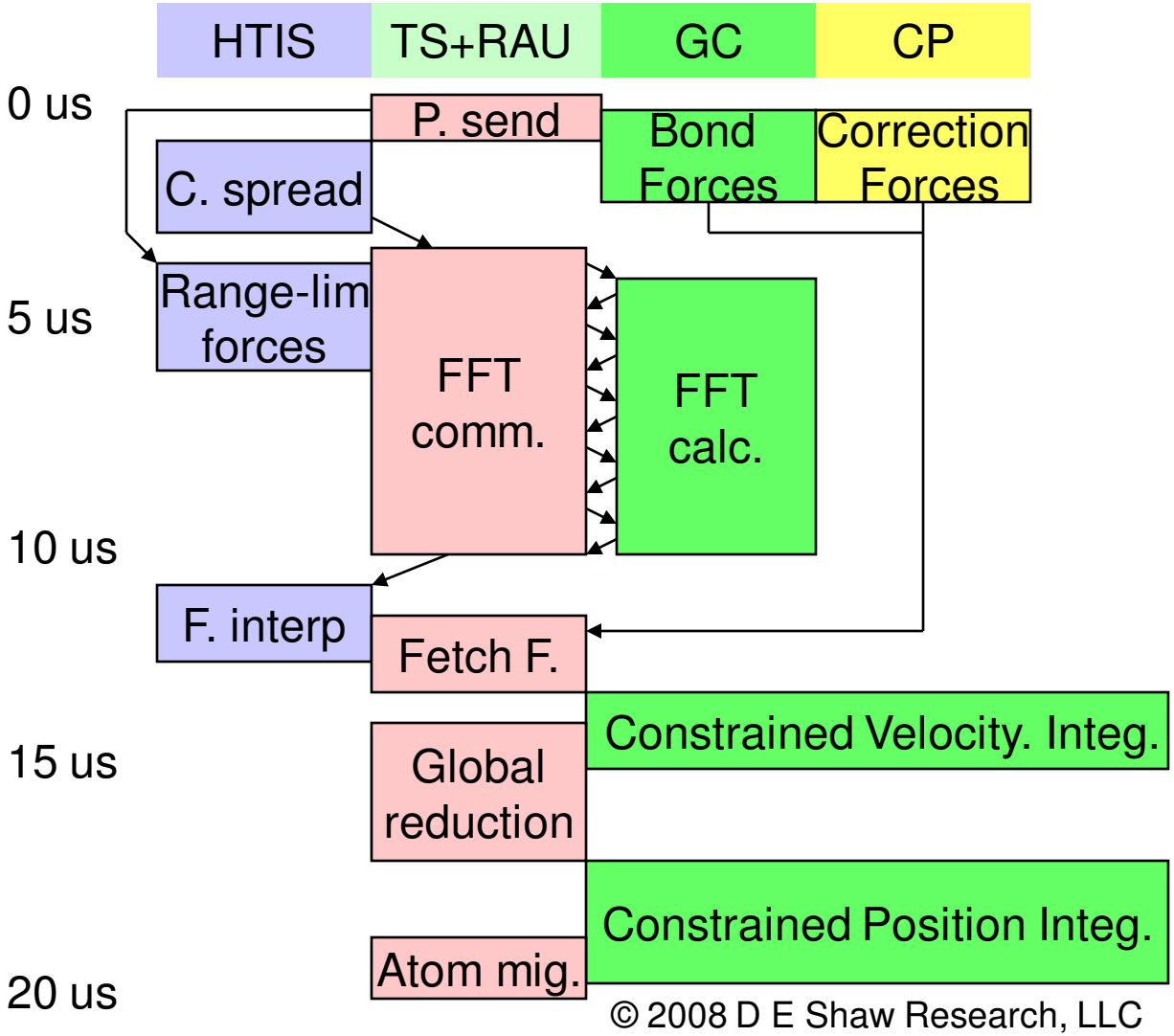
Parallelization Speedups

Task	Anton 512-node speedup over GROMACS	Divided by 512 (single-node speedup)
Range-limited non-bonded forces	50,000	98.5
Charge spreading and force interpolation	5,400	10.5
FFT and Fourier space multiplication	3,700	7.2
Bond forces	4,600	9.0
Correction forces	3,700	7.2
Position and velocity updates	1,200	2.4
Constraint calculations	1,400	2.7
Temperature computation	400	0.8
Total	9,000	17.6

What makes Anton fast?

