

ИСПОЛЬЗОВАНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ НА БАЗЕ ТЕХНОЛОГИИ CUDA ПРИ МОДЕЛИРОВАНИИ ПТШ

Э.Э. АСАНОВ, С.А. ЗУЕВ, Г.В. КИЛЕССА, Н.И. СЛИПЧЕНКО

В работе представлены основные теоретические положения, на основе которых построена реализация численной модели ПТШ на GaAs, проведены исследования характеристик ПТШ. Для увеличения производительности модели отдельные этапы вычисления проводились на GPU с использованием технологии NVIDIA CUDA.

Ключевые слова: полевой транзистор с затвором Шоттки, моделирование методом макрочастиц, параллельные вычисления на базе CUDA.

ВВЕДЕНИЕ

Наиболее существенным недостатком существующих в настоящее время численных моделей вакуумных и полупроводниковых приборов являются скорость и, соответственно, время вычислений. Особенно это актуально для моделей, использующих кинетические подходы при моделировании. Большинство численно-аналитических моделей электронных приборов были разработаны достаточно давно, но и в настоящее время они широко эксплуатируются для исследования физических процессов, оптимизации характеристик приборов и т.д. Основной причиной низкой скорости вычислений является использование архитектуры SISD (Single instruction – single data). В настоящее время широкое применение для ускорения процессов моделирования находят методы параллельных вычислений.

Целью данной работы является демонстрация возможности и опыта использования параллельных вычислений на базе технологии CUDA (Compute Unified Device Architecture [1]) в модели полевого транзистора с затвором Шоттки.

ИСХОДНЫЕ ПОЛОЖЕНИЯ

Для построения достоверной модели процессов, происходящих в активной области транзистора, необходимо рассчитать движение свободных носителей заряда этой области во внешнем поле и полях зарядов внутренней области. В кинетическом приближении эволюция ансамбля частиц описывается уравнением Больцмана [2]:

$$\frac{\partial f}{\partial t} + \sum_i \frac{\partial f}{\partial \vec{r}_i} \frac{d\vec{r}_i}{dt} + \sum_i \frac{\partial f}{\partial \vec{p}_i} \frac{d\vec{p}_i}{dt} = S, \quad (1)$$

где f – функция распределения, \vec{r}_i и \vec{p}_i – радиус-векторы и импульсы свободных носителей заряда, S – интеграл столкновений.

Данное уравнение решается методом макрочастиц, при этом система (1) сводится к решению системы уравнений движения для макрочастиц в кристалле с учетом механизмов рассеяния:

$$\frac{d\vec{r}}{dt} = \frac{1}{\hbar} \frac{d\epsilon_k}{d\vec{k}} \quad \text{и} \quad \frac{d\vec{k}}{dt} = \frac{1}{\hbar} \frac{\vec{F}}{m^*}, \quad (2)$$

где ϵ_k , m^* – энергия и эффективная масса носителя заряда, соответственно, \vec{F} – действующая на него внешняя сила.

Для определения силы в (2), решается уравнение Пуассона:

$$\Delta\varphi = \frac{4\pi e}{\epsilon\epsilon_0} \left[\sum_i \int f_i d\vec{k} - N \right], \quad (3)$$

где $N = N(r)$ – распределение ионизированных атомов в решетке, ϵ – диэлектрическая проницаемость кристалла.

Интеграл столкновений в правой части кинетического уравнения Больцмана (1) интерпретируется через вероятности рассеяния на различных центрах. В модели учтены следующие механизмы рассеяния [2]:

- рассеяние на акустических фононах;
- рассеяние на оптических фононах;
- электрон-электронное рассеяние;
- рассеяние на нейтральных примесях;
- рассеяние на ионах примеси.

Каждый процесс рассеяния описывается правилами: алгоритмом для расчета нового волнового вектора \vec{k}' из предыдущего его значения \vec{k} и формулой для вычисления вероятности процесса рассеяния.

