

УДК 681.3

М.К. Буза, О.М. Кондратьева

**ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДА  
МОЛЕКУЛЯРНОЙ ДИНАМИКИ НА МНОГОЯДЕРНОМ ПРОЦЕССОРЕ**

*Предлагается методика распараллеливания пакета XMD молекулярной динамики на базе вычислительной системы с общей памятью. Выполняется параллельная реализация алгоритма молекулярной динамики в форме многопоточного Windows-приложения. Исследуется эффективность полученной реализации для моделирования ряда процессов в физике полупроводников на многоядерных компьютерах.*

**Введение**

В настоящее время не только суперкомпьютеры, но и персональные компьютеры и ноутбуки основаны на технологиях, поддерживающих параллельную обработку. На сегодняшний день разработан, отлажен и апробирован огромный объем программного обеспечения. Естественно, хотелось бы его эффективно использовать при работе на многопроцессорных и многоядерных вычислительных установках. Для этого необходимо создать механизм разделения последовательного алгоритма на части и организовать их параллельное выполнение на разных процессорах.

В статье рассматривается алгоритм классической молекулярной динамики (МД). Для описания межатомных взаимодействий используется эмпирический потенциал, а динамика ансамбля атомов описывается системой уравнений движения Ньютона. Моделирование методом МД относится к классу больших задач и активно исследуется на предмет распараллеливания. Существуют различные пакеты МД-моделирования, ориентированные как на последовательные, так и на параллельные вычисления.

Цель настоящей работы – увеличить производительность пакета XMD (Molecular Dynamics for Metals and Ceramics) [1], ставшего практически стандартным для МД-моделирования. Для достижения поставленной цели к пакету XMD добавлены многопоточные версии функций вычисления сил, что позволяет сократить время моделирования на многопроцессорных компьютерах с общей памятью. Эффективность модифицированного пакета была исследована при решении следующих задач по моделированию в физике полупроводников: отжиг, диффузия, аморфизация и распыление.

**1. Метод МД**

Основной идеей метода молекулярной динамики [2] является то, что атомы вещества рассматриваются как точечные массы, движение которых описывается классическими уравнениями Ньютона. Интегрирование этих уравнений позволяет найти траектории движения всех атомов в заданном объеме пространства. Анализируя их, можно получить необходимую информацию о моделируемой системе в целом: плотности, температуре, фазовом состоянии, дефектах (в случае кристалла) и т. п.

Пусть  $N$  – число атомов в рассматриваемом объеме. Обозначим  $\vec{r}_i, \vec{v}_i, \vec{p}_i$  соответственно трехмерные координаты, скорости и импульсы атомов;  $\vec{F}_i$  – силы, действующие на атомы,  $i = 1, \dots, N$ . Тогда поведение атомов, образующих данный ансамбль, описывается системой уравнений

$$\begin{cases} \frac{d\vec{r}_i}{dt} = \vec{v}_i; \\ \frac{d\vec{p}_i}{dt} = \vec{F}_i. \end{cases} \quad (1)$$

Силы межатомного взаимодействия можно найти как градиент потенциальной энергии системы, причем энергия определяется из потенциалов взаимодействия для входящих в нее

атомов. Межатомные потенциалы в классической молекулярной динамике являются эмпирическими [3] и вычисляются для данного типа атомов подгонкой набора параметров потенциала к набору избранных свойств реальной системы. Набор параметров определяется функциональной формой потенциала, а свойства реальной системы – экспериментально или из квантово-механических расчетов.

По причине ограниченности вычислительных ресурсов при использовании метода МД обычно рассматриваются системы, содержащие от нескольких сотен до нескольких сотен тысяч атомов, в реальных же системах число атомов значительно больше. Поэтому в модельных системах возникают искусственные поверхностные эффекты, вызванные тем, что аномально большая доля атомов находится на поверхности. Для преодоления этой трудности используют периодические граничные условия, когда моделирование выполняется для частиц, лежащих в пределах ограниченной ячейки (как правило, имеющей форму прямоугольного параллелепипеда), а атомы, пересекающие стенку ячейки и выходящие наружу, входят в ячейку вновь с противоположной стороны. Накладывают ограничение также на максимальное расстояние взаимодействия между частицами, вводя так называемый радиус обрезания.