- Specialization
- Parallelization
- System Integration

Mapping the MD Time Step to Anton



- A long-range time step takes ~20 µsec. (8000 clocks)
- This *time step diagram* gives the task schedule on hardware subsystems
- Schedule minimizes critical path
- We optimize for latency!
  - Not throughput
  - Not utilization

De Rigueur Performance Slide

- Anton performance on a common benchmark system (DHFR (*Dihydrofolate reductase)* 23,558 atoms)
  - Each time step is 2.5 femtoseconds of simulated time
  - Anton performs a pair of time steps in 30 microseconds
  - This yields 14,000 simulated nanoseconds/day
  - Or about 1 millisecond (i.e., 4×10^{11} time steps!!!) of simulated time in 2.5 months

- Our goal: 1000x faster than best MD when we started
  - 2003: NAMD on a fast cluster: 10 **milli**seconds per time step
  - 2008: Anton: 15 **micro**seconds per time step

- Equivalently, 100x better than 5 years of Moore's law

Power Consumption in Anton

512 Node Anton Segment contains:

512 ASICS @ ~75 W =	~38.4 KW
512 DRAM sys. @~30 W =	~15.36 KW
128 node brd. misc. @~50 W =	~ 6.4 KW
– **Total logic**	**~60.2 KW**
– **After DC Conversion**	
440VAC -> 48VDC -> 12VDC -> end use 1- 2 V	
60.2 KW / $(0.9)^3$ = 60.2 / 0.729 =	**~82.5 KW**
Fans, etc. in racks	**~ 4.0 KW**
Total Segment Power	**~ 86.5 KW**
Chilling of Cooling Water	**~ 30 KW**
Grand Total	***~116.5 KW***

It's Not Easy Being Green: Anton *vs* General Purpose Computers

System	MD Performance* (ns/day)	Power Consumption (KW)	kWh/ns
Anton 512 Nodes	14,115	~116.5	0.198
Desmond C on 512 Cores	~260	~44	44.06
Blue Gene L / Blue Matter on 512 Cores	23.75	~39	39.41

