An FPGA Implementation of Reciprocal Sums for SPME

Sam Lee and Paul Chow

Edward S. Rogers Sr.
Department of Electrical and Computer Engineering
University of Toronto

Objectives

- Accelerate part of Molecular Dynamics Simulation
  - Smooth Particle Mesh Ewald
- Implementation
  - FPGA based
  - Try it and learn
- Investigation
  - Acceleration bottleneck
  - Precision requirement
  - Parallelization strategy

Presentation Outline

- Molecular Dynamics
- SPME
- The Reciprocal Sum Compute Engine
- Speedup and Parallelization
- Precision
- Future work

Molecular Dynamics Simulation

Molecular Dynamics

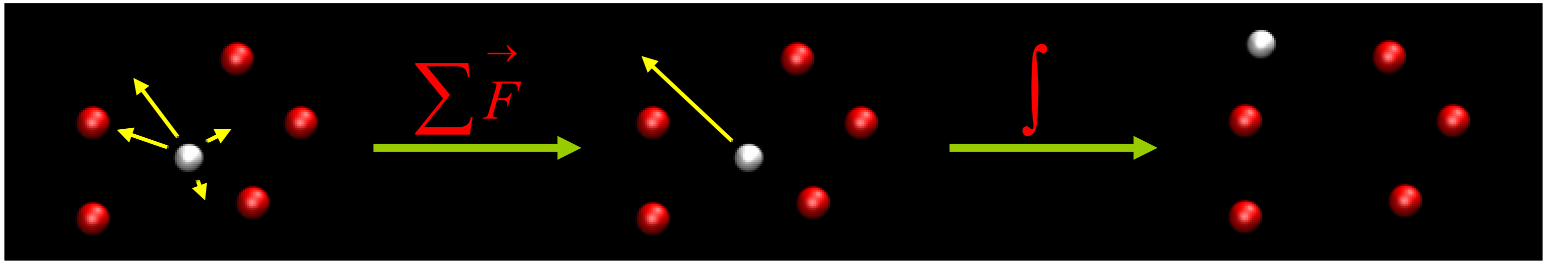
- Combines empirical force calculations with Newton's equations of motion.
- Predict the time trajectory of small atomic systems.
- Computationally demanding.

1. Calculate interatomic forces.
2. Calculate the net force.
3. Integrate Newton's equations of motion.

$$\vec{a} = \vec{F} \cdot m^{-1}$$

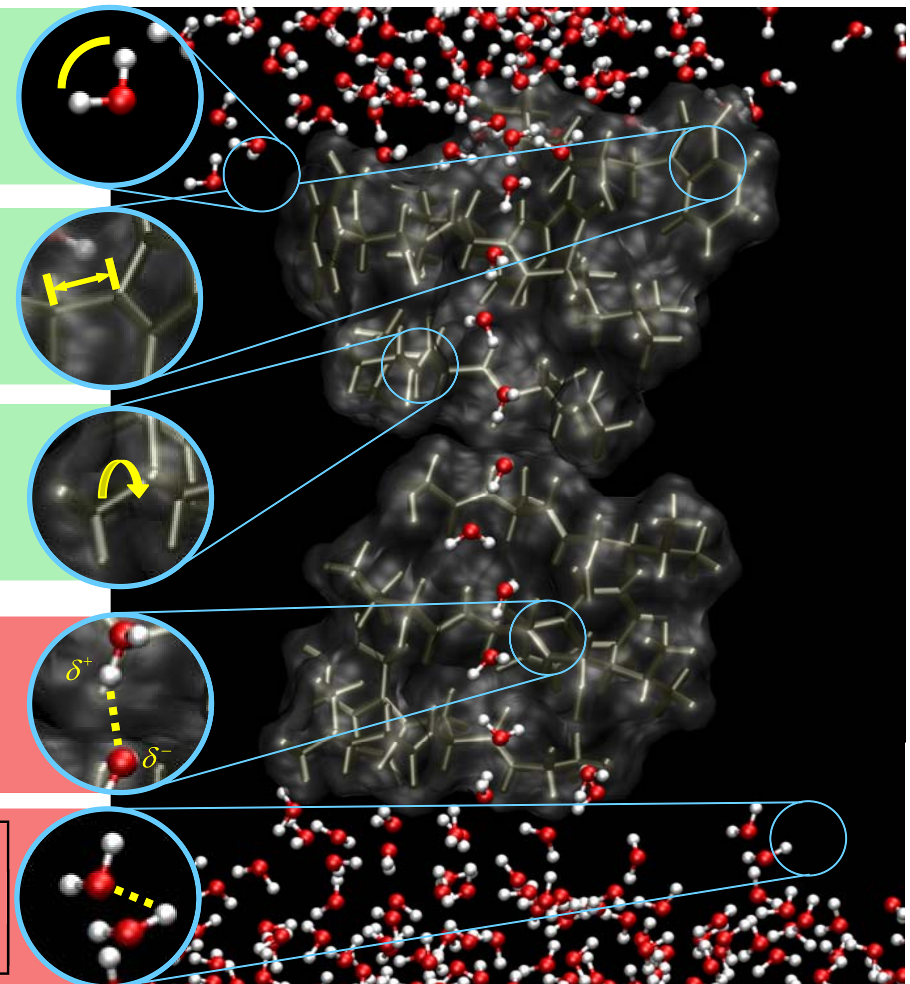
$$\vec{r}(t+\delta t) = \vec{r}(t) + \delta t\,\vec{v}(t) + 0.5\delta t^2\,\vec{a}(t)$$

$$\vec{v}(t+\delta t) = \vec{v}(t) + 0.5\delta t\left[\vec{a}(t) + \vec{a}(t+\delta t)\right]$$

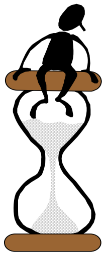


Molecular Dynamics

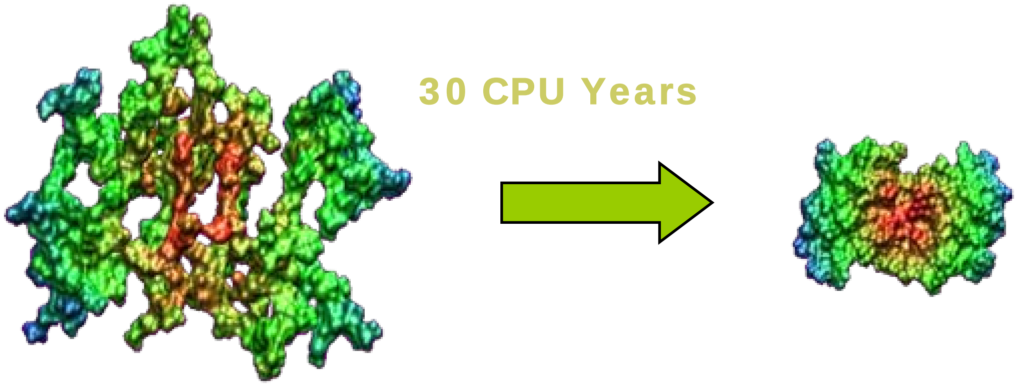
$$U = \left\{ \sum_{All\ Angles} -k_{\Theta}(\Theta-\Theta_o)^2 + \sum_{All\ Bonds} k_b(l-l_o)^2 + \sum_{All\ Torsions} A[1+\cos(n\tau+\phi)] + \sum_{All\ Pairs} \frac{q_1 q_2}{r} + \sum_{All\ Pairs} 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \right.$$



MD Simulation



- Problem scientists are facing:
  - SLOW!
  - O(N^2) complexity.