Потенциальная энергия моделируемой системы  $V(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$  определяет силы, действующие на атомы в данный момент времени, и является, вообще говоря, функцией координат всех частиц. В данной работе рассматриваются системы атомов, для которых не учитываются кулоновские взаимодействия и, следовательно, используются короткодействующие потенциалы. Дополнительным общепринятым упрощением для функциональной формы межатомных потенциалов является то, что потенциальная энергия  $V(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$  есть сумма функциональных выражений, зависящих лишь от пар и троек атомов, входящих в систему, а более сложные групповые взаимодействия не учитываются [2].

В этом случае формула для потенциальной энергии имеет вид

$$V(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) = \sum_{i=1, N} \sum_j \Phi(\vec{r}_i, \vec{r}_j) + \sum_{i=1, N} \sum_j \sum_k G(\vec{r}_i, \vec{r}_j, \vec{r}_k), \quad (2)$$

где функции  $\Phi(\vec{r}_i, \vec{r}_j)$  и  $G(\vec{r}_i, \vec{r}_j, \vec{r}_k)$  описывают парные взаимодействия и взаимодействия троек атомов соответственно, индексы  $j, k$  внутренних сумм пробегают множество номеров атомов, расположенных вблизи атома  $i$  в пределах радиуса обрезания, которое называется множеством соседей.

Таким образом, алгоритм МД может быть представлен как последовательность следующих шагов:

1. Инициализация координат и скоростей всех атомов.
2. Построение списка соседей для каждого атома.
3. Вычисление потенциальной энергии системы и сил, действующих на атомы, используя межатомный потенциал.
4. Нахождение новых положений и скоростей всех атомов на основе вычисленных сил через некоторый малый временной интервал.

Шаги 2–4 повторяются в течение всего времени моделирования, позволяя найти положения и скорости атомов на каждом временном шаге.

## 2. Выбор пакета МД-моделирования

При выборе пакета программ МД для решения требуемых задач сначала были отобраны те пакеты, которые являются доступными, хорошо себя зарекомендовавшими и документированными. Кроме того, они должны были включать достаточно широкий набор эмпирических потенциалов для полупроводниковых ковалентных материалов. Коммерческие, чрезмерно сложные и ресурсоемкие пакеты не рассматривались. В результате отбора остались кандидатуры XMD [1], LAMMPS [4], DL\_POLY [5], IMD [6]. Дальнейшая работа с пакетами: установка, настройка, изучение документации и исходных кодов, запуск тестовых задач – показала, что наиболее подходящим для решения требуемых задач по моделированию является пакет XMD.

Он включает необходимый набор потенциалов и содержит большое количество примеров входных файлов для различных ситуаций моделирования, а также обладает понятной структурой исходного кода, что упрощает модификацию пакета.

Более того, пакет XMD может быть настроен для параллельного выполнения на компьютерах с общей памятью, которые поддерживают POSIX-потoki. Однако в XMD параллельное выполнение поддерживается только для вычисления EAM-потенциалов [7], предназначенных для металлов. Для решаемых задач нужны потенциалы, предназначенные для ковалентных материалов, например Tersoff [3]. Сказанное позволяет сформулировать задачу расширения XMD для параллельного выполнения с использованием необходимых потенциалов.

### 3. Распараллеливание алгоритма МД

Алгоритм МД обладает естественным параллелизмом: расчет потенциальной энергии системы, сил, координат и скоростей можно выполнять одновременно для всех атомов в пределах одного временного шага. Тогда формула (2) преобразуется к виду

$$V(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N) = \sum_{i=1, n} \left( \sum_{\beta_i} \sum_j \Phi(\vec{r}_i, \vec{r}_j) + \sum_{\beta_i} \sum_j \sum_k G(\vec{r}_i, \vec{r}_j, \vec{r}_k) \right), \quad (3)$$

где  $\{\beta_1, \dots, \beta_n\}$  – разбиение множества из  $N$  атомов на  $n$  блоков. Если вычислительная система имеет  $n$  процессоров, то можно параллельно вычислить  $n$  сумм. Таким образом, вычисления, соответствующие шагу 3 алгоритма МД, могут быть разделены на независимые части.