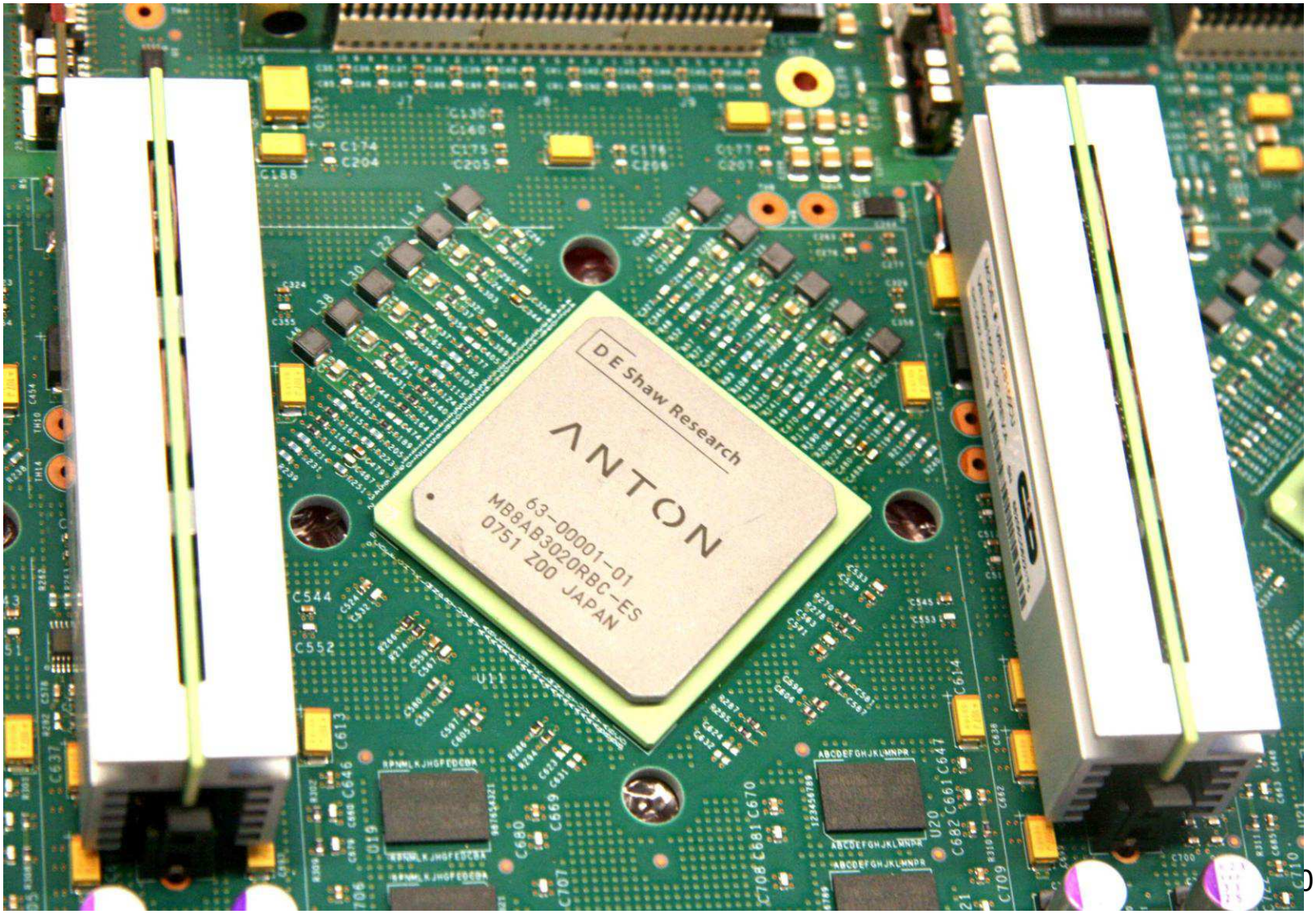
** Production parameters. Dihydrofolate reductase benchmark; 23,558 atoms*

All numbers include power for cooling!

Traditional Conclusions Slide

- *Specialization*, *Parallelization*, and *System Integration* each contribute to Anton's performance.
- Amdahl's law drove the design.
- Anton will enable millisecond MD simulations!
- Power efficiency of specialized design is very good compared with GP systems

Anton ASIC


D E Shaw Research
ANTON
63-00001-01
MB8AB3020RBC-ES
0751 Z00 JAPAN

Anton PCB

Anton ASIC

- 90 NM CMOS – implemented in Fujitsu CS100HP Process
- 17.3 mm x 17.3 mm
- 2108 ball CBGA with ~1000 signal IO
- 33 Million gates, 11 Mbit SRAM (~ 200 million transistors total)
- 6 High Speed IO Channels – 80 GByte/s aggregate bandwidth
- 256 bit wide DDR2-800 DRAM - ~25 GByte/s bandwidth
- ~75 W worst case power dissipation

Anton ASIC (2)

- Flexible Subsystem – 400 MHz
  - 4 Tensilica Cores
  - 8 Geometry Cores – 2 lane VLIW won 4 element SIMD vectors
- High Throughput Interaction Subsystem – 800 MHz
  - 32 PPIMs
  - 1 Tensilica core
- On-chip Ring Network – 400 MHz
  - 51.2 GByte/s bandwidth
  - All subsystems communicate on Ring as peers
- Memory Subsystem
  - 2 DDR2-800 DRAM Controllers
  - Fast atomic Add-to-Memory operations for force accumulation

Tradeoffs in Anton

- Specialized hardwired logic gives enormous advantage in performance and area – >100x relative to General purpose CPU
  - Don't know another way to achieve Anton goals for HTIS functionality
  - Can't deal with certain changes to functional form (*but we don't expect such changes in this calculation*)
  - Need to anticipate at design time variability that will be required later
- Flexible Subsystem is still specialized – Perhaps 10x advantage over GP cores
  - Can compute anything needed, but portability, tool support weak
  - Requires expert programmers, poor programmer productivity
  - Usual system services and protections do not exist