Regular Paper

Design and Preliminary Implementation of a Particle Simulation Machine for Efficient Short-range Interaction Computations

Ryo Takata,[†,††] Kenji Kise,[†] Hiroki Honda[†]
and Toshitsugu Yuba[†]

This paper describes the architecture of a high performance particle simulation machine, DEM-1 (Dentsudai EAM Machine-1), for short-range particle interaction computations. All existing particle simulation machines have specialized pipelines to calculate long-range particle interactions effectively. However, their ability to perform particle simulations efficiently diminishes with short-range interactions. Communication cost in particle simulations will play a significant role in performance when the computation cost becomes O(N), where N is the number of particles. In DEM-1 a three-dimensional torus high-speed network reduces this cost while 2048 local processors perform time integration. Specially designed cutoff units reduce the number of calculations in DEM-1. Each specialized pipeline consists of a dedicated data path supported by position vector pre-fetch dual ported memory (pre-fetch DPM). A prototype pipeline has been implemented on an FPGA test-bed. Performance of the DEM-1 pipeline has been measured and total system performance has been estimated with very large-scale Embedded Atom Method (EAM) molecular dynamics simulations.

1. Introduction

Particle simulation has been widely used in various fields such as astrophysics, molecular dynamics, gas-solid flows, and computer graphics. These simulations can be executed in parallel, because most of computation time is spent on calculation of interactions that can be evaluated simultaneously. The machine GRAPE [1] has been developed to reduce execution time of large-scale astrophysical simulations. The machines MD GRAPE [2], MD One [3], MD Engine [4], MD Engine II [5], and MDM [6] have been developed to reduce the execution time of molecular dynamic simulations of protein and other particle simulations. These parallel particle simulation machines have specialized parallel hardware pipelines to calculate arbitrary central forces to increase performance.

In certain particle simulations, short-range cutoff interactions dominate system behavior. Molecular dynamics simulations of metallic solids using the Embedded Atom Method, EAM, fall into this category of computations. Available particular simulation machines are not optimized to compute such short-range interactions efficiently.

To show the possibility that hardware integration will enable high-speed implementation of these calculations, MD One/E [7] was developed from 1998 to 1999 in cooperation between Image Technology Laboratory Corporation and the Central Research Institute of Electric Power Industry. This machine is based on MD One [3] and is provided with hardware for supporting the linked cell list addressing. However, its improvement is limited to linked cell list addressing support.

We previously proposed a particle simulation machine DEM-1 optimized for such applications [8],[9]. We have now implemented a prototype pipeline on an FPGA test-bed and have observed a pipeline performance improvement of 6 over available particle simulation machines.

Section 2 describes the background and motivation. Section 3 focuses on the architectural details of DEM-1. Section 4 summarizes the prototype implementation. Section 5 summarizes the performance evaluation of DEM-1. Section 6 summarizes related work. Finally, section 7 concludes this paper.

2. Background and Motivation

2.1 Application

The Embedded Atom Method, EAM [10], is widely used in molecular dynamics simulations of metallic solids. In EAM, the potential energy is calculated by using the following equation;

[†] Graduate School of Information Systems, University of Electro-Communications
[††] Image Technology Laboratory Corporation

$$E_{tot} = \sum_t F_i(\rho_{h,i}) + \frac{1}{2}\sum_{i,j}\phi_{i,j}(r_{i,j}) \quad (1)$$

where, E_{tot} is the total energy of the metal, F_i is the embedding energy of atom i, $\rho_{h,i}$ is electron density at atom i, ϕ_{ij} is repulsive potential between atoms i and j, r_{ij} is relative distance between atoms i and j. Electron density $\rho_{h,i}$ is calculated by Equation (2);

$$\rho_{h,i} = \sum_j \rho_j(r_{ij}) \quad (2)$$

where, ρ_j is electron density attributed to atom j. The force exerted on atom i, $\vec{f_i}$, is represented by;

$$\vec{f_i} = -\sum_{j(\neq i)}(F'(\rho_i)\rho'_a(r_{ij}) + F'(\rho_j)\rho'_a(r_{ij}) + \phi'(r_{ij}))\vec{r}_{ij}/|r_{ij}| \quad (3)$$

where, F', ρ', ϕ' are derivatives of F, ρ, ϕ, respectively and r_{ij} is the relative distance vector between atoms i and j.

Usually, a linked cell list method (see **Fig. 1**) is used to reduce computational cost of interaction calculations to $O(N)$. In the linked cell list method, simulation space is divided into small regions called "cells." A cell is a cube (or a rectangular solid) of length l_{cell} on a side, which is represented by the following equation;

$$l_{cell} = r_{cut}(1+\varepsilon)/N_{rc} \quad (4)$$

where r_{cut} is the cutoff radius of interactions, ε is a small number much less than unity, and N_{rc} is a natural number. Increasing the N_{rc} value will cause the memory usage and the cell list operation overhead to be increased. In particular, $N_{rc} > 1$ is not desirable for such a simulation with a small r_{cut}, as in EAM, as it will generate many empty cells. The DEM-1 architecture can accommodate an arbitrary N_{rc} value whereas the following argument assumes $N_{rc} = 1$.

For example, consider a simulation of BCC iron. The cutoff radius for such a computation is 3.46 Angstrom. Typically four particles exist inside a cell. So approximately 100 particles have a chance to interact with the target particle in the linked cell list method. However, only 10-20 particles actually exert force on the target particle.

The simulation parameters of the computation vary according to the dimensions of the target system and the objective of the simulation. Radiation damage simulation of iron requires more than one million atoms and ten thousand steps. Creep or crack growth simulations require over 100 million atoms.

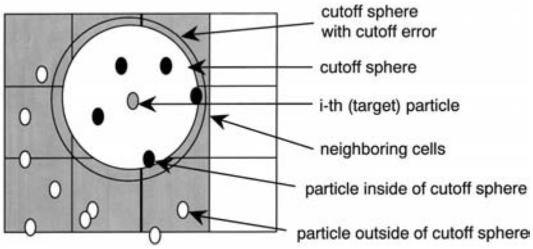

cutoff sphere with cutoff error
cutoff sphere
i-th (target) particle
neighboring cells
particle inside of cutoff sphere
particle outside of cutoff sphere

Fig. 1 Linked cell list method and cutoff sphere (2-dimensional view).

In EAM simulations with MD One/E [7], two potentials (ρ, ϕ) and two forces (ρ', ϕ') are calculated by the hardware pipeline, while $F'(\rho)$ calculations or time step integration such as particle position vector updates are performed by the host computer.

2.2 Performance Bottleneck Analysis

We analyzed performance of MD One particle simulation machine with a model [11];

$$T_{step} = T_{pre} + T_w + T_{calc} + T_r + T_{other} + T_{move} + T_{ovh} \quad (5)$$

where T_{step} is the overall time required for a step, T_{pre} is the initialization time of MD One, T_w is the transfer time of position vectors and coefficients, T_{calc} is the pipeline calculation time (including target particle position vector write time), T_r is the result read time, T_{other} includes other calculation time of bond interactions or center of mass velocity correction, etc., T_{move} measures the time to update position vectors, velocities, etc., and T_{ovh} measures other overheads such as logging in, respectively.

All existing particle simulation machines assume that the calculation time dominates total performance.

$$T_{calc} \gg T_{pre} + T_w + T_r + T_{other} + T_{move} + T_{ovh} \quad (6)$$

This may be true for simulations of interactions whose calculation costs are $O(N^2)$ or $O(N^{1.5})$. Here N is the number of particles in the system. In many molecular dynamics applications such as EAM, the calculation cost of interactions is $O(N)$ and the effects of other terms cannot be neglected. Most of these costs can be attributed to: (1) communication time between the host computer and the hardware pipelines, (2) execution time of calculations related to the portion of simulations other than the interaction calculation.

In addition to these bottlenecks, most of the existing particle simulation machines have in-efficient pipelines for computations related to particle pairs encountered in the linked cell list method particle search. These pipelines use approximately 13 times more operations than software calculations (see Appendix A.1). This reduces the effective performance of existing particle simulation machines by one order of magnitude. The MD Engine[4] has a facility to utilize generated pair lists without software overhead, but its performance improvement factor did not exceed 2. Thus, it appears possible to achieve another performance improvement of 6.44.

As described in section 1, MD One/E[7] has been developed to show the possibility that hardware integration will enable high-speed implementation of these calculations. The hardware works to reduce T_{calc} and T_r in Equation (5) by eliminating the software overhead to handle pipeline operations.

Although MD One/E is provided with a function for increasing the speed of the calculation with a cutoff and computational cost $O(N)$, it has certain problems represented by the following:

(1) Sharing of Particle Coordinates Given to the Pipeline

On the MD One/E simulation machine, pipeline data common to four MD Chips is supplied from a single particle memory mounted on the board. When the effect of the six virtual multi-pipelines in those MD Chips is considered, up to 24 particles (equivalent to six cells on average) will be calculated in parallel (**Fig. 2**). The particles flowing in the pipeline must be a logical sum of sets of particles neighboring these 24 particles. Therefore, although it was originally required only to calculate the interaction for 27 cells, the computation will now be executed for 72 cells (3*3*(6+2)=72), reducing the computation efficiency to 3/8.

(2) Register I/O Overhead

MD One/E carries out the particle coordinate write, pipeline calculation, and calculation result read in sequential order. In general molecular dynamics simulations, the contribution of thousands to several ten thousands of particles is calculated during the pipeline computation. Since the pipeline computation time is sufficiently long, the register I/O overhead does not adversely influence the calculation performance. However, when the interaction is calculated with a small cutoff radius as in EAM, the register I/O time is nearly equivalent to the pipeline computation time, so that the register I/O overhead will reduce the calculation performance.

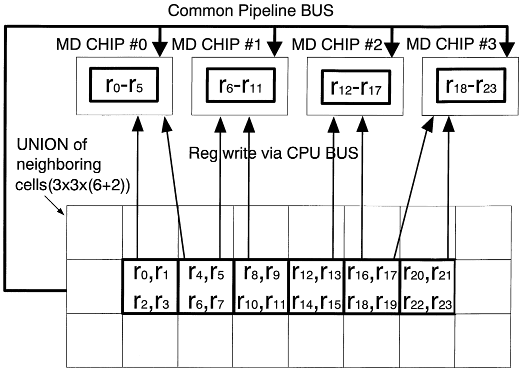


Fig. 2 Pipeline overhead introduced by data sharing (2-dimensional view).

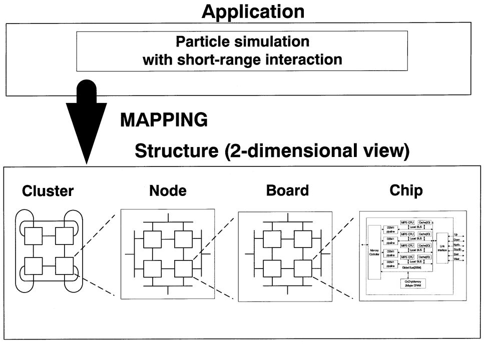

Application
Particle simulation with short-range interaction
MAPPING
Structure (2-dimensional view)
Cluster
Node
Board
Chip

Fig. 3 Structure of DEM-1 system.

3. Architecture of DEM-1

3.1 Design Policy

The structure of the DEM-1 system is illustrated in **Fig. 3**. DEM-1 consists of communicating microelements, the DEM-1 chip, each integrates four MIPS microprocessors, four hardware pipelines and a link interface along with external memory. Eight elements are connected together to create a higher-level structure. Thus, three levels of integration are defined in DEM-1 system configurations. These higher-level elements always have faster interconnections than the lower-level elements. DEM-1 has a three-dimensional torus network at its top level and in which the simulation space can be mapped on to it spatially. The processors, located adjacent to the hardware pipelines, work in parallel to accelerate $O(N)$ tasks such as time integration. In addition to that, proximity of local processors makes the local time step integration possible, resulting in

Table 1 Three configurations of DEM-1.

Configuration	Pipelines	Processors	Performance	Memories	Link speed	Link
			(Gflops)	(Gbytes)	(Mbytes/sec)	interface
Board	32	32	115.2	2	133*6	LVDS
Node	256	256	921.6	16	1000*3	RHiNET2
Cluster	2048	2048	7373	128	(2000)	RHiNET2

a large communication cost reduction.

DEM-1 is equipped with specialized pipelines to calculate short-range interactions of particles. Even though the calculation cost is only $O(N)$, each potential or force calculation requires as many as 30 arithmetic operations. More than 90% of the total computation time is spent on such computations. These calculations can be accelerated by parallel hardware pipelines, as in existing particle simulation machines. Naturally, care must be taken not to block operations by external and/or internal bus bottleneck when designing the pipeline. DEM-1 integrates dual ported memory and prefetches position vectors (and coefficients) of the interacting particles. To eliminate unnecessary calculations at the main pipeline, we introduce multiple cutoff units before it.

DEM-1 can be provided to users in three configurations:

(1) A Compact PCI add-on board.
(2) A node of up to eight DEM-1 boards.
(3) A cluster consisting of up to eight DEM-1 nodes.

System specifications of DEM-1 are listed in **Table 1**.

Figure 4 describes these three different configurations and their hierarchical connections. By adding one DEM-1 board to a PC, particle simulations can reach performance over 100 Gflops. A DEM-1 node in a 6U system rack arrangement could provide 1 Tflops-computing speed by the side of a desk.

The DEM-1 pipelined processor chip is designed to achieve the goals outlined above. The DEM-1 pipeline is upward compatible with MD-GRAPE chip[2] and can calculate arbitrary central forces. Therefore, DEM-1 should be able to execute most of the particle simulations running on MD-GRAPE[2] or MD One[3] such as molecular dynamics simulations of proteins. A DEM-1 Chip integrates four pipelines and four processors to achieve 14.4 Gflops effective performance (see Appendix A.2).

3.2 Detailed Design

3.2.1 DEM-1 Cluster

A DEM-1 cluster consists of eight DEM-1 nodes connected via three RHiNET2-SWs[12]. Each node can communicate simultaneously with three other nodes, with each connection operating at approximately 1 Gbyte/sec. An additional RHiNET2-SW connects the cluster to an external network.

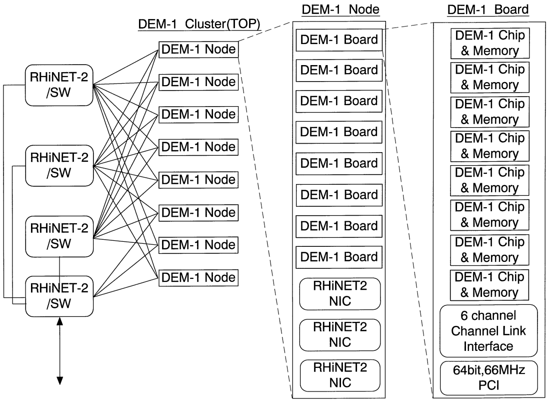


Fig. 4 Global architecture of DEM-1.

3.2.2 DEM-1 Node

Up to eight DEM-1 boards are tightly connected to make a DEM-1 node. Each node has a maximum of three RHiNET2 NICs to communicate with other computers/storages or DEM-1 nodes.

3.2.3 DEM-1 Board

A DEM-1 board integrates eight DEM-1 chips, each with its own particle memory, a switch-FPGA with six channel-link interface transceivers and a 64 bit, 66 MHz PCI interface. A DEM-1 board has a peak performance of 115.2 Gflops. Its total memory can store 8,000,000 particles. The PCI interface is used to communicate with a host CPU in a single board configuration. Up to 256 Mbytes of DDR FCRAM attached to each DEM-1 chip is used to store position vectors, velocities, accelerations, derivative of accelerations, coefficients and linked cell lists. Each of the six LVDS channel-link interfaces connected to the switch FPGA provides a transfer rate of 133 Mbytes per second to each channel in each direction.

3.2.4 DEM-1 Chip

Figure 5 shows a block diagram of the DEM-

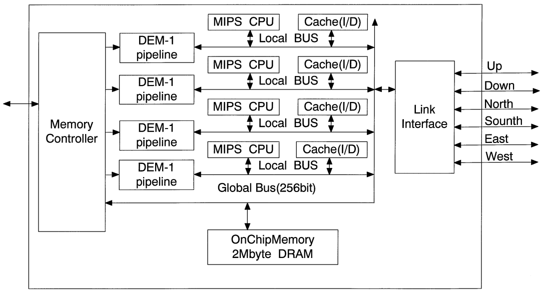


Fig. 5 DEM-1 chip block diagram.

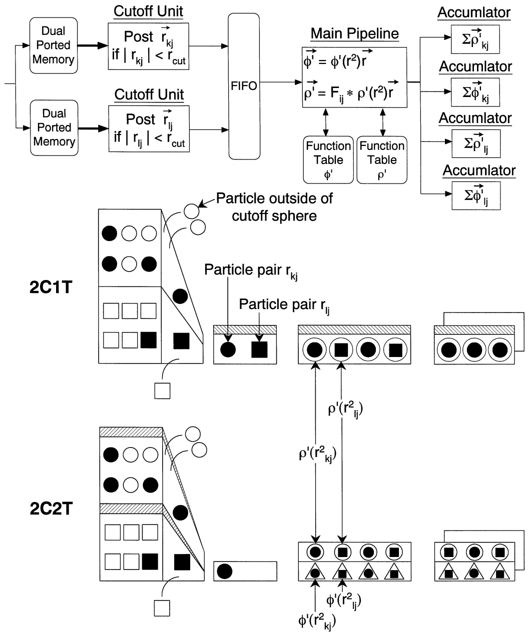


Fig. 6 The DEM-1 pipeline and its data flow. The symbol 2C2T corresponds to the DEM-1 pipeline configuration.

1 chip. The DEM-1 chip integrates four hardware pipelines with dedicated on-chip processors and cache memories, a memory controller which handles linked cell list addressing, an on-chip DRAM memory and local link interface in a single LSI. The 8-bytes wide memory bus of a DEM-1 chip can transfer 16 bytes per main CLK cycle with its DDR FCRAM interface. It has two memory banks having a transfer rate of 2.13 Gbytes/sec each. The first bank stores linked cell list, coordinate vectors, $F'(\rho)$ and such. The second bank stores other particle information such as velocity vectors or acceleration vectors.