Выделяют два основных подхода [8, 9] к решению задачи по разработке параллельной версии МД. Эти подходы определяют способы построения разбиений множества атомов на блоки. При первом подходе набор частиц разбивается на подмножества по количеству процессоров и каждому процессору ставится в соответствие определенное подмножество. Данное соответствие остается неизменным в ходе всего моделирования. Этот подход имеет различные варианты: декомпозиция по частицам, силовая декомпозиция, метод параллельных реплик и др. Во втором подходе, который называют методом пространственной декомпозиции, каждый процессор связывается с определенной подобластью всей расчетной ячейки. Для частиц, находящихся в данной подобласти, закрепленный за ней процессор ведет расчет сил и интегрирование уравнений движения. Поскольку частицы могут в процессе моделирования переходить из одной подобласти в другую, происходит обмен данными между процессорами на каждом временном шаге.

Анализ исходного кода пакета XMD показал, что в существующем параллельном варианте вычисления EAM-потенциалов разработчики используют первый подход: список всех частиц делится на равные части и каждый процессор вычисляет сумму, соответствующую распределенным ему частицам.

### 4. Распараллеливание пакета XMD на многоядерном процессоре

Для многоядерных процессоров естественным является использование моделей параллельных программ с общей памятью. Как правило, технологии разработки программ в этом случае проще, а накладные расходы меньше по сравнению с использованием распределенной памяти.

В рамках технологии программирования, которая опирается на общую память, параллельную программу будем представлять следующим образом. Текст программы состоит из последовательных и параллельных областей. При запуске приложения создается первичный поток, который начинает выполнение программы со стартовой точки. Первичный поток исполняет все последовательные области программы. При входе в параллельную область первичный поток порождает дополнительные потоки. Все порожденные потоки исполняют один и тот же код, соответствующий параллельной области. При выходе из параллельной области первичный поток дожидается завершения остальных потоков и продолжает дальнейшее выполнение программы. Потоки осуществляют взаимодействие через общие переменные, которые существуют в одном экземпляре и доступны всем потокам.

Обратимся к пакету XMD. Последовательная версия функции вычисления сил, например, с использованием потенциала Tersoff (функция `csi_calcforg()`) состоит из следующих шагов:

1. Инициализация локальных переменных.
2. Построение списка соседей.
3. Вычисление вектора силы для каждой частицы.

Функция `csi_calcforg()` вызывается на каждом шаге времени моделирования и реализует вычисления, соответствующие шагам 2 и 3 алгоритма МД. В качестве параметра функция получает список всех частиц. Таким образом, вычисление вектора силы для каждой частицы – та часть функции, которую можно реализовать как многопоточную, используя декомпозицию по частицам. Более того, профилировка кода пакета XMD показала, что для рассматриваемых задач в среднем 95 % времени занимают вычисления сил с использованием потенциала Tersoff, из которых 65 % тратится на вычисление векторов сил.

При модификации пакета XMD в данной работе был применен первый подход к распараллеливанию алгоритма МД, когда на каждом временном шаге весь массив частиц делится на равные части и вычисление поля сил проводится независимо на каждом вычислительном узле. Тогда вычисление векторов сил включает следующие этапы:

- подготовка данных для потоковых функций;
- создание потоков для вычисления векторов сил;
- ожидание завершения всех дополнительных потоков;
- закрытие потоков.

Потоковая функция вычисляет вектор силы для каждой частицы из назначенного ей диапазона частиц. Функция производит вычисления в локальных переменных, а перед завершением изменяет значение общих переменных. Для организации взаимоисключающего доступа к общим переменным используется примитив мьютекс. Эта завершающая часть является дополнительным участком последовательного кода.