Solutions

- Parallelize to more compute engines
- Accelerate with FPGA
- Especially: The non-bonded calculations
- To be more specific, this paper addresses:
  - Electrostatic interaction (Reciprocal space)
  - Smooth Particle Mesh Ewald algorithm.

Previous Work

- Software SPME Implementations:
  - Original PME Package written by Toukmaji.
  - Used in NAMD2.

- Hardware Implementations:
  - No previous hardware implementation of reciprocal sums calculation.
  - MD-Grape & MD-Engine uses Ewald Summation.
  - Ewald Summation is $O(N^2)$; SPME is O(NLogN)!

Smooth Particle Mesh Ewald

Electrostatic Interaction

- Coulombic equation:

$$v_{coulomb} = -\frac{q_1 q_2}{4\pi\varepsilon_0 r}$$

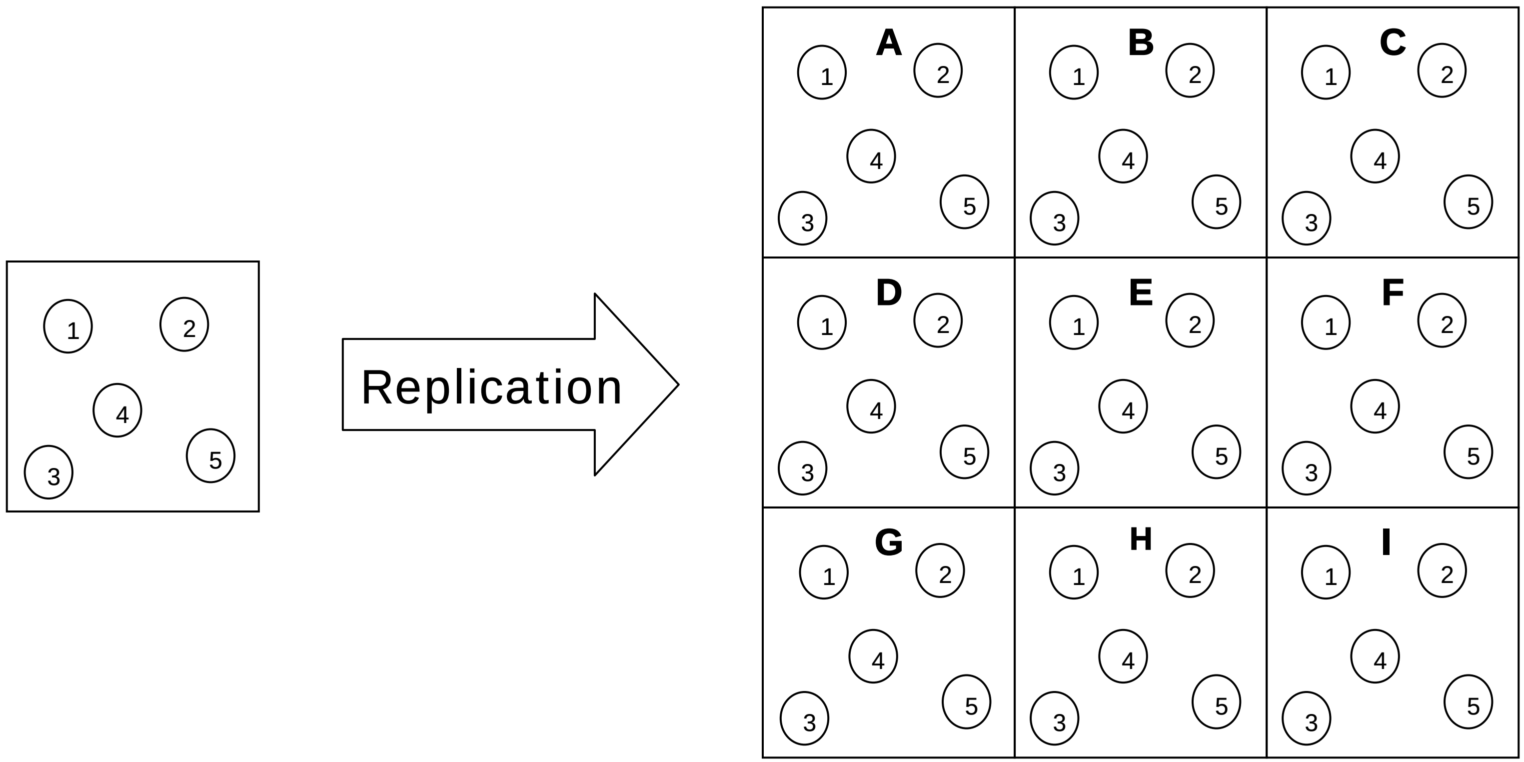
- Under the Periodic Boundary Condition, the summation to calculate Electrostatic energy is only ...

  Conditionally Convergent.

$$U = \frac{1}{2}\sum_{n}^{'}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{r_{ij,n}}$$

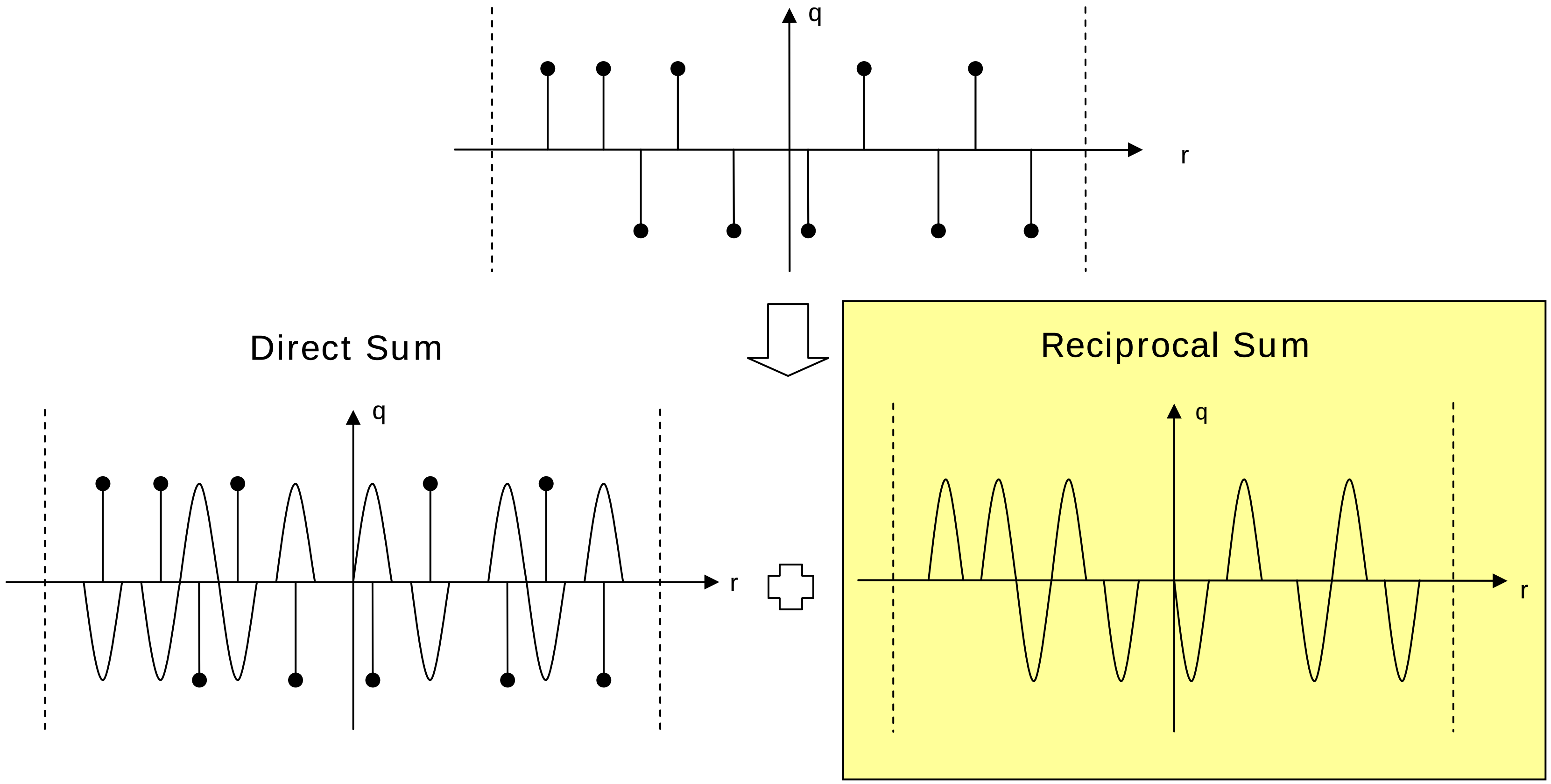
Periodic Boundary Condition

- To combat Surface Effect...



Ewald Summation Used For PBC

- To calculate the Coulombic Interactions
- $O(N^2)$ Direct Sum + $O(N^2)$ Reciprocal Sum



Smooth Particle Mesh Ewald

- Shift the workload to the Reciprocal Sum.
- Use Fast Fourier Transform.
- O(N) Real + O(NLogN) Reciprocal.
- RSCE calculates the Reciprocal Sums using the SPME algorithm.

SPME Reciprocal Contribution

Energy:

$$\tilde{E} = \frac{1}{2\pi V} \sum_{m \neq 0} \frac{\exp(-\pi^2 m^2/\beta^2)}{m^2} B(m_1, m_2, m_3) \bullet \underbrace{F(Q)}_{\text{FFT}}(m_1, m_2, m_3) \underbrace{F(Q)}_{\text{FFT}}(-m_1, -m_2, -m_3)$$

$$\tilde{E} = \frac{1}{2} \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} Q(m_1, m_2, m_3) \bullet (\theta_{rec} * Q)(m_1, m_2, m_3)$$

$$\theta_{rec} = F(B \cdot C) \qquad B(m_1, m_2, m_3) = |b_1(m_1)|^2 \bullet |b_2(m_2)|^2 \bullet |b_3(m_3)|^2$$

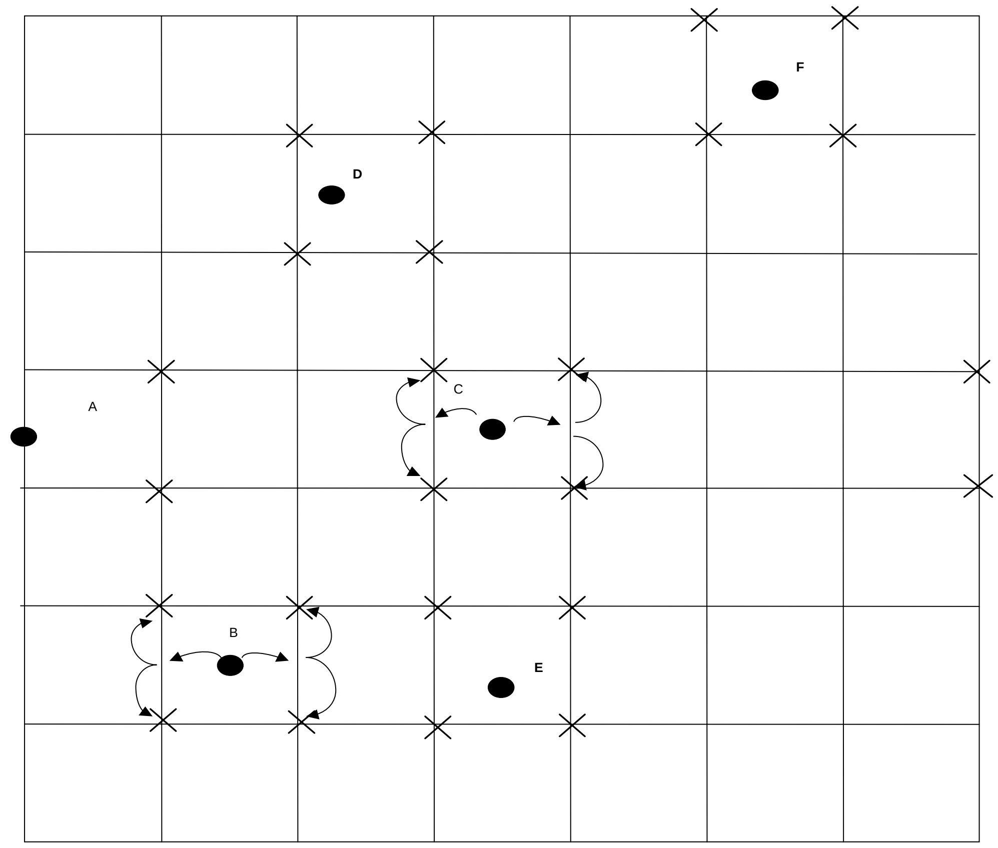
$$b_i(m_i) = \exp\left(\frac{2\pi i(n-1)m_i}{K_i}\right) \times \left[\sum_{k=0}^{n-2} M_n(k+1) \exp\left(\frac{2\pi i m_i k}{K_i}\right)\right]^{-1}$$

$$C(m_1, m_2, m_3) = \frac{1}{\pi V} \frac{\exp(-\pi^2 m^2/\beta^2)}{m^2} \qquad m \neq 0, c(0,0,0) = 0$$