3.2.5 DEM-1 pipeline

A block diagram of the DEM-1 pipeline is shown in **Fig. 6**. Design of the DEM-1 pipeline is based on the MD GRAPE CHIP[2)] with two major enhancements: (1) Cutoff units and two function tables, (2) position vector pre-fetch dual ported memory. The DEM-1 pipeline integrates two pre-fetch DPMs, two cutoff units, a cutoff FIFO, which has two write ports and a single read port, a main pipeline, four accumulators and a result FIFO (not shown in Fig. 6). The main pipeline integrates a relative distant vector unit, a function generator with two function tables and a vector-scalar multiplier. Three types of Linked Cell list Read Unit (LCRU) support linked cell list addressing in the pipeline operation. Pipeline structures and operations are described in the next section.

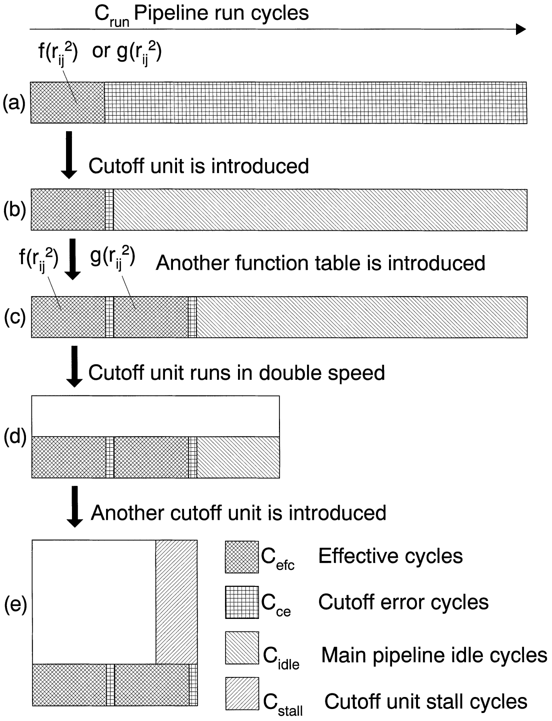

C_{run} Pipeline run cycles
$f(r_{ij}^2)$ or $g(r_{ij}^2)$
(a)
Cutoff unit is introduced
(b)
$f(r_{ij}^2)$
$g(r_{ij}^2)$
Another function table is introduced
(c)
Cutoff unit runs in double speed
(d)
Another cutoff unit is introduced
(e)
C_{efc} Effective cycles
C_{ce} Cutoff error cycles
C_{idle} Main pipeline idle cycles
C_{stall} Cutoff unit stall cycles

Fig. 7 Effective performance improvements by cutoff units and two function tables.

3.3 Pipeline Operations

3.3.1 Cutoff Units and Two Function Tables

As described in section 2.2, the effective performance of the pipelines of existing particle simulation machines are 1/6.44 of their peak performance, because they calculate particles outside of the cutoff sphere. The DEM-1 pipeline is designed to achieve effective performance nearly equal to its peak performance by eliminating the calculation of unnecessary particle pairs with little increase in hardware resource utilization.

Figure 7 describes the effect of the additional function table and the cutoff units. Additional function table was introduced at the early design stage of the DEM-1 pipeline[8]. Figure 7 (a) through (c) describe how the additional function table works to improve performance.

Figure 7 (a) illustrates the pipeline operation of existing particle simulation machines (referred as "reference pipeline" hereafter). It needs two pipeline run cycles to calculate two types of interactions. Most of the pipeline cycle time is spent on calculating particle pairs outside of cutoff sphere (C_{ce}). Figure 7 (b) illustrates how cutoff judgment affects the behavior of the pipeline. The overall execution time remains the same because the pipeline has to search through particle pairs as the reference pipeline does. However, the pipeline can now distinguish cutoff error cycles (C_{ce}), from effective cycles (C_{efc}) and they are labeled as idle cycles (C_{idle}). Additional function table makes it possible to finish the calculation of two interactions within a single particle search by using the part of the idle cycles (Fig. 7 (c)), resulting in relative performance gain of two compared to the reference pipeline. The main pipeline calculates 2 interactions of a particle pair with 2 clock cycles. This reduces the cutoff FIFO read rate to 0.5 particle pair per clock.

Adding a function table costs only 16 Kbytes of memory and few logic gates. However, if only one function is calculated at a time, it has no effect on the overall pipeline performance. In EAM simulations, only two functions may be calculated simultaneously. Therefore, adding a third function table will not improve performance.

So we introduced cutoff units next. A cutoff unit, running at twice the frequency of the main pipeline, boosts the main pipeline performance by a factor of two. Figure 7 (d) illustrates the situation. We could have simply implemented four cutoff units to obtain eight times acceleration. However, adding cutoff units increases memory bus traffic to fill prefetch DPM, which may introduce another difficulty into the pipeline design. Therefore we combine these acceleration methods, two cutoff units and two function tables, to obtain a maximum performance gain of eight, which is greater than the design goal of 6.44 times performance improvements (Fig. 7 (e)).

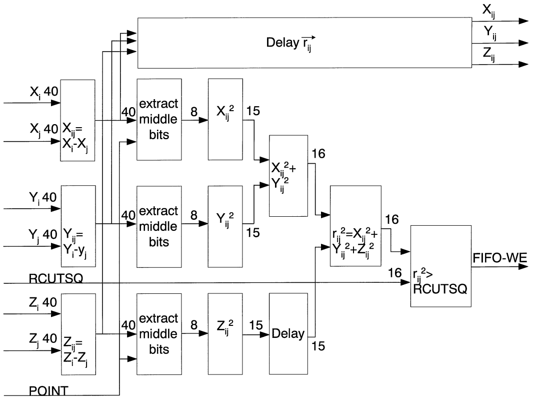
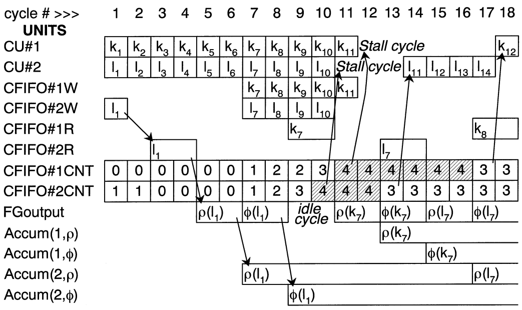


Fig. 8 Cutoff unit block diagram.

The DEM-1 pipeline operation is illustrated in the lower section of Fig. 6, where a pipeline with two cutoff units and one function table, denoted as "2C1T", is also illustrated for comparison. In 2C1T configuration, two cutoff units running at twice the frequency of the main pipeline, search for effective particle pairs with 4 times faster speed compared to the main pipeline. Since only 1/6.44 of the searched pairs actually goes into the main pipeline, main pipeline idle cycles are still observed. On the contrary, the DEM-1 pipeline, denoted as "2C2T," has no idle cycle at the main pipeline, because the two function tables decrease the cutoff FIFO read rate to 0.5 pair per main clock cycle. Cutoff units feed approximately 4/6.44 particle pairs into the cutoff FIFO at a rate faster than its read rate of 0.5 pair/clock.

The cutoff units use fixed-point arithmetic and shorter word length. Because of this imprecise arithmetic, several particles actually lying outside of the cutoff sphere may pass through the unit to generate cutoff error cycles, C_{ce}. However, this low accuracy improves the operating clock speed and hardware resource utilization. Each cutoff unit operates at twice the frequency of the main pipeline. Hardware resources required for the cutoff units are estimated to be less than 10% of the entire pipeline.

Figure 8 shows the block diagram of the cutoff unit. In the DEM-1 system, coordinates are converted to 40 bit signed fixed-point format and then stored in memory. Fixed-point 40 bit subtractions are performed to calculate the relative distance vector between particles i and j. A relative distance is also represented in 40 bit fixed-point format but its maximum absolute value is limited to twice the unit cell size,


cycle # >>> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
UNITS
CU#1 k1 k2 k3 k4 k5 k6 k7 k8 k9 k10 k11 Stall cycle k12
CU#2 l1 l2 l3 l4 l5 l6 l7 l8 l9 l10 Stall cycle l11 l12 l13 l14
CFIFO#1W k7 k8 k9 k10 k11
CFIFO#2W l1 l7 l8 l9 l10
CFIFO#1R k7 k8
CFIFO#2R l1 l7
CFIFO#1CNT 0 0 0 0 0 0 1 2 2 3 4 4 4 4 4 4 3 3
CFIFO#2CNT 1 1 0 0 0 0 1 2 3 4 4 4 3 3 3 3 3 3
FGoutput ρ(l1) φ(l1) idle cycle ρ(k7) φ(k7) ρ(l7) φ(l7)
Accum(1,ρ) ρ(k7)
Accum(1,φ) φ(k7)
Accum(2,ρ) ρ(l1) ρ(l7)
Accum(2,φ) φ(l1)

CU : Cutoff Unit
CFIFO : Cutoff FIFO
FGoutput : Function Generator output
Accum(n,m) : Accumlator (CU#, Function)

Latency of each unit is not correct.
The figure just illustrates the design concept of the DEM-1 pipeline.

Fig. 9 Pipeline timing.

because particles i and j are contained within neighboring cells. Therefore we can drop some upper bits without any problems. Also, we drop some lower bits to allow less accuracy in the rough cutoff.

In the current design, we extract an 8-bit signed integer to calculate an approximate squared distance for particle pair ij. The extract point depends on the simulation conditions (i.e., system size and unit cell size), and the extract point parameter (POINT) can be adjusted in software. The cutoff radius, RCUTSQ, used in the final comparator stage is also programmed by software. The programmer is expected to take account of the errors introduced by the imprecise arithmetic when he determines RCUTSQ.

Pipeline operation timing is shown in **Fig. 9**. Note that the two cutoff units run at double frequency. In the example shown in Fig. 9, cutoff unit #2 finds a neighboring particle pair (r_{l1}) and put it into cutoff FIFO #2. Then main pipeline reads relative distance vector r_{l1} from cutoff FIFO #2 to calculate ρ and ϕ sequentially. The pipeline controller selects accumulator $(2, \rho)$ and accumulator $(2, \phi)$, respectively, to accumulate these results. Those accumulations occur in an event driven manner. Cutoff units fail to find effective particle pairs for one main pipeline clock cycle so that a main pipeline idle cycle is inserted. The idle cycle occurs only at the start up of the pipeline, because the cutoff units search for particle pairs faster than the main pipeline consumes them in average. The cutoff units stall time to time due to almost full condition of cutoff FIFO. Cycles #7 through #11 illustrates such a situation, in which nine particle pairs are found to be inside the cutoff sphere and are put into cutoff FIFO. The "almost full" threshold is set to 4 to demonstrate FIFO full operation in Fig. 9. Cutoff units stall until the cutoff FIFO exits the almost full condition. Then they resume the search operation.

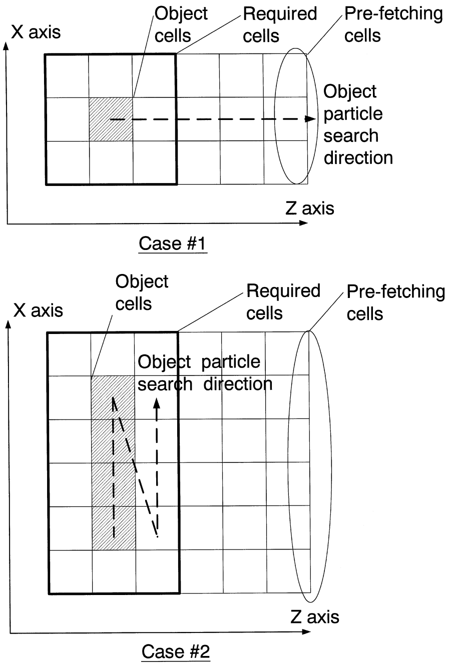


Fig. 10 Pre-fetch strategies (2-dimensional view).

3.3.2 Pre-fetch DPM

Unlike other particle simulation machines, DEM-1 has dedicated data flow paths to each pipeline. This feature eliminates pipeline overhead introduced by pipeline data sharing and increases its effective performance.

A DEM-1 chip has eight 16 Kbyte position vector pre-fetch dual ported memories. Usually, data have been written to these eight DPMs by broadcast. However, their data reads can occur independently. Position vectors and coefficients are used many times repeatedly during the pipeline calculation. Thus they can be pre-fetched via a relatively slower external bus without reducing pipeline performance. The external bus bandwidth required to keep internal pipelines busy depends on the size of the dual ported memory and the pre-fetch strategy. **Figure 10** show some typical cases of pre-fetch strategies. In the examples shown here, the memory controller of the DEM-1 Chip pre-

Table 2 DPM transfer rate requirement.

	Symbols	Equations	Case#1	Case#2	Case#3
Number of interactions	N_i		2	2	2
Number of particles in a cell	N_{pc}		4	4	8
Neighboring cells	N_n	3^3	27	27	27
Number of pipelines	N_{pipe}		4	4	4
Number of neighboring particles	N_{np}	$N_{pc} * N_n$	108	108	216
Number of interacting particles	N_{ip}	N_{np}/R_{eff}	16.8	16.8	33.5
Bytes per particle	N_{bpp}		20	20	20
Size of each side of cells	N_{lc}	*(strategy)*	1	4	2
Object particles *	N_{pz}	$N_{lc}^2 * N_{pc}$	4	64	32
Clocks per particle	C_{clk}	$N_{ip} * N_i/N_{pipe}$	8.4	8.4	16.8
Clocks to calculate object cell(s) *	C_{zclk}	$N_{pz} * C_{clk}$	33.5	536.6	536.6
Cells reqired *	N_{zcell}	$(N_{lc} + 2)^2$	9	36	16
Particles reqired *	N_{scz}	$N_{zcell} * N_{pc}$	36	144	128
Bytes required *	N_{br}	$N_{scz} * N_{bpp}$	720	2880	2560
Bytes per clock to fill DPM	B_{bw}	N_{br}/C_{zclk}	21.5	5.4	4.8
Size of a DPM	N_{bytes}	16384	16384	16384	16384
Number of xy planes in DPM	N_{zpl}	N_{bytes}/N_{br}	22.8	5.7	6.4

* Listed per xy plane

fetches particles along the z axis.

Table 2 estimates the transfer performance of external memory required with some simulation parameters. The particle count N_{np} in the (average) neighboring cells of a single particle is expressed as shown below.

$$N_{np} = N_{pc} * N_n \qquad (7)$$

where, N_{pc} is the average particle count in a cell and N_n is the number of neighboring cells. The following discussion assumes $N_n = 27$. Among these particles, those that actually cause the interaction are expressed as shown below.

$$N_{ip} = N_{np}/R_{eff} \qquad (8)$$

where, R_{eff} is a cutoff ratio of approximately 6.44 (see Appendix A.1). The size of a cell targeted for computation depends upon the read strategy. If the size of each side of cells to be read on the xy plane is N_{lc}, the cell count N_{zcell} required for computation would be a set of its neighboring cells expressed as shown below.

$$N_{zcell} = (N_{lc} + 2)^2 \qquad (9)$$

On the other hand, the clock cycle count C_{clk} spent for each particle targeted for computation is expressed by the following equation.

$$C_{clk} = N_{ip} * N_i/N_{pipe} \qquad (10)$$

where, N_i is the number of interactions calculated at the same time and N_{pipe} is the number of main pipelines. Since DEM-1 can calculate two interactions at the same time, N_i is 2 and the pipeline count N_{pipe} is 4. The clock cycle count C_{zclk} for calculating the interaction of all particles of a cell subject to the xy plane calculation is expressed as shown below.

$$C_{zclk} = C_{clk} * N_{lc}^2 * N_{pc} \qquad (11)$$

Loading a new cell from coordinate memory to pre-fetch DPM must be balanced with the reading speed. The particle count N_{szc} contained in the neighboring cells on the xy plane, which will cause an interaction, is expressed by the following equation.

$$N_{szc} = N_{zcell} * N_{pc} \qquad (12)$$

Its total byte count N_{br} can be represented as shown below when the information byte count per particle is N_{bpp}.

$$N_{br} = N_{szc} * N_{bpp} \qquad (13)$$

On the machine DEM-1, N_{bpp} is 20 bytes including the 120-bit particle coordinates, 32-bit coefficient, and 8-bit tag. Consequently, the required external bus bandwidth B_{bw} is expressed by the following equation.

$$B_{bw} = N_{br}/C_{zclk} \qquad (14)$$

In addition, the XY plane count N_{zpl} in DPM can be shown as follows when the DPM size is N_{bytes}.

$$N_{zpl} = N_{bytes}/N_{br} \qquad (15)$$

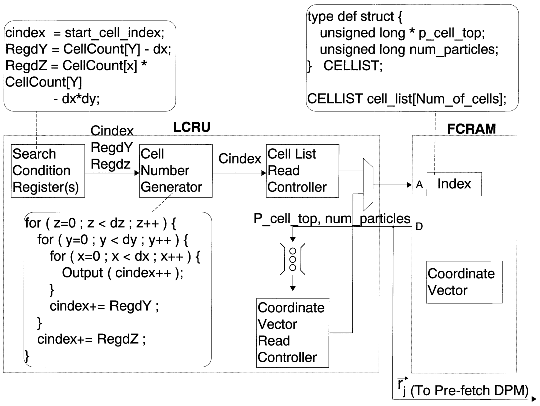
In case #1, N_{lc} equals to 1. Cells of a rectangular solid, of which size is $(1, 1, N_{lz})$, are selected as target cells. N_{lz} is the number of cells along the z-axis, which are assigned to a DEM-1 chip. This case requires a transfer rate of 21.5 bytes/clock estimated from Equation (14), which is higher than the 16 bytes/clock peak transfer rate of the DEM-1 Chip external memory bus. On the other hand, a transfer rate of only 5.4 bytes/clock is expected in case #2, in which $(4, 4, N_{lz})$ rectangular solid cells are calculated at a time and surrounding $(6, 6, N_{lz}+2)$ cells are read into pre-fetch DPM. The number of xy planes contained in 16 Kbytes is 5.7 in case #2. Note that the minimum number of xy planes is 4 considering the overlapped pre-fetch during pipeline calculations. The average number of particles, which are contained in a cell, may increase depending on the interactions we adopt. We can adjust "object cell size" (N_{lc}) parameter to get optimum transfer rate (B_{bw}) and number of xy planes in DPM (N_{zpl}). Case #3 shown in Table 2, in which the number of particles within a cell (N_{pc}) equals to 8, describes such a situation.

3.3.3 LCRU

The DEM-1 pipeline has Linked Cell list Read Unit (LRCU) to support linked cell list addressing. The design of the LCRU is based on that of MD One/E[7]. The LCRU designate cells subject to calculation in a simulation space by using the three-dimensional offset values (X_s, Y_s, Z_s) and size (dX, dY, dZ). DEM-1 has three types of LCRU, which are named P-LCRU, R-LCRU and C-LCRU.

DEM-1 has a pre-fetch LCRU (P-LCRU) common to all 8 pre-fetch DPMs. The block diagram of the P-LCRU is shown in **Fig. 11** with pseudo-codes to explain operations of each unit and data structure. The memory controller of the P-LCRU reads the cell top pointer (`p_cell_top`) and the number of particles contained in the cell (`num_particles`) from the cell list. They are used to read actual particle coordinates from FCRAM. The three cells, which are read along x direction, have successive cell numbers so that reading of these three cells can be executed in burst access. Remembering that each cell has four particles on average, we expect 12 particles, 288 (= 12*3*8) bytes, to be read at a time. It takes 18 (= 12*3/2) clocks to read 12 particles in burst mode with the DDR interface of the DEM-1 memory controller. The same discussions can be applied to reading the linked cell list, which takes three clock cycles to read six consecutive words. If we assume random access latency of FCRAM to be three clocks, the effective transfer rate of the FCRAM can be estimated as 16 bytes/clock*18/(3+18+3+3) = 10.7 bytes/clock including the linked cell list search. The FIFO between the cell list read controller and coordinate vector read controller accommodates the rate fluctuations of coordinate memory access.

The register write LCRU (R-LCRU) is common to 8 cutoff units, and is linked with the register write function to automatically pick up a particle, which undergoes the interaction, from the linked cell list and write its data to the pipeline particle shadow registers. The R-LCRU searches for target particles and sequentially assigns them to the registers. The shadow registers are loaded into the particle registers before the cutoff unit operations.



```
cindex  = start_cell_index;
RegdY = CellCount[Y] - dx;
RegdZ = CellCount[x] *
CellCount[Y]
        - dx*dy;
```

```
type def struct {
  unsigned long * p_cell_top;
  unsigned long num_particles;
}  CELLIST;

CELLIST cell_list[Num_of_cells];
```

```
for ( z=0 ; z < dz ; z++ ) {
  for ( y=0 ; y < dy ; y++ ) {
    for ( x=0 ; x < dx ; x++ ) {
      Output ( cindex++ );
    }
    cindex+= RegdY ;
  }
  cindex+= RegdZ ;
}
```

LCRU
FCRAM
Search Condition Register(s)
Cindex
RegdY
Regdz
Cell Number Generator
Cindex
Cell List Read Controller
A
Index
P_cell_top, num_particles
D
Coordinate Vector Read Controller
Coordinate Vector
$\vec{r}_j$ (To Pre-fetch DPM)

Fig. 11 Linked Cell list Read Unit (LCRU).

Upon completion of data loading, the eight cutoff unit LCRUs (C-LCRU) give particles contained in cells having the potential of causing an interaction to each cutoff unit pipeline.

In the general linked cell list method, cells subject to computation are cubic with $dX = dY = dZ = 3$ when $N_{rc} = 1$ is provided in Equation (4). Generally, the LCRU requires only updating one register (e.g., Xs) to move a cell to be searched, reducing the software overhead and increasing the parallel operation efficiency. The respective LCRUs have a function for automatically processes an imaginary cell that appears in periodic boundary conditions, so that performance will not be degraded under those conditions.

3.3.4 Result FIFO

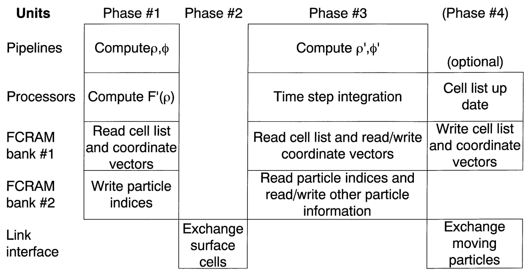
Computed interactions are posted from the accumulators to the Result FIFOs (RFIFO) upon completion of computation. The processors attached to the DEM-1 pipeline can read calculated interactions at any time. The RFIFO works to accommodate the rate fluctuations of the pipeline.

3.4 EAM simulations on DEM-1

Figure 12 shows the time required for a single step of the EAM simulation. This illustration simulates the calculation and communication overhead of each phase. In the first phase, potentials ρ and ϕ are first calculated during the interaction computation with the pipeline. During this phase, potential accumulators accumulate ϕ while processors compute $F'(\rho_i)$ form the obtained ρ_i. Particle indices

Table 3 DEM-1 pipeline prototype implementation parameters.

	FPGA Prototype	DEM-1 Pipeline	DEM-1 Chip
Main pipeline frequency	33 MHz	133 MHz	133 MHz
Cutoff unit frequency	66 MHz	266 MHz	266 MHz
Processor frequency	N/A	N/A	266 MHz
Number of cutoff units	2	2	8
Number of main pipelines	1	1	4
Number of processors	0	0	4
Pipeline performance	830 Mflops	3.33 Gflops	13.3 Gflops
Relative performance	1	4	16



Units	Phase #1	Phase #2	Phase #3	(Phase #4)
Pipelines	Computeρ,φ		Compute ρ',φ'	(optional)
Processors	Compute F'(ρ)		Time step integration	Cell list up date
FCRAM bank #1	Read cell list and coordinate vectors		Read cell list and read/write coordinate vectors	Write cell list and coordinate vectors
FCRAM bank #2	Write particle indices		Read particle indices and read/write other particle information	
Link interface		Exchange surface cells		Exchange moving particles

Fig. 12 Calculation and communication overhead of each phase and each unit.

obtained in the linked cell list method particle search are stored temporarily in FCRAM. Subsequently, the coordinates of neighboring particles required for the computation and the resultant $F'(\rho_i)$ are exchanged between adjacent nodes. At this phase, the DEM-1 chips, boards or nodes are exchanging "surface" cells. Then, ρ' and ϕ' are calculated with the pipeline and the result is used as the base for updating the particle information (coordinates, velocities, and such). At this phase, particle indices obtained in the previous phase are used to prefetch particle information, because calculation order and particle assignment to the pipelines are the same as the first phase. The particle information resulting from the computation is written back to memory. Then, once in several times steps, linked cell list is updated. If a particle coordinate update result is outside the space area assigned to each DEM-1 Chip, the concerned coordinates and other particle information are exchanged between adjacent DEM-1 Chips, DEM-1 boards, or DEM-1 nodes.

4. Implementation of DEM-1 Pipeline

We have implemented a prototype on an FPGA test-bed called REX[13] to evaluate the performance of the DEM-1 pipeline. A complete DEM-1 pipeline (see Fig. 6) was implemented on one of two Xilinx FPGAs, XCV2000EFG1156-6, of REX. Gate and memory usage were 92% and 88%, respectively. The PCI bus interface was used to control and monitor the pipeline. The main pipeline runs with a 33 MHz system clock to deliver 830 Mflops per second in performance. **Table 3** summarizes the specifications of the FPGA prototype along with the DEM-1 chip specifications. The FPGA implementation makes it possible to simulate different pipeline configurations with few changes which is very useful at this design stage.

Even though pipeline register access speed is very slow due to the overhead of the current device driver, the test-bed can simulate pipeline calculations for 1 million atoms within an hour, which is 1,000 times faster than function simulator we use, Modelsim PE.

Several diagnostic registers have been implemented to monitor pipeline performance: the AF Count register (AFC) and the RD Count register (RDC). AFC counts cutoff unit stall cycles caused by almost full conditions of the cutoff FIFO while RDC counts particle pairs that fall inside the cutoff sphere and go into the main pipeline. RDC also counts some overhead cycles at pipeline start/stop. These diagnostic registers can be used to monitor the DEM-1 pipeline performance during execution of simulations. They provide useful information to determine optimized pipeline settings ideal for each type of target simulation.

"Foundation ISE 4.2" integrated development system from Xilinx was used to develop the FPGA prototype along with "FPGA Express" synthesis tool from Synopsis. The pipeline prototype was written in VHDL.

5. Performance Evaluation

5.1 Evaluation Methods

DEM-1 performance was analyzed for a 512-million atom EAM simulation. Each DEM-1 chip computes interactions over 1-million atoms with its four pipelines and four processors. Pipeline performances were measured on a prototype test-bed while processor perfor-

Table 4 DEM-1 pipeline performances: 1-million atoms, 2 interactions, double clock cycles.

	0C1T	1C1T	1C2T	2C1T	2C2T
C_{efc}	6.05E+07	6.05E+07	6.05E+07	6.05E+07	6.05E+07
C_{rdc}	4.32E+08	6.93E+07	6.93E+07	6.97E+07	6.97E+07
C_{stall}	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.75E+07
C_{run}	4.32E+08	2.16E+08	1.08E+08	1.12E+08	6.97E+07
P_{eff}	0.1400	0.2800	0.5601	0.5411	0.8682
P_{busy}	1.0000	0.3207	0.6413	0.6230	0.9997
P_{gain}	1.0000	2.0000	4.0000	3.8642	6.2008

mances and communication performance were estimated from design parameters.

5.2 Pipeline Performance Measurement

First, performance of a reference pipeline with no cutoff unit and a single function table is estimated. The number of effective pipeline cycles is represented by;

$$C_{efc} = \sum_{i}^{all\_particles} N_{nb}(i) \qquad (16)$$

where C_{efc} is the total number of effective cycles and $N_{nb}(i)$ is the number of particles that are contained in cutoff sphere around i-th particle, which equals effective cycles calculating the interaction between the neighboring particles and the i-th particle. The number of clock cycles, C_{src}, to compute the interaction among all particles with the reference pipeline is equal to the number of clock cycles to search through the linked cell list and is represented by;

$$C_{src} = \sum_{i}^{all\_cells} (N(i) * \sum_{j}^{neighboring\_cells} N(j)) \qquad (17)$$

where $N(i)$ or $N(j)$ is the number of particles contained in a cell. The number of clock cycle to compute all interaction among all particles with different pipeline configurations is represented by;

$$C_{run} = C_{src}/K_f + C_{stall} \qquad (18)$$

where K_f is the acceleration factor of each pipeline and C_{stall} is the cutoff unit stall cycle time due to the almost full condition of the cutoff FIFO. The ratio of C_{efc} to C_{run} can be considered the main pipeline efficiency (P_{eff}) and is expected to be approximately 1/6.44 for the reference pipeline, as discussed in section 2.

$$P_{eff} = C_{efc}/C_{run} \qquad (19)$$

Since the DEM-1 pipeline has the ability to calculate two interactions of a particle pair simultaneously, we now discuss pipeline performance to obtain two interactions here. It takes $2{*}C_{run}$ main pipeline clock cycles (e.g., $4{*}C_{run}$ double clock cycle) to calculate two interactions with the reference pipeline, because the reference pipeline can calculate a single function per clock cycle time.

The DEM-1 pipeline performance to execute 1-million atom bcc crystal interaction calculations has been measured with different pipeline configurations. The lattice parameter of the crystal was assumed to be 2.87 Angstrom, while cutoff radius and unit cell size were assumed to be 3.46 Angstrom and 3.58 Angstrom, respectively.

Table 4 summarizes measured performances. A configuration symbol of "nCmT" corresponds to a pipeline with n cutoff units and m function tables. The reference pipeline is represented by "0C1T". Continuous operation of the pipeline was emulated by software since the memory controller has not been implemented in the current prototype design.

In Table 4, the main pipeline busy factor P_{busy} is calculated as;

$$P_{busy} = C_{rdc}/C_{run} \qquad (20)$$

where C_{rdc} is the number of particle pairs posted into the main pipeline. C_{rdc} is obtained by reading the diagnostic register, RDC. Performance gain, P_{gain}, is defined as run time ratio over the reference pipeline;

$$P_{gain} = C_{run}(0C1T)/C_{run} \qquad (21)$$

The maximum performance gain of 6.2008 (2C2T) over the reference pipeline architecture is close to the expected value 6.44.

The following describes the pipeline operation in each configuration. First, since the pipeline does not have a cutoff unit in the reference pipeline configuration (0C1T), all particles searched in the linked cell list method are entered to the main pipeline. Therefore, C_{run} becomes equivalent to C_{rdc} whereas C_{efc} becomes approximately 1/6.44 of C_{rdc}. $C_{rdc} - C_{efc}$ is deemed a cutoff error, C_{ce}.

$$C_{ce} = C_{rdc} - C_{efc} \qquad (22)$$

When this configuration is provided with a single unit of cutoff operating with a double clock, the 1C1T configuration is formed. Since the linked cell list search is carried out at double speed in the 1C1T configuration, K_f is 2 and the C_{run} value becomes half that in the 0C1T configuration. Due to operation of the cutoff unit, the number of particles (C_{rdc}) entered to the main pipeline becomes closer to C_{efc}. However, a cutoff error causes $C_{rdc} > C_{efc}$.

When the 1C1T configuration is provided with two function tables to alternately calculate two types of interactions, the 1C2T configuration is established. Two types of interactions can be calculated during a single search in the 1C2T configuration. Thus, K_f is 4 and C_{run} becomes half the rate in the 1C1T configuration. The C_{rdc} value does not change from the rate in the 1C1T configuration. However, since the C_{run} value becomes half, the overall pipeline operation rate (P_{eff}) is double the rate in the 1C1T configuration.

In the 2C1T configuration, the effect of double P_{eff} can also be expected due to C_{run} being half the rate in the 1C1T configuration. Nevertheless, the two cutoff units actually mounted at present are not completely independent, but do operate in synchronization, showing slightly degraded performance caused by the connection wait of two cutoff units. Consequently, the C_{run} value in the 2C1T configuration is larger by several percentage than 1/2 in the 1C1T configuration. These connection wait cycles seems to be counted by RDC as C_{rdc}, which is slightly higher than that in 1CXT configuration. This behavior is not intended and it should be investigated later.

The 2C2T configuration indicates phenomena different from the above configurations. The effect of up to 8-times acceleration is obtained from two cutoff units and two function tables in the 2C2T configuration. However, since approximately 1/6 of all particles enter the main pipeline as a result of cutoff (including the cutoff error), the main pipeline is subject to rate fluctuation. As a result, the cutoff unit stall-cycle C_{stall} is generated by the Almost Full event of the cutoff FIFO in this configuration, so that the P_{gain} value is stable at 6.2 (not 8) close to the cutoff ratio.

Figure 13 visualizes the results discussed above. Together with the results listed in Table 4 (denoted as #1), performance was measured with a 1-million bcc crystal distorted by random perturbations (denoted as #2). Although a slightly higher amount of cutoff errors is observed in the calculations of a crystal with perturbations (#2), the errors have small effects on overall pipeline performance.

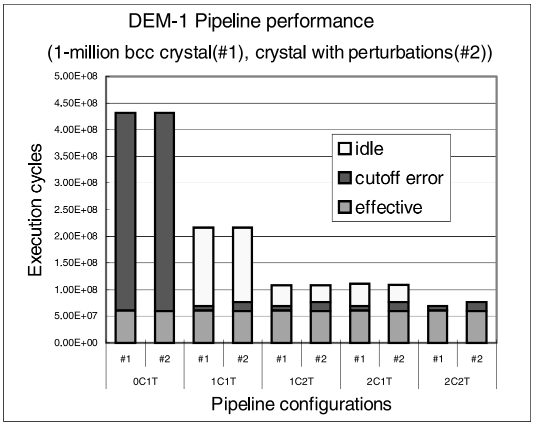


Fig. 13 Pipeline performance.

Cutoff error due to limited arithmetic accuracy of cutoff units results in somewhat higher P_{busy} compared to P_{eff}. However, performance losses are not observed in the following three configurations, 1C1T, 1C2T, 2C1T, where no cutoff unit stall cycle occurs.

Cutoff unit stall cycles are observed in calculations with 2C2T pipeline configurations. In this configuration, the main pipeline limits the overall pipeline performance, while cutoff units are occasionally stall. Therefore, cutoff units can search among more particle pairs within the same pipeline run time. This is realized by setting a larger unit cell size. Increasing unit cell size results in reduction of computational cost and memory resource requirements to build the linked cell list. Optimum cell size can be determined on the fly by monitoring the RDC and AFC diagnostic registers.

Figure 14 shows how cutoff arithmetic accuracy affects the overall pipeline performance for 2C2T configuration. The low accuracy case was measured by setting the extraction point at higher positions. As cutoff accuracy increases, the number of pipeline execution cycles decrease toward the ideal performance without cutoff errors. Observed performance losses with 8-bit accuracy in cutoff units were 10-20% depending on simulation parameters such as the number of particles or amount of perturbation. Note that even 4-bit cutoff accuracy is acceptable with a pipeline configuration of one cutoff unit and one table, 1C1T, where idle cycles dominate the total execution cycle time.

DEM-1 chip performance was estimated from

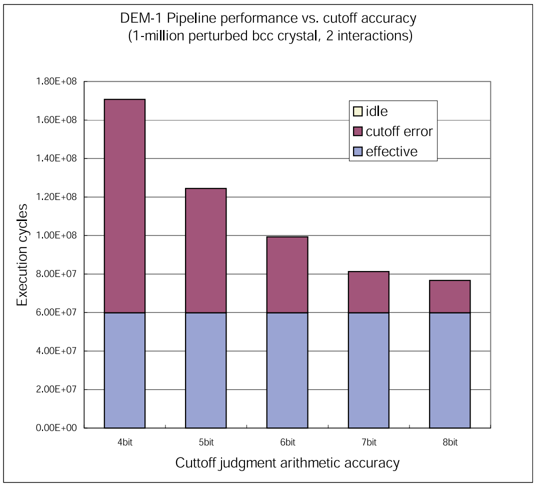


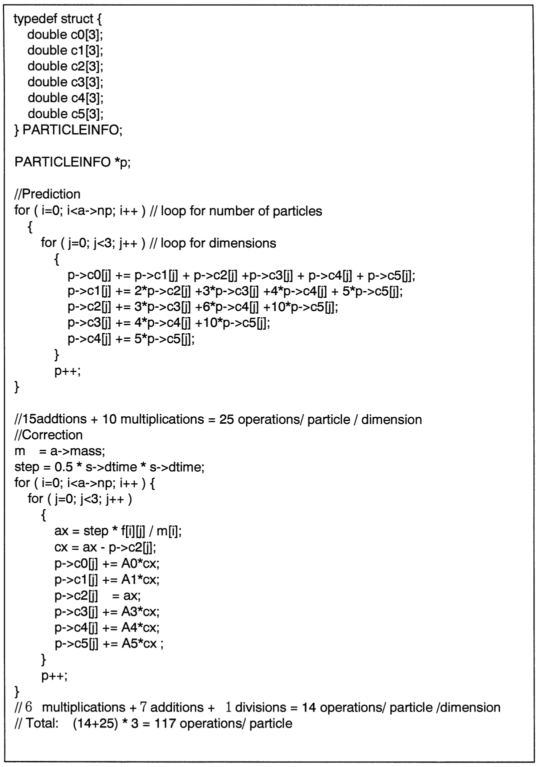
Fig. 14 2C2T pipeline performance with different cutoff accuracies.

the experimental results. The DEM-1 chip has four pipelines running at 133 MHz main pipeline clock frequency. Therefore, execution time to calculate four interactions of EAM over 1 million atoms was estimated as the following.