Накладные расходы включают: создание потоков, обмен данными между потоками (создание мьютекса, взаимоисключающий доступ к общим переменным), синхронизацию (ожидание завершения всех дополнительных потоков), закрытие потоков, операции выделения и инициализации памяти под локальные переменные.

В результате реализованы многопоточные Windows-версии [10] функций вычисления сил для ковалентных материалов. Многопоточные функции позволяют использовать пакет XMD для параллельного выполнения на компьютерах с общей памятью.

## 5. Описание моделируемых процессов

Для исследования эффективности построенного расширения пакета XMD были отобраны типичные процессы, используемые при моделировании в физике полупроводников (табл. 1).

Таблица 1

Основные характеристики моделируемых процессов

Название	Описание	Материал	Используемые потенциалы
Диффузия	Система находится при постоянной температуре, отслеживается самодиффузия атомов решетки	Кремний, карбид кремния	SW
Отжиг	Система находится при постоянной высокой температуре, отслеживаются диффузия и рекомбинация дефектов решетки	Кремний	EDIP, SW
Аморфизация	Облучение системы низкоэнергетическими частицами до перехода в аморфное состояние	Кремний	EDIP, SW
Распыление	Ускоренные атомы падают извне на поверхность, вызывая распыление	Кремний, германий	TERSOFF

## 6. Тестирование параллельной версии пакета XMD

Для определения качества параллельной версии программы использовались показатели ускорения и эффективности. Ускорение  $S(k)$  для  $k$  потоков определялось в соответствии с формулой

$$S(k) = \frac{T(1)}{T(k)},$$

где  $T(k)$  – время выполнения расчета для  $k$  потоков;  $T(1)$  – время выполнения последовательной программы.

Эффективность  $E(k)$  для  $k$  потоков определялась в соответствии с формулой

$$E(k) = \frac{S(k)}{k}.$$

Эксперименты проводились на персональном компьютере с четырехъядерным процессором Intel Core 2 Quad Q9400 и ноутбуке с двухъядерным процессором Celeron Dual-Core T3000 под управлением операционной системы Microsoft Windows XP.

Для исследования эффективности и корректности параллельной программы были подготовлены наборы входных данных, которые соответствуют моделируемым процессам и сгруппированы в тесты «Диффузия», «Отжиг», «Аморфизация» и «Распыление».

Профилировка кода последовательной версии пакета XMD показала, что доля вычислений, которые не распараллеливаются и связаны в основном с построением списка соседей, зависит не только от типа задачи, но и от ее размерности. Эта доля растет по мере увеличения количества частиц в моделируемой системе. Она составляет 7–37 (среднее 21) % для теста «Отжиг», 12–40 (26) % – «Диффузия», 23–51 (41) % – «Аморфизация», 67–85 (76) % – «Распыление». Ясно, что эффективность распараллеливания также зависит от типа и размера рассматриваемых задач.

В результате экспериментов, проведенных на персональном четырехъядерном компьютере, были определены зависимости показателей эффективности многопоточной версии программы от количества потоков (рис. 1–4). Ускорение и эффективность определялись для распараллеливаемой области текста программы (на участке) и всего пакета (общее). Ускорение и эффективность учитывают накладные расходы. Показатели эффективности вычислялись как среднее значение показателей, полученных для различных входных наборов соответствующего теста.

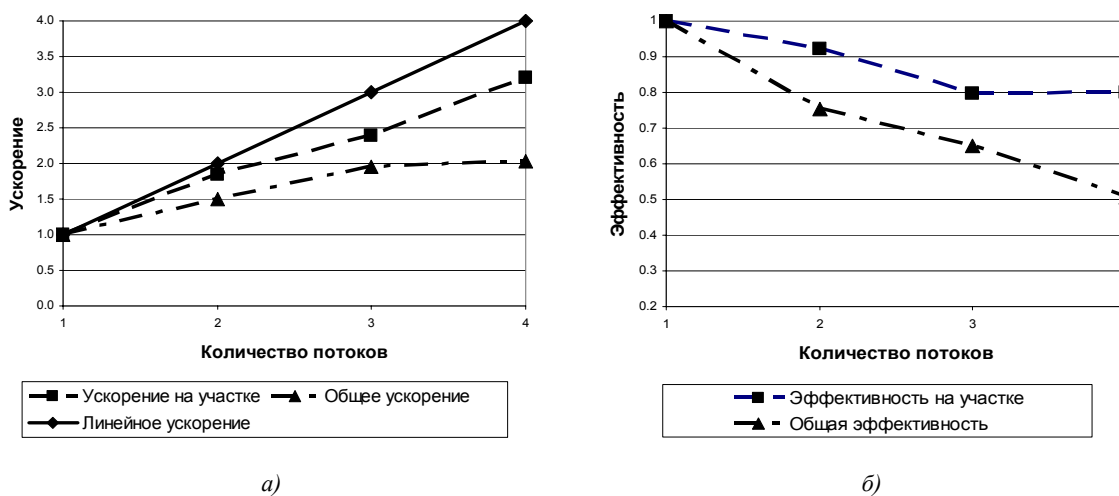


Рис. 1. Показатели для теста «Отжиг»: а) ускорение; б) эффективность

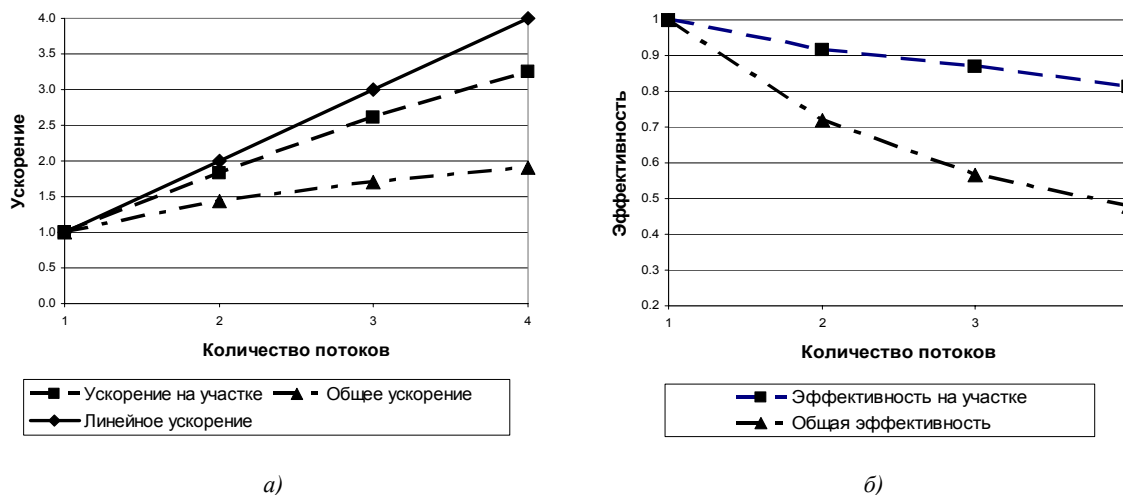


Рис. 2. Показатели для теста «Диффузия»: а) ускорение; б) эффективность

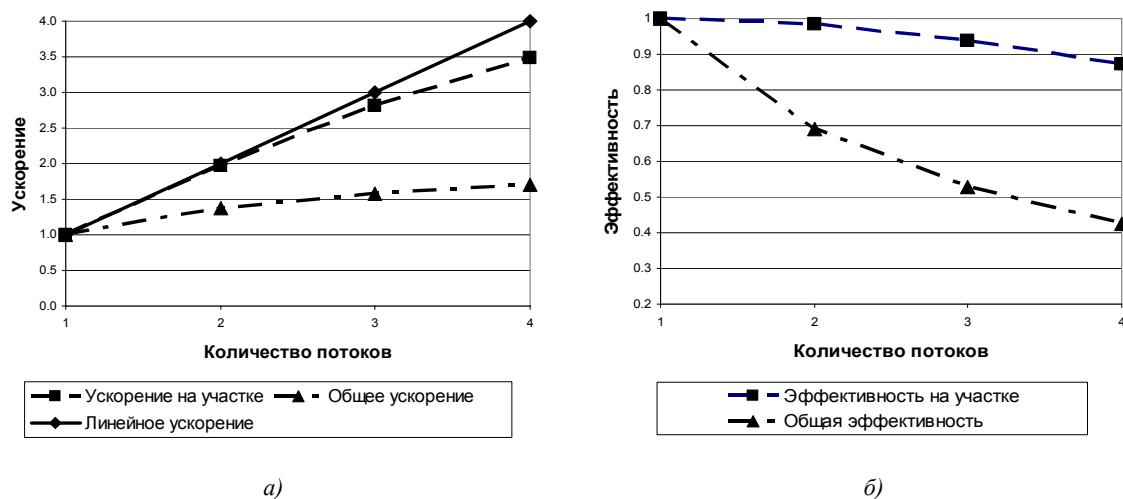


Рис. 3. Показатели для теста «Аморфизация»: а) ускорение; б) эффективность

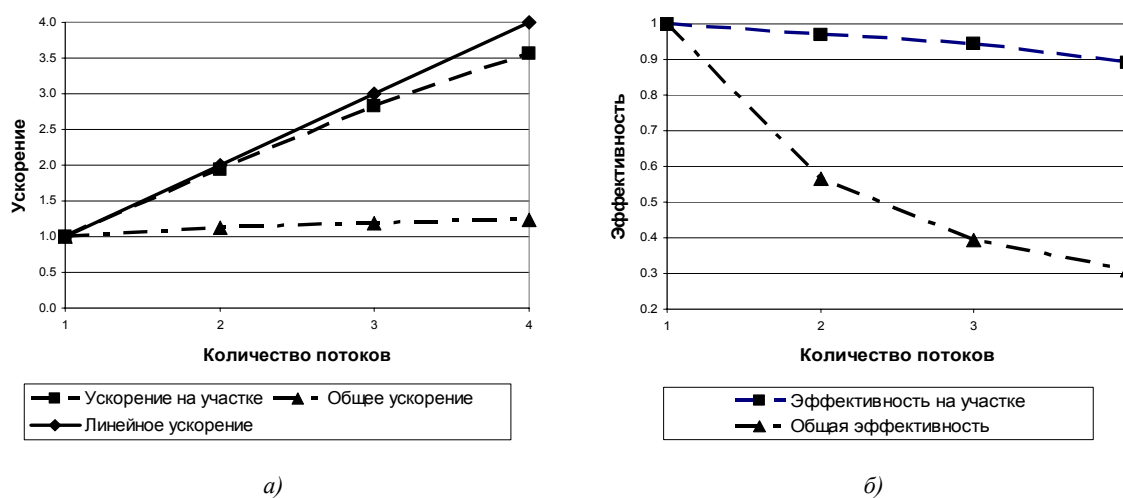


Рис. 4. Показатели для теста «Распыление»: а) ускорение; б) эффективность

Из полученных результатов следует:

– ускорение на участке является почти линейным, что обусловлено хорошими свойствами используемого параллельного алгоритма и малыми накладными расходами на поддержку потоков;

– общие ускорение и эффективность самые высокие у задач типа «Отжиг» и «Диффузия», так как они имеют самый низкий процент последовательных вычислений, связанных с построением списка соседей;

– наибольшая эффективность достигается при использовании двух потоков, когда выигрыш во времени, получаемый за счет параллельного выполнения потоков, максимально превышает временные затраты, связанные с организацией потоков;

– задачи «Отжиг», «Диффузия» и «Аморфизация» дают приемлемое общее ускорение для двухпоточного варианта, равное примерно 1,5;

– увеличение количества ядер не повышает эффективность, так как ресурс параллелизма полностью расходуется при использовании двух-четырех ядер.

Результаты экспериментов, проведенных на двухъядерном ноутбуке для варианта «два потока», практически совпадают с результатами, полученными на четырехъядерном персональном компьютере для варианта «два потока». В качестве примера в табл. 2 более подробно приведены различные количественные показатели теста «Отжиг».