Force:

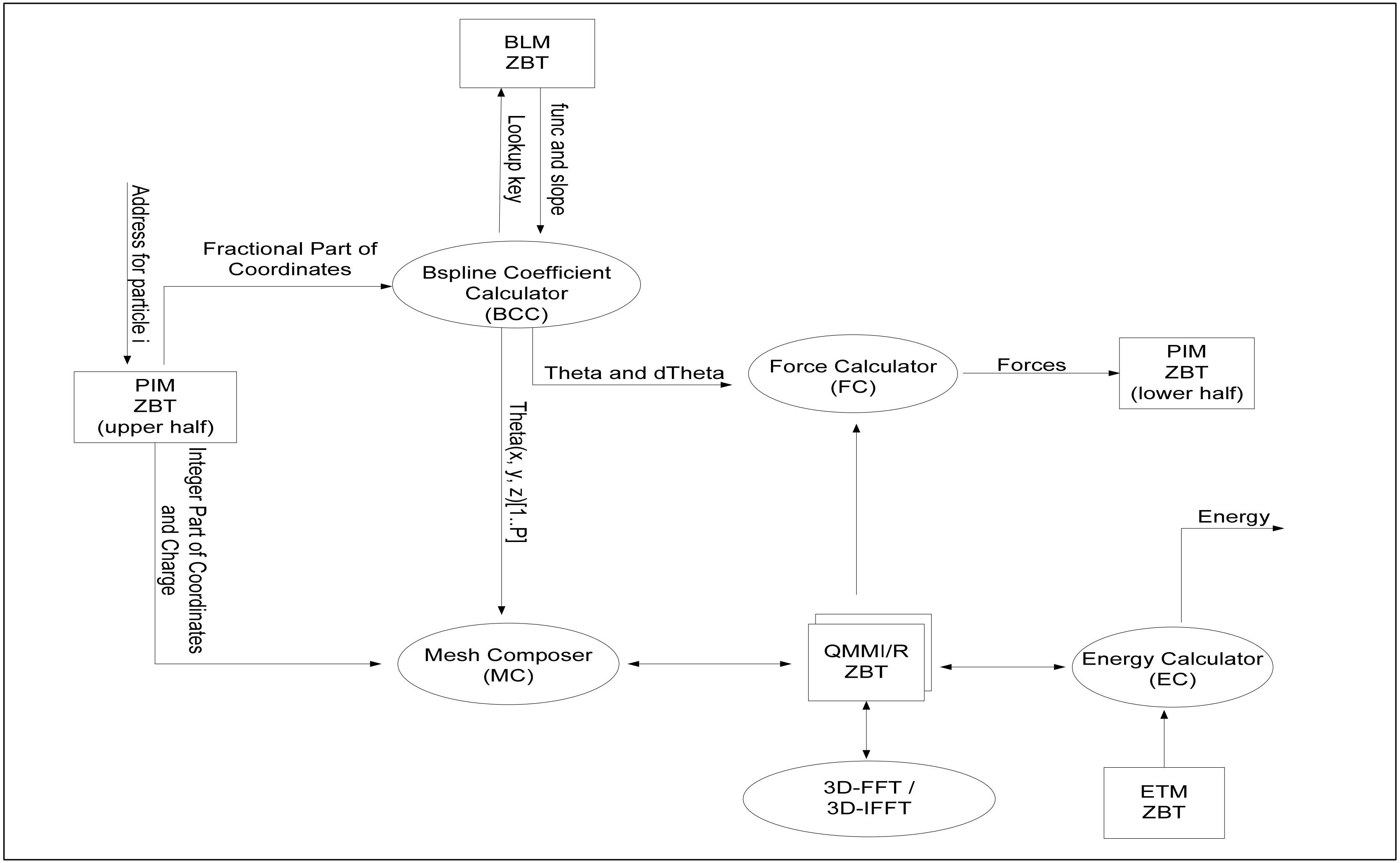
$$F = \frac{\partial \tilde{E}_{rec}}{\partial r_{\alpha i}} = \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} \frac{\partial Q}{\partial r_{\alpha i}}(m_1, m_2, m_3) \bullet (\theta_{rec} * Q)(m_1, m_2, m_3)$$