By Applying $f_{dclk} = 266\,\mathrm{MHz}$ and $C_{run} = 6.97\mathrm{e}{+}07$, we obtained $T_{calc} = 131\,\mathrm{msec}$.

5.3 Processor Performance Estimation

On-chip processors tightly integrated with the DEM-1 pipelines execute tasks other than interaction calculations. Computational load depends on simulation parameters and methods, such as temperature control or pressure control. The most time consuming part of the processor tasks is time step integration, in which position vectors, velocity vectors, acceleration, etc. are updated. We analyzed EAM molecular dynamics simulation software code, XMD [14], and estimated the calculation cost of it to be 117 steps per particle (**Fig. 15**). As seen in Fig. 15, the prediction-correction method reads/writes six vectors, which counts for 6*3*8=144 bytes per particle. Remembering that the memory controller of DEM-1 transfers 8 bytes per 266 MHz clock, it takes 6*3*2=36 clocks to read/write the six vectors. Since the DEM-1 chip has four processors, it takes 36*4=144 clocks, which is larger than the estimated 117 clocks of processor computation cycles, to read/write six vectors into each processor's cache memory. In this case, the external memory bus limits the processor performance.



```
typedef struct {
   double c0[3];
   double c1[3];
   double c2[3];
   double c3[3];
   double c4[3];
   double c5[3];
} PARTICLEINFO;

PARTICLEINFO *p;

//Prediction
for ( i=0; i<a->np; i++ ) // loop for number of particles
   {
      for ( j=0; j<3; j++ ) // loop for dimensions
         {
            p->c0[j] += p->c1[j] + p->c2[j] +p->c3[j] + p->c4[j] + p->c5[j];
            p->c1[j] += 2*p->c2[j] +3*p->c3[j] +4*p->c4[j] + 5*p->c5[j];
            p->c2[j] += 3*p->c3[j] +6*p->c4[j] +10*p->c5[j];
            p->c3[j] += 4*p->c4[j] +10*p->c5[j];
            p->c4[j] += 5*p->c5[j];
         }
         p++;
}

//15addtions + 10 multiplications = 25 operations/ particle / dimension
//Correction
m    = a->mass;
step = 0.5 * s->dtime * s->dtime;
for ( i=0; i<a->np; i++ ) {
   for ( j=0; j<3; j++ )
      {
         ax = step * f[i][j] / m[i];
         cx = ax - p->c2[j];
         p->c0[j] += A0*cx;
         p->c1[j] += A1*cx;
         p->c2[j]    = ax;
         p->c3[j] += A3*cx;
         p->c4[j] += A4*cx;
         p->c5[j] += A5*cx ;
      }
      p++;
}
// 6   multiplications + 7 additions +   1 divisions = 14 operations/ particle /dimension
// Total:    (14+25) * 3 = 117 operations/ particle
```

Fig. 15 A typical program to perform time step integration.

In order to avoid false sharing between processors, particle information vectors are packed into a structure. Particle information structure boundary is aligned to the 128 bit boundary of the processor cache line.

Another time consuming part of the processor task is re-building of linked cell list. At this phase, the tasks are assigned to the processors by spatial decomposition scheme to avoid false sharing. This process may be executed only once in several time steps. As described in the previous subsection (5.2), it is possible to use larger unit cell size without blocking the pipeline performance, thus reducing the rate of the cell list re-build process in EAM simulation with DEM-1.

Additional profiling analysis of the software shows that the time step integration occupies approximately 1/3 of the execution time of tasks assigned to local processors. From these observations, we estimated the total computational cost to be 450 clocks per particle. In a 512-million atom EAM simulation, each DEM-1 chip computes movements of 1 million atoms per time step. Since the DEM-1 chip has four on-chip processors, each processor executes 250,000*450 steps to perform this task. With

Table 5 Estimated communication time required to exchange surface particles.

	System	Node	Board	Chip
Particles	512 M	64 M	8 M	1 M
Cells	512^3	256^3	128^3	64^3
Surface cells	1.5 M	384 K	96 K	24 K
Surface particles	6 M	1.5 M	384 K	96 K
Transfer bytes	N.A.	120 M	30.7 M	7.7 M
Link speed (bytes/sec)	N.A.	3 G	800 M	400 M
Required time (msec)	N.A.	40	38.4	19.2

the processor core running at 266 MHz, it takes approximately 420 msec to perform tasks assigned to the processors.

In order to achieve the performance discussed above, the minimum memory stall cycle is allowed. This is made possible by tight integration of pipeline and processor. The memory controller on the DEM-1 chip schedules all other data transfers to keep the processors running. This is possible because much regularity is observed with memory access patterns in particle simulations.

5.4 Communication Performance Estimation

Times required to exchange particle data are estimated in **Table 5**. The DEM-1 chips, boards, or nodes exchange 160 bits (120-bit coordinates, 32-bit $F'(\rho)$ and 8-bit tag) for each particle inside a surface cell in phase #2 of Fig. 12. During phase #4, only a few among the surface particles travels across the cell boundary. The total amount of data to be transferred for moving particles from one surface cell to a neighboring unit depends on the simulation methods and parameters. It is 144 bytes per particle for the example shown in the previous section. It seems quite reasonable to assume that the total amount of data transfered does not exceed 40 bytes per particle during each time step of EAM simulation. As seen in Table 5, a DEM-1 system can exchange particles within 40 msec per time step when it executes a 512-million particle bulk iron simulation.

5.5 Overall Performance Estimation

By combining the execution times we have calculated in the previous sections, we obtain 590 msec for computation time per step. All other overheads can be much less than the discussed execution times. Therefore, a DEM-1 system is expected to execute each time step of an EAM simulation of a 512-million atom bulk iron within a second.

6. Related Work

MD GRAPE[2)] was developed at the University of Tokyo. MD One[3)] is a commercial version of MD GRAPE developed by Image Technology Laboratory Corporation. Both of them have a primitive linked-cell-list addressing support hardware, and a neighbor list FIFO that stores indices of particles falling inside the cutoff sphere. A single host processor controls multiple boards with multiple pipelines, which share common data paths within the board.

MD Engine[4)] and MD Engine II[5)], developed at Fuji Xerox Co., Ltd., have similar single host processor system architecture. They have a special purpose circuit for utilizing the neighbor list. It reduces the calculation cost by 50% under certain conditions.

MDM[6)] is a very large-scale, and high speed version of MD GRAPE with dedicated wave number space integration accelerator WINE-2. MDM has four nodes of 6-CPU parallel host computers running MPI. However, its basic architecture is not much different from that of MD GRAPE. Therefore, most of the deficiencies discussed in earlier sections of this paper exist and performance is reduced when it executes particle simulations with short-range interactions.

MD One/E[7)] has sophisticated linked cell list addressing support hardware to accelerate short-range interaction calculations. However, it has certain problems described in section 2.2.

Available particle simulation machines have no pre-fetch DPM to feed data to pipelines. Instead, MD GRAPE[2)], MD One[3)], MD One/E[7)] and MDM[6)] have virtual multiple pipelines at chip level to reduce external bus bandwidth without sacrificing pipeline performance in simulations without cutoff. However, if we execute simulations with short-range interactions, their performance is limited by pipeline data sharing, as discussed in section 2. MD Engine II[5)] have multiple pipelines to calculate 2 interactions on 2 particles at a time. MD Engine[4)] does not have multiple pipelines. However, both of them have shared particle memory architecture too, in which pipeline

Table 6 Particle simulation machines.

	MD GRAPE	MD Engine	MDM	**DEM-1**
Developed at	The University of Tokyo	Fuji Xerox Co., Ltd.	Riken	The University of Electro-Communications
Target application	Long-range interactions	Long-range interactions	Long-range interactions	Short-range interactions
Pipeline data path	Shared	Shared	Shared	Independent
Cutoff unit directly controls pipeline	No	No	No	Yes
Pipelines/LSI	1	1	4	4
Pipeline operations/clock	30	4	30	50
Pipeline speed	35 MHz	20 MHz	100 MHz	133 MHz
Processors/LSI	0	0	0	4
Processors/system	1	1	24	2048
Tight integration of pipeline and processor	No	No	No	Yes

data sharing overhead still exists.

Available particle simulation machines have no cutoff units that directly control the operation of the main pipeline. The prototype DEM-1 pipeline running at 33 MHz main pipeline clock frequency is 16 times faster than the pipeline of MD One/E, considering 8/3 times performance improvement by the elimination of data sharing overhead by pre-fetch DPM and six times measured performance gain by two cutoff units and two function tables. Since the DEM-1 chip has four pipelines running at 133 MHz, the DEM-1 chip computes short-range interactions 256 times faster than the MD One/E pipeline.

Available particle simulation machines have no local processors tightly integrated to the pipelines. Therefore, communication overhead and time step integration overhead limit their ability to perform particle simulations with short-range interactions.

The features of particle simulation machines are summarized and compared in **Table 6** along with the features of DEM-1.

7. Conclusion

This paper introduces the design of the particle simulation machine DEM-1 and its preliminary implementation. DEM-1 has cutoff units that directly control the main pipeline to boost its performance by a factor of six. Together with the dedicated pipeline data path supported by pre-fetch dual ported memory, the DEM-1 pipeline can execute short-range interaction computations more than 10 times faster than existing particle simulation machine pipelines even when compared with the same clock frequency. Tight integration of pipeline and processor eliminates communication bottlenecks between pipelines and processors.

The DEM-1 pipeline prototype was implemented on an FPGA test-bed REX. Performance of the pipeline has been measured with different pipeline configurations and simulation parameters. Up to six times pipeline performance improvement was measured for configurations with two cutoff units and two function tables. On-chip processor performance and communication performance have also been estimated for a 512-million atom bulk iron simulation. DEM-1 is expected to execute a single time step of such an application within 1 second.

Acknowledgments The Authors would like to thank reviewers for their valuable comments. They would also like to thank Dr. Clement Leung for his comments. Hajime Kondo designed part of the DEM-1 pipeline.

References

1) Makino, J., Fukushige, T. and Koga, M.: A 1.349 Tflops simulation of black holes in a galactic center on GRAPE-6, *Proc.IEEE/ACM SC2000 Conference* (2000).

2) Fukushige, T., Taiji, M., Makino, J., Ebisuzaki, T. and Sugimoto, D.: A highly-parallelized special-purpose computer for many-body simulations with an arbitrary central force: MD-GRAPE, *Astrophysical Journal*, Vol.468, pp.51–61 (1996).

3) Komeiji, Y., Uebayasi, M., Takata, R., Shimizu, A., Itsukashi, K. and Taiji, M.: Fast and accurate molecular dynamics simulation of a protein using a special-purpose computer, *Journal of Computational Chemistry*, Vol.18, No.12, pp.1546–1563 (1997).

4) Toyoda, S., Miyagawa, H., Kitamura, K.,

Amisaki, T., Hashimoto, E., Ikeda, H., Kusumi, A. and Miyakawa, N.: Development of MD Engine: high-speed accelerator with parallel processor design for molecular dynamics simulations, *Journal of Computational Chemistry*, Vol.20, No.2, pp.182–199 (1999).

5) Amisaki, T., Toyoda, S., Miyagawa, H. and Kitamura, K.: Two algorithms designed for realizing efficient combinations of fast multipole method and dedicated hardware for molecular dynamics simulations (in Japanese), *The Journal of Computer Chemistry, Japan*, Vol.1, No.3, pp.73–82 (2002).

6) Narumi, T., Susukita, T., Koishi, T., Yasuoka K., Furusawa, H., Kawai, A. and Ebisuzaki, T.: 1.34 Tflops molecular dynamics simulation for NaCl with a special-purpose computer, MDM, *Proc. IEEE/ACM SC2000 Conference* (2000).

7) Takata, R. and Soneda, N.: An addressing unit used in a many body simulation machine (in Japanese), JP2000-045606A (2000), Patent pending.

8) Takata, R., Kise, K., Honda, H. and Yuba, T.: A particle simulation machine designed for simulations with short-range interactions: DEM-1 (in Japanese), *Proc. JSPP2001* pp.287–294 (2001).

9) Takata, R., Kise, K., Honda, H. and Yuba, T.: DEM-1: a particle simulation machine for efficient short-range interaction computations, *Proc. IPDPS2002* (2002).

10) Daw, M.S. and Baskes, M.I.: Embedded atom method: Derivation and application to impurities, surfaces, and other defects in metals, *Physical Review B*, Vol.29, No.12, pp.6443–6453 (1984).

11) Takata, R., Honda, H. and Yuba, T.: A particle simulation machine based on hierarchical parallel structure and operation chaining (in Japanese), *IPSJ SIG Notes*, 99-ARC-134, pp.127–132 (1999).

12) Nishimura, S., Harasawa, K., Matsudaira, N., Akutsu, S., Kudoh, T., Nishi, H. and Amano, H.: RHiNET-2/SW: a large-throughput, compact network-switch using 8.8-Gbit/s optical interconnection, *New Generation Computing*, Vol.18, No.2, pp.188–197 (2000).

13) Sayano, K., Katashita, T., Koike, H., Kodama, Y., Sakane, H. and Koumura, Y.: A multiprocessor emulation system by the application of large-scale FPGA (in Japanese), *Proc. JSPP2001*, pp.79–80 (2001).

14) `http://www.ims.uconn.edu/centers/simul/#Software`

Appendix

A.1 Pipeline Performance of Existing Particle Simulation Machines

Linked cell list method selects particles in 27 neighboring cells (Fig. 1). Particles inside the cutoff sphere affect the ith particle that we are focusing on. Usually the cell size is equal to cutoff radius, r_{cut}. One can assume that the number of particles in each area is proportional to its volume, so that the ratio of particles within the 27 adjacent cells and within the cutoff sphere can be represented as:

$$R_{eff} = 27 * r_{cut}^3/(4\pi r_{cut}^3) \cong 6.44 \qquad (23)$$

In addition, the hardwired pipeline cannot take advantage of vectorized linked cell list method, in which $\vec{f_{ji}}$ is calculated as $-\vec{f_{ij}}$ (Newton's third law) to reduce the calculation cost by 1/2. Together, the effective performance of the pipeline of existing particle simulation machines is estimated to be 1/13 of their peak performance in simulations with cutoff. Smaller cell size may be used to reduce the ratio, but it will introduce more communication cost and more complexity such as empty cells.

A.2 DEM-1 Chip Performance Estimation

As described in Fig. 6, the pipeline executes four cutoff operations per main clock cycle time, while 0.5 particle pair is fed through the main pipeline, which calculates two interactions on the same particle pair within two clock cycles. Therefore, the total effective operation of the DEM-1 pipeline is represented by;

$$C_{op} = 4 * R_{effcju} * C_{cju} + C_{main} \qquad (24)$$

where C_{op} is the total operations of the pipeline per clock, R_{effcju} is the efficiency of cutoff unit, C_{cju} is the number of operations of cutoff unit, C_{main} is the number of operations of the main pipeline, respectively. The efficiency of the cutoff unit is limited by the ratio R_{eff} (see Appendix A.1):

$$4 * R_{effcju} \leq 0.5 * R_{eff} \qquad (25)$$

This leads to $R_{effcju} = 0.81$, which indicates that cutoff units should sometimes stall. If we assume $C_{cju} = 9$ and $C_{main} = 21$ (we omit the operations of squared distance calculation in C_{main} because it has been already been calculated in C_{cju}), the overall performance is estimated to be 50 (= 4*0.81*9+21). We should take the non-vectorized nature of the hardware pipeline into account (see also Appendix A.1). The effective performance of the pipeline is 25 operations per clock. This corresponds to 3.33 Gflops with 133 MHz clock. The MIPS processor attached to the pipeline operates at 266 Mflops. We expect approximately 14.4 Gflops performance with four pipelines and four processors.

(Received September 23, 2002)
(Accepted February 4, 2003)

Ryo Takata received the B.E. and M.E. degrees from the University of Tokyo in 1980 and 1982 respectively. He works in Image Technology Laboratory Corporation since 1986. He is working on a Ph.D. in the Graduate School of Information Systems, University of Electro-Communications. His research interests include particle simulation machine. He is a student member of IPSJ, ACM and a member of IEICE.

Kenji Kise received the B.E. degree from University of Nagoya in 1995. He received Ph.D. degree from the University of Tokyo in 2000. Sine 2000 he has been in the Graduate School of Information Systems, University of Electro-Communications as a Teaching Assistant. His research interests include computer architecture.

Hiroki Honda received the B.E., M.E., and Ph.D. degrees from Waseda University in 1984, 1986 and 1991, respectively. From 1987 to 1991, he worked for Waseda University as a Research Assistant. In 1991, he joined Yamanashi University, and was a Lecture from 1991 to 1992, an Associate Processor from 1992 to 1997. He is currently an Associate Professor of the Graduate School of Information Systems, University of Electro-Communications. His research interests include parallel processing, parallelizing compiler, parallel computer architecture and grid computing. He is a member of IPSJ, IEEE-CS and ACM.

Toshitsugu Yuba received the B.E. and M.E. degrees from Kobe University in 1964 and 1966, and the Ph.D. degree from the University of Tokyo in 1982. In 1966 he joined the Nomura Research Institute. From 1967 to 1993, he was with the Electrotechnical Laboratory (ETL) of the Ministry of International Trade and Industry. His last position at ETL was the director of the Computer Science Division. Since 1993, he is the professor of the Graduate School of Information Systems, the University of Electro-Communications. His current research interests are parallel/distributed processing, high-performance computing and networking. Dr. Yuba is a fellow of the Institute of Electronics, Information and Communication Engineers, and the Information Processing Society of Japan. He is also a member of the Japan Society for Software Science and Technology, the Robotics Society of Japan, ACM and IEEE.