Таблица 2

Показатели теста «Отжиг»

Количество частиц	Время, затрачиваемое на построение списка соседей, %	Ускорение на участке для двух потоков		Общее ускорение для двух потоков	
		четыре ядра	два ядра	четыре ядра	два ядра
516	7,13	1,64	1,56	1,53	1,48
1728	15,11	1,85	1,80	1,57	1,55
4096	22,88	1,89	1,87	1,52	1,50
8192	23,43	1,91	1,89	1,51	1,51
16 384	36,96	1,91	1,90	1,39	1,39
Среднее значение параметров					
–	21,10	1,84	1,80	1,50	1,49

### Заключение

Выполнено расширение хорошо зарекомендовавшего себя пакета XMD, позволяющее использовать его для параллельного выполнения МД-моделирования на компьютерах с общей памятью. Добавленные к пакету многопоточные функции поддерживают вычисления эмпирических потенциалов для полупроводниковых ковалентных материалов.

Открытый код пакета XMD позволил расширить его без переработки архитектуры, что уменьшает риск внесения новых ошибок и позволяет выставить новую версию для широкой группы исследователей, использующих базовый вариант.

Проведенное тестирование показало, что существует ряд задач, для которых можно значительно повысить быстродействие пакета и эффективно использовать как многопроцессорные системы с общей памятью, так и настольные многоядерные компьютеры.

### Список литературы

1. XMD – Molecular Dynamics for Metals and Ceramics [Electronic resource]. – March, 2010. – Mode of access : <http://xmd.sourceforge.net/about.html>. – Date of access : 01.09.2010.
2. Haile, J.M. Molecular Dynamics Simulation. Elementary Methods / J.M. Haile. – N.Y., Chichester, Brisbane, Toronto, Singapore : John Wiley & Sons Inc., 1992. – 490 p.

3. Highly optimized empirical potential model of silicon / T.J. Lenosky [et al.] // *Modelling Simul. Mater. Sci. Eng.* – 2000. – Vol. 8. – P. 825–841.
4. LAMMPS Molecular Dynamics Simulator [Electronic resource]. – January, 2009. – Mode of access : <http://lammps.sandia.gov/>. – Date of access : 01.09.2010.
5. The DL\_POLY Molecular Simulation Project [Electronic resource]. – January, 2010. – Mode of access : [http://www.cse.scitech.ac.uk/ccg/software/DL\\_POLY/](http://www.cse.scitech.ac.uk/ccg/software/DL_POLY/). – Date of access : 01.09.2010.
6. The ITAP Molecular Dynamics Program [Electronic resource]. – March, 2010. – Mode of access : <http://www.itap.physik.uni-stuttgart.de/~imd/index.html>. – Date of access : 01.09.2010.
7. Mishin, Y. Interatomic potentials for monoatomic metals from experimental data and ab initio calculations / Y. Mishin, D. Farkas // *Phys. Rev. B.* – 1999. – Vol. 59. – P. 3393–3407.
8. Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics / S. Plimpton // *Journal of Computational Physics.* – 1995. – Vol. 117, № 1. – P. 1–19.
9. Smith, W. Molecular dynamics on hypercube parallel computer / W. Smith // *Computer Physics Communications.* – 1991. – Vol. 62. – P. 229–248.
10. Рихтер, Дж. Windows для профессионалов: создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows / Дж. Рихтер. – СПб. : Питер, 2001. – 752 с.

Поступила 12.11.10

*Белорусский государственный университет,  
Минск, пр. Независимости, 4  
e-mail: bouza@bsu.by,  
kondratjeva@bsu.by*

**M.K. Bouza, O.M. Kondratjeva**

## **SIMULATION OF MOLECULAR DYNAMICS ON MULTI-CORE PROCESSORS**

Methods for parallelization of the molecular dynamics package XMD are proposed using shared memory systems. A parallel version of the molecular dynamics algorithm was implemented in the form of multithread Windows application. The efficiency of the implementation was investigated on the modeling problems in semiconductor physics using multi-core computers.