Charge Interpolation

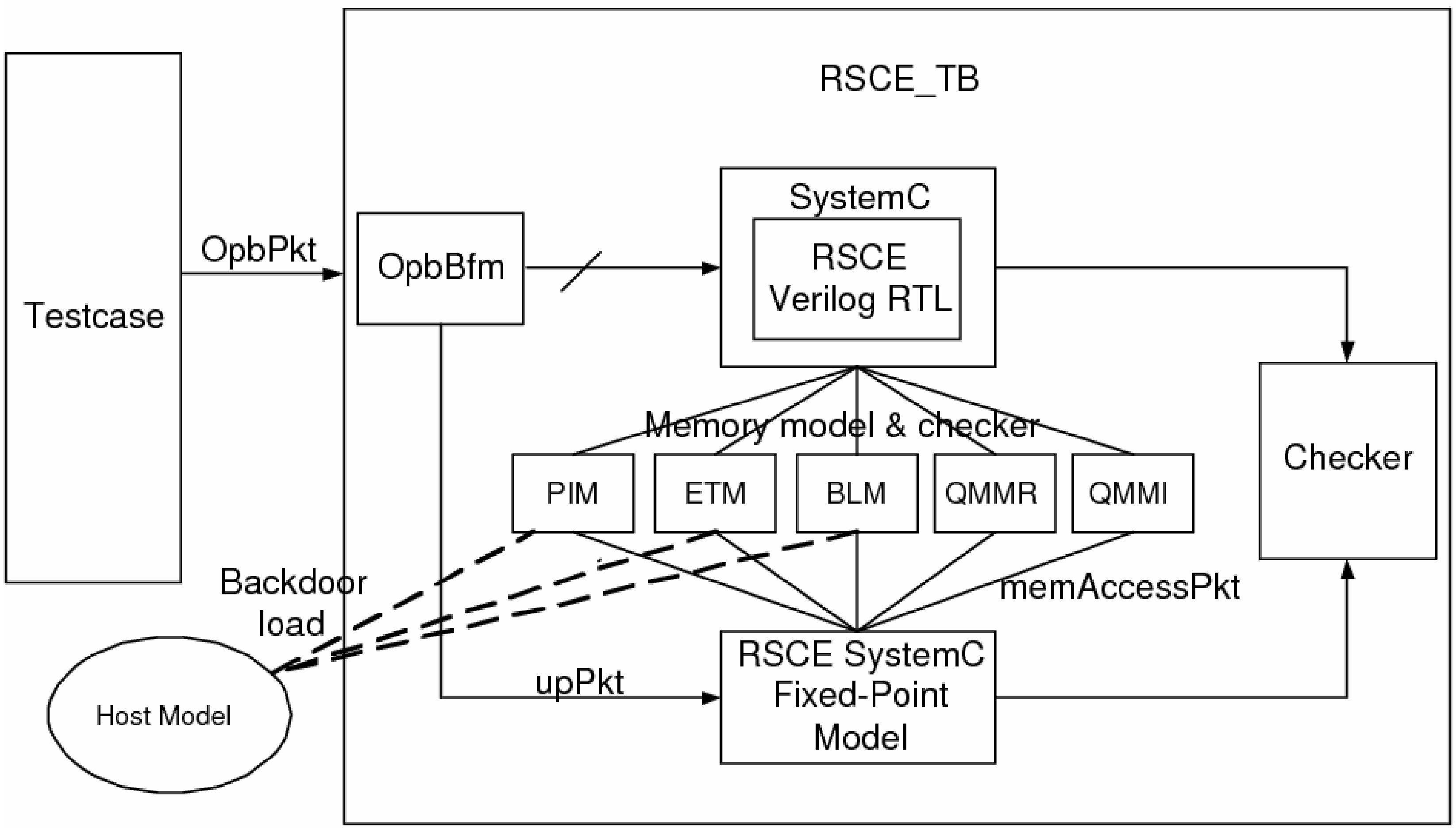


Reciprocal Sum Compute Engine

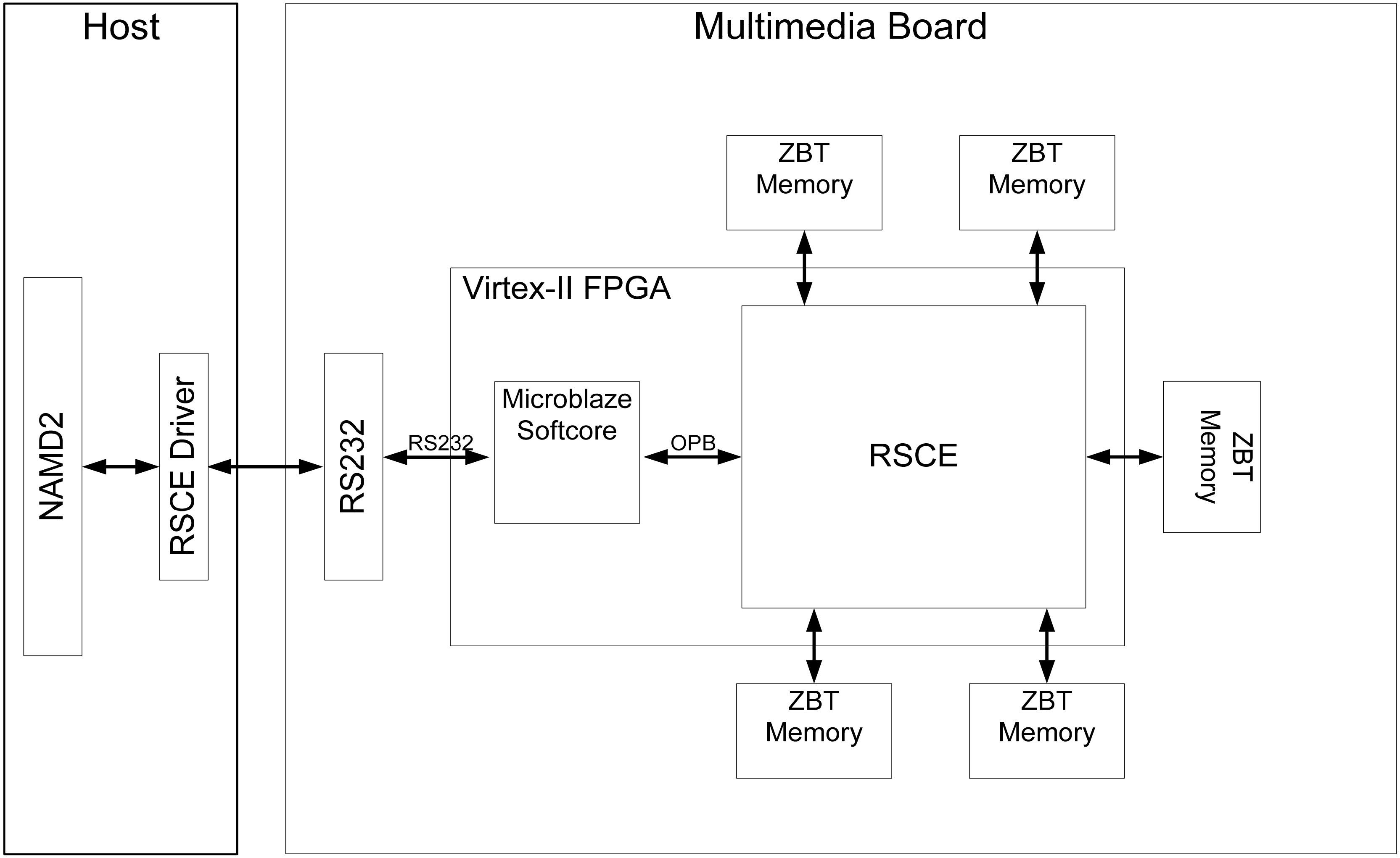
RSCE Architecture



RSCE Verification Testbench



RSCE Validation Environment



Speedup Estimate

RSCE vs. Software Implementation

RSCE Speedup

- RSCE @ 100MHz vs. P4 Intel @ 2.4GHz.
  - Speedup: 3x to 14x
- Why so insignificant?
  - Reciprocal Sums calculations not easily parallelizable.
  - QMM memory bandwidth limitation.
- Improvement:
  - Using more QMM memories can improve the speedup.
  - Slight design modifications are required.

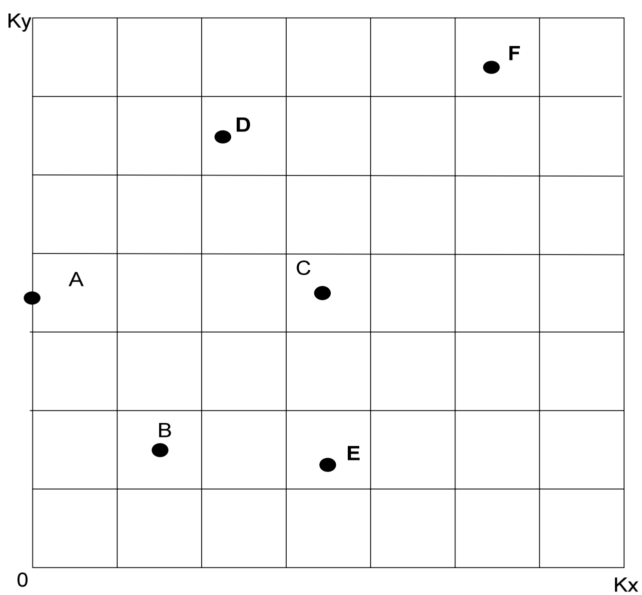
Parallelization Strategy

Multiple RSCE

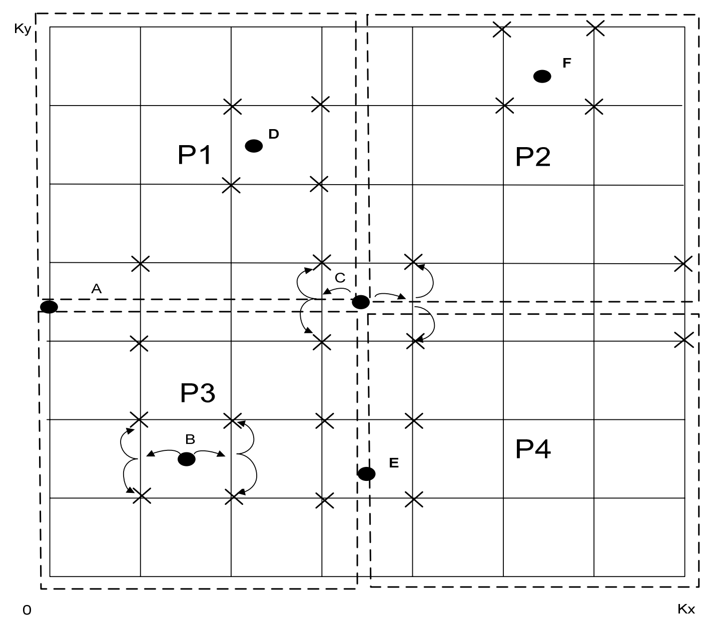
RSCE Parallelization Strategy

- Assume a 2-D simulation system.
- Assume P=2, K=8, N=6.
- Assume NumP = 4.

An 8x8x8 mesh

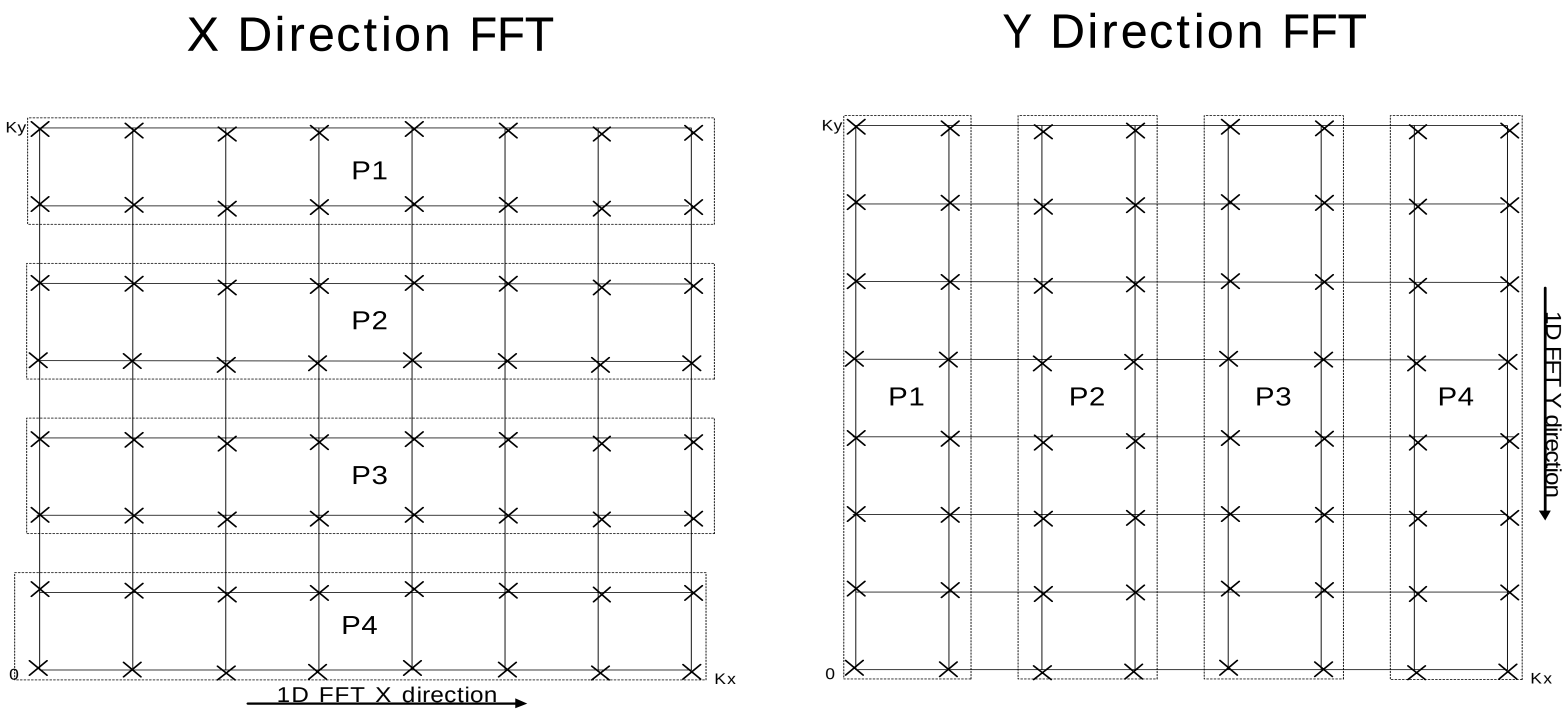


Four 4x4x4 Mini Meshes



RSCE Parallelization Strategy

- Mini-mesh composed -> 2D-IFFT
- 2D-IFFT = two passes of 1D-FFT (X and Y).

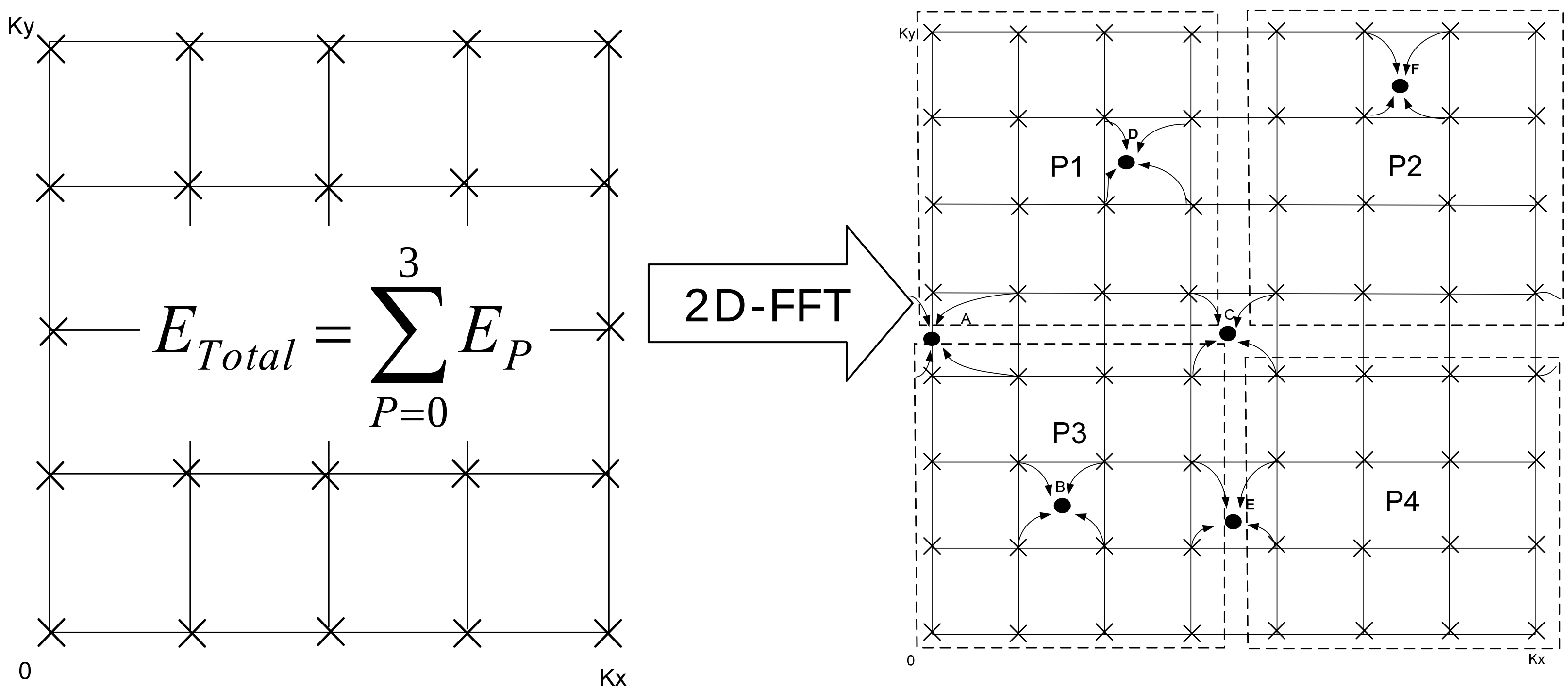


Parallelization Strategy

- 2D-IFFT -> Energy Calculation -> 2D-FFT
- 2D-FFT -> Force Calculation

Energy Calculation

Force Calculation



$$E_{Total} = \sum_{P=0}^{3} E_P$$

MD Simulations
RSCE + NAMD2

RSCE Precision

- Precision goal: Relative error bound $< 10^{-5}$.
- Two major calculation steps:
  - B-Spline Calculation.
  - 3D-FFT/IFFT Calculation.
- Due to the limited logic resource & limited precision FFT LogiCore.

  => Precision goal cannot be achieved.

RSCE Precision

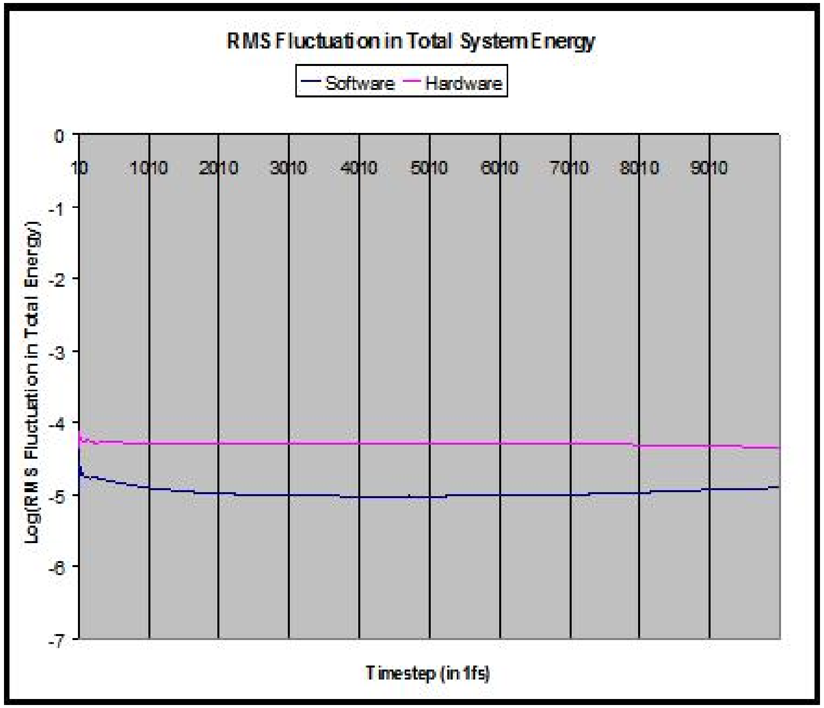
- To achieve the relative error bound of $< 10^{-5}$.
- Minimum calculation precision:
  - FFT {14.30}, B-Spline {1.27}

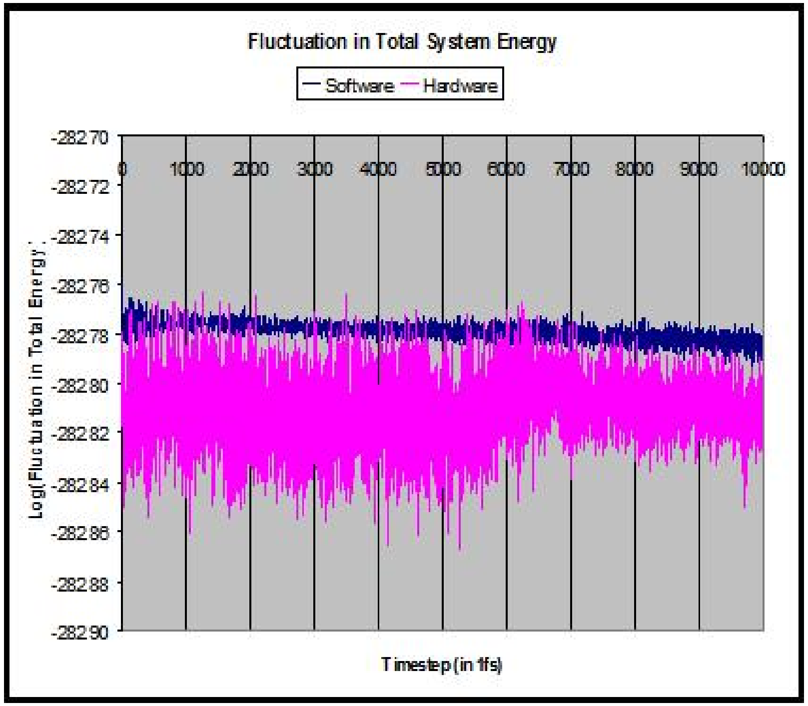
FFT SFXP	BSP SFXP	Avg. Energy Relative Error (E-06)	Max. Energy Relative Error (E-06)	Avg. Force RMS Relative Error (E-06)	Max. Force RMS Relative Error (E-06)
14.26	1.21	0.755	2.12	36.7	476
14.26	1.24	0.687	1.45	4.01	39.2
14.26	1.27	0.611	1.17	3.35	20.8
14.26	1.30	0.601	1.03	2.06	11.2
14.30	1.21	0.716	1.78	5.29	827
14.30	1.24	0.581	1.00	3.98	38.9
14.30	*1.27*	*0.599*	*1.28*	*1.98*	*9.80*
14.30	1.30	0.570	1.01	1.70	7.80

MD Simulation with RSCE

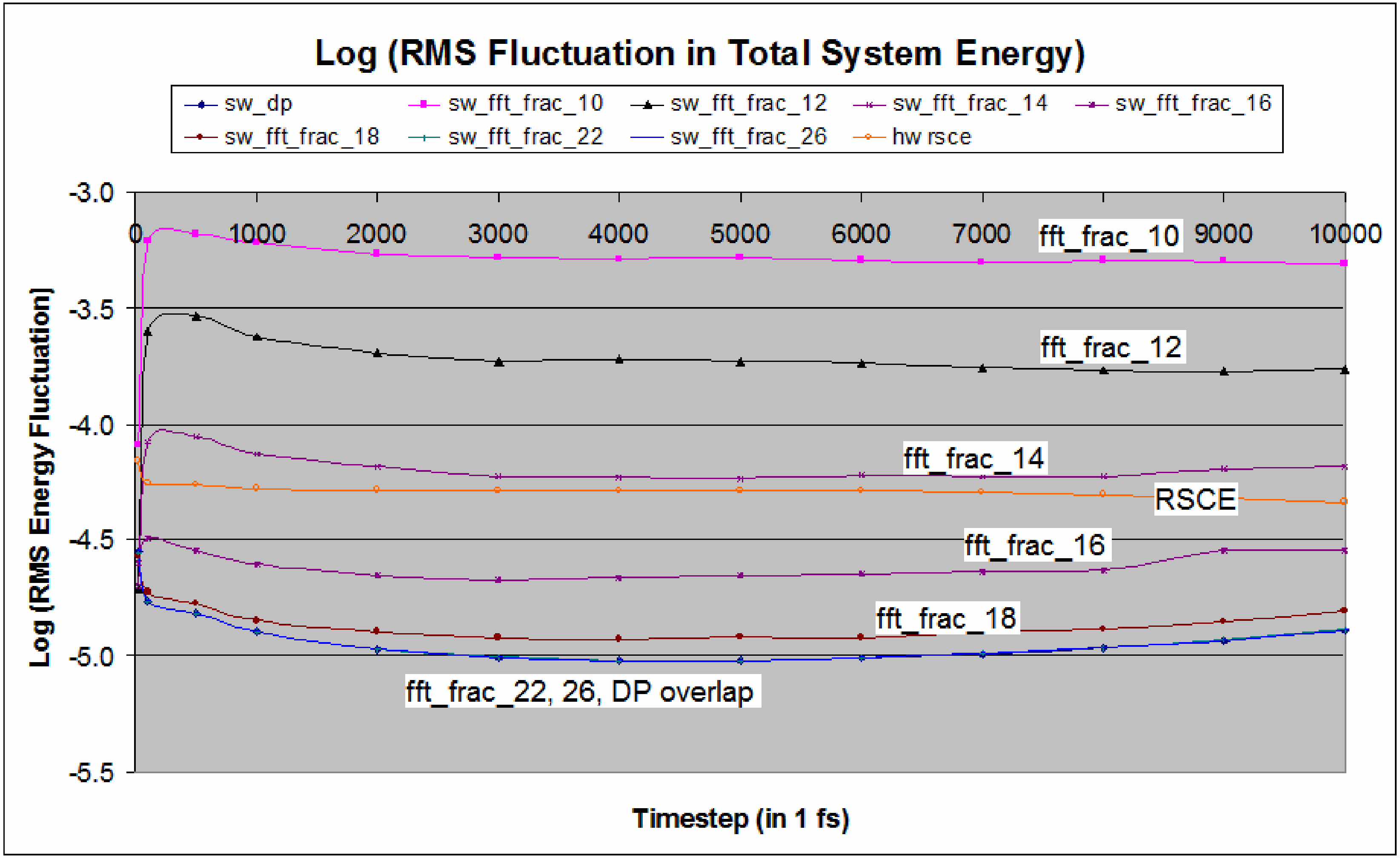
- RMS Energy Error Fluctuation:

$$RMS\ \ Energy\ \ Fluctuation = \frac{\sqrt{\left|\langle E^2 \rangle - \langle E \rangle^2\right|}}{\left|\langle E \rangle\right|}$$





FFT Precision Vs. Energy Fluctuation



Summary

- Implementation of FPGA-based Reciprocal Sums Compute Engine and its SystemC model.
- Integration of the RSCE into a widely used Molecular Dynamics program called NAMD2 for verification
- RSCE Speedup Estimate
  - 3x to 14x
- Precision Requirement
  - B-Spline: {1.27} & FFT: {14:30} => 10^{-5} rel. error
- Parallelization Strategy

Future Work

- More in-depth precision analysis.
- Investigation on how to further speedup the SPME algorithm with FPGA.

Questions