Если полная интенсивность рассеяния для всех процессов является постоянной, скажем Γ [с⁻¹], то время свободного пробега электрона между двумя рассеяниями равно [3]:

$$\delta t = -\frac{\ln(1-r)}{\Gamma}, \quad (4)$$

где r – случайное число из однородного вероятностного распределения в диапазоне [0,1).

Для реальных процессов полная интенсивность рассеяния не является постоянной, поэтому используется дополнительный (фиктивный) механизм «саморассеяния», который не дает изменений волнового вектора частицы, а его интенсивность при любой энергии выбирается такой, чтобы привести интенсивность рассеяния к выбранной постоянной Γ .

По истечению времени свободного пробега выбор процесса рассеяния проводится с помощью случайной величины r^* , равномерно распределенной в интервале $0 \leq r^* < \Gamma$.

Процесс рассеяния с номером m выбирается из следующего условия:

$$\sum_{i=1}^{m-1} \lambda_i(k) \leq r^* < \sum_{i=1}^m \lambda_i(k), \quad (5)$$

где $\lambda_i(k)$ – вероятность рассеяния для i -го процесса.

РЕАЛИЗАЦИЯ ЧИСЛЕННОЙ МОДЕЛИ

Численная модель ПТШ написана на языке C++ с использованием технологии объектно-ориентированного программирования. Структурная схема базовых классов приведена на рис. 1.

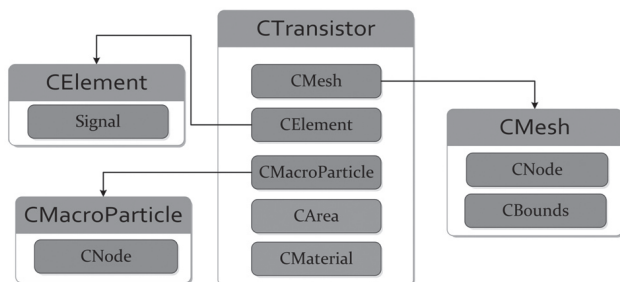


Рис. 1. Структурная схема базовых классов модели ПТШ

Класс CTransistor – это основной класс, отвечающий за взаимодействие отдельных составляющих модели. CElement реализует модель контакта, а также содержит в себе сигнал, подаваемый на этот контакт. Сигнал создается классом Signal и представляет собой периодическую дискретную функцию $U = U(t)$, где $t = t_0, t_1 \dots t_N$, с периодом T . CMesh содержит в себе дискретное множество узлов, составляющих сетку, наложенную на область моделирования ПТШ. Каждый узел – это экземпляр класса CNode, который содержит следующие параметры: плотность заряда, потенциал, концентрацию примесей, а также температуру. CMesh включает в себя и класс CBounds, который отвечает за задание граничных условий. Моделируемая область прибора разбивается пользователем на зоны (класс Area) с различными значениями концентрации основных носителей заряда. Кроме этого в CTransistor хранится список макрочастиц, представленных экземплярами класса CMarcoParticle. CMarcoParticle кроме основных параметров (заряд, масса, позиция, волновой вектор и энергия) содержит в себе указатель на верхний левый узел ячейки, в которой он находится. Это позволяет сократить время, затрачиваемое на вычисления при размазывании заряда, т.к. не приходится определять, в какой ячейке находится макрочастица. CMaterial – абстрактный класс, который содержит в себе характеристики материала моделируемого прибора, а также абстрактные функции рассеяния. Классы, наследуемые от CMaterial, реализуют эти функции в соответствии с формулами для частот рассеяния для данного материала.

Общая блок-схема, демонстрирующая алгоритм работы модели, приведена на рис. 2.



Рис. 2. Общая блок-схема работы модели

Как видно из рис. 2, после запуска программы пользователю необходимо задать параметры исследуемого ПТШ, после чего происходит инициализация объекта типа CTransistor. После создания экземпляра класса CTransistor модель находится в режиме ожидания и только при вызове функции CTransistor::Update() происходит обновление на одну итерацию. Вызов функции CTransistor::Update() осуществляет Управляющий класс, который также отвечает за создание окна для вывода результатов работы и визуализирует текущее состояние. Процесс «Обновление на шаг DT» является основным в модели. Именно на этом этапе выполняются такие функции, как размазывание зарядов, расчет потенциалов, рассеивание частиц. Развернутая блок-схема данной процедуры приведена на рис. 3.

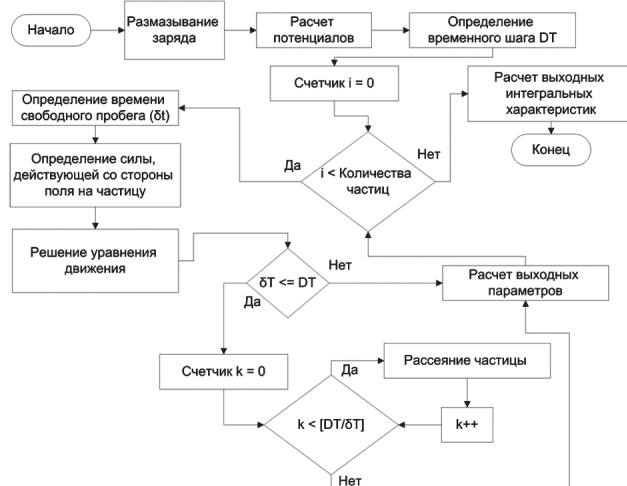


Рис. 3. Блок-схема процесса «Обновление на шаг DT»

Первым шагом алгоритма является размазывание заряда, необходимое для того, чтобы получить более гладкую функцию распределения. Затем решается уравнение Пуассона, в результате чего мы получаем значение потенциалов в узлах сетки. После расчета потенциалов происходит определение временного шага DT, т.е. временной шаг не является постоянной величиной, задаваемой пользователем или вычисляемой при инициализации прибора. DT определяется из следующего условия: носитель заряда с наибольшей

скоростью за время равное временному шагу должен пролетать не более одной ячейки.

На следующем этапе запускается цикл и для каждой макрочастицы определяется время свободного пробега, силы, действующие со стороны поля, и решаются уравнения движения. После чего сравнивается время свободного пробега с временным шагом ΔT и, если оно меньше или равно ΔT , то выполняется рассеяние этой макрочастицы (количество рассеяний может быть более одного) с выбором механизма рассеяния по методу Монте-Карло. По завершению цикла происходит расчет выходных интегральных характеристик (например, таких как токи, протекающие через контакты и др.).

РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЙ

Для получения результатов с приемлемой точностью необходимы большие временные затраты, даже при выполнении расчета на мощном компьютере. При использовании одного CPU достичь высокой скорости расчета основных параметров ПТШ не представляется возможным, поэтому необходимо распараллелить вычислительные процессы при реализации модели. Для выполнения параллельных вычислений используется Graphics Processing Unit (GPU) с применением технологии CUDA [1].

На начальном этапе был проведен анализ алгоритма и выявлены участки программного кода, которые могут быть легко распараллелены. Критерием выбора служило то, что выполняемая процедура могла быть легко реализована с использованием архитектуры SIMD (Single instruction – multiple data) [4]. Анализ процесса вычислений показывает, что целесообразно распараллелить операции, которые выполняются на каждом шаге моделирования и занимают значительную часть вычислительного времени – «Расчет потенциалов» и «Рассеяние электрона». Данный выбор позволяет достичь максимального ускорения в вычислениях. В дальнейшем возможна разработка алгоритмов параллельных вычислений для остальных процедур, приведенных на рис. 3 (решение уравнения движения, рассеяние частиц).

Рассмотрим алгоритм расчета потенциалов. Как упоминалось выше, для нахождения потенциалов необходимо решить уравнение Пуассона (3). Для его решения был выбран метод FACR (Fourier Analysis and Cyclic Reduction) [5]. В Фурье-анализе рассчитывается большое количество сумм, следовательно, можно предположить, что, распараллелив эту часть алгоритма, мы получим большой выигрыш по времени. Разработанный алгоритм для выполнения Фурье-анализа на GPU представлен на рис. 4.

По мере формирования матрицы плотностей зарядов с соответствующими граничными условиями (процедура «Размазывание заряда» на рис. 3) CPU отдает команду на ее копирование из

DRAM в GPU DRAM (на рис. 4 команда «Copy in GPU»). После того, как копирование завершится, CPU подает следующую команду GPU – запуск ядра («launch»), а далее вызывается специальная функция `cudaThreadSynchronize`, необходимая для синхронизации работы CPU и GPU. Как только на GPU поступит команда «launch», он начнет выполнять расчет («execution») над данными, находящимися в GPU DRAM, т.е. будет выполнять ядро, в то время как CPU будет ожидать «ответа» от GPU, который поступит («return») после того, как завершится выполнение ядра. Получив возможность выполнять программу дальше, следующим шагом CPU отдает команду («Copy from GPU DRAM») копирование данных из GPU DRAM в DRAM (если не вызывать функцию `cudaThreadSynchronize`, то CPU после команды «launch» сразу же даст команду «Copy from GPU DRAM», т.е. еще до завершения расчета («execution») данные скопируются из GPU DRAM в DRAM).

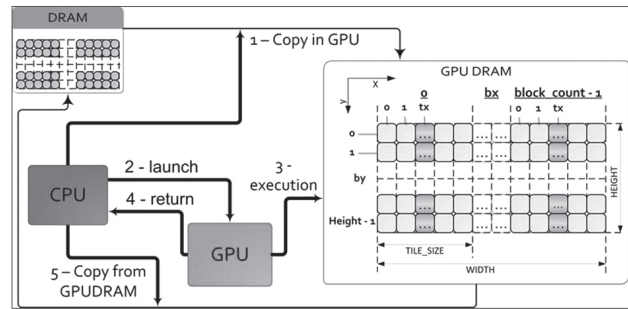


Рис. 4. Алгоритм для выполнения Фурье-анализа на GPU

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

В качестве примера был промоделирован прибор с геометрией (в мкм), приведенной на рис. 5.

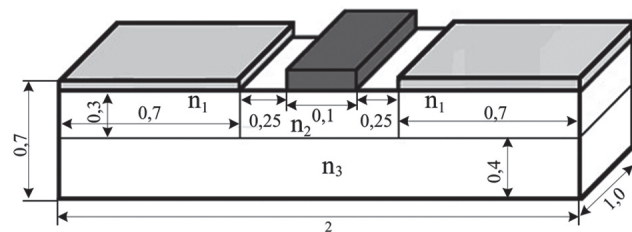


Рис. 5. Геометрия моделируемого ПТШ (размеры приведены в мкм)

Значения концентраций были заданы следующие: n_1 – контактный слой – $2 \cdot 10^{23} \text{ м}^{-3}$, n_2 – канал – 10^{21} м^{-3} , n_3 – $2 \cdot 10^{19} \text{ м}^{-3}$.

Для получения выходных характеристик ($I_C = I_C(U_C)$, при $U_3 = \text{const}$) на сток подавалось постепенно возрастающее напряжение, и после того как устанавливался «стационарный» режим, снимались значения I_C . Для получения сток-затворной характеристики ($I_C = I_C(U_3)$, при $U_C = \text{const}$), на затвор подавалось постепенно возрастающее напряжение при фиксированном напряжении на стоке.

Полученные ВАХ транзистора (рис. 6) имеют традиционный для полевых транзисторов вид.

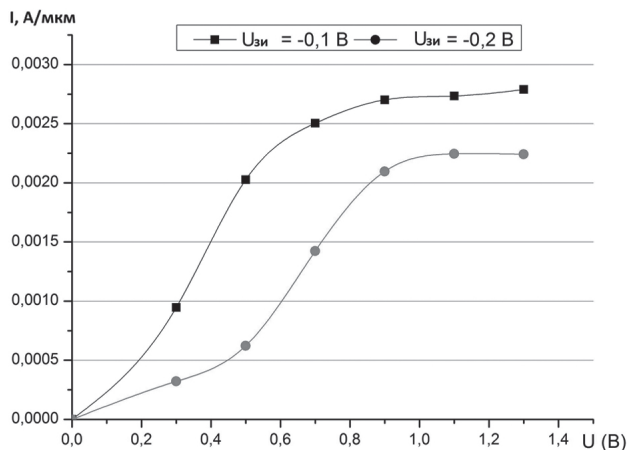


Рис. 6. ВАХ ПТШ с длиной затвора 100 нм (зависимость тока стока $I_C = I_C(U_C) U_3 = const$)

Как видно из рис. 6, разработанная модель адекватно описывает реальные приборы ПТШ. При росте модуля управляющего напряжения наблюдается уменьшение тока насыщения, т.е. запираение канала. Из приведенных зависимостей также видно, что ток насыщения при $U = 0$ В примерно равен 47 мА. Напряжение отсечки примерно равно -2 В. Таким образом, задавая различные значения концентраций и (или) геометрических размеров можно добиться необходимых расчетных параметров ПТШ.

ЗАКЛЮЧЕНИЕ

Построенная и реализованная модель с использованием параллельных вычислений ПТШ достоверно описывает процессы, происходящие в активной области транзистора. Разработанная модель позволяет получать как дифференциальные характеристики ПТШ (плотности зарядов, токов и т.п.), так и практически все интегральные характеристики, такие как ВАХ, АЧХ, предельные частотные характеристики. Переход на SIMD архитектуру (с реализацией на GPU и применением технологии CUDA) позволил получить ощутимый прирост в скорости счета. За счет распараллеливания расчетов достигнуто ускорение выполнения отдельных процедур в 300–350 раз и общее ускорение моделирования в 2,5–2,7 раза.

Литература

- [1] NVIDIA CUDA C Programming Guide. Version 3.2, 2010. – 183 p.
- [2] Шалимова К.В. Физика полупроводников. – М.: «Энергоатомиздат», 1985. – 392 с.

- [3] Реклайтис А.С., Мицкявичус Р.В. Новое в жизни, науке, технике: Метод Монте-Карло в физике полупроводников. – Москва: Знание, 1988. – 38 с.
- [4] David B.Kirk, Wen-mei W. Hwu. Programming Massively Parallel Processors. 2010 – 279 p.
- [5] Хокни Р., Иствуд Д. Численное моделирование методом частиц. – М.: Мир, 1987. – 638 с.
- [6] Помтер Д. Вычислительные методы в физике. – М.: Мир, 1975. – 392 с.

Поступила в редколлегию 31.05.2012

Асанов Эрнест Эдемович, фото и сведения об авторе см. на с. 382.

Зуев Сергей Александрович, фото и сведения об авторе см. на с. 382.

Килесса Григорий Владимирович, фото и сведения об авторе см. на с. 382.

Слипченко Николай Иванович, фото и сведения об авторе см. на с. 382.

УДК 621.382.323

Використання паралельних обчислень на базі технології CUDA при моделюванні ПТШ / Е.Е. Асанов, С.О. Зуев, Г.В. Килесса, Н.І. Слипченко // Прикладна радіоелектроніка: наук.-техн. журнал. – 2012. – Том 11. № 3. – С. 431–434.

У роботі подано основні теоретичні положення, на основі яких побудована реалізація чисельної моделі ПТШ на GaAs та проведені дослідження характеристик ПТШ. Для збільшення продуктивності моделі окремі етапи обчислення проводилися на GPU з використанням технології NVIDIA CUDA.

Ключові слова: польовий транзистор із затвором Шоткі, моделювання методом макрочастинок, паралельні обчислення на базі CUDA.

Лл. 06. Бібліогр.: 06 найм.

UDC 621.382.323

Using parallel calculations on the basis of the CUDA technology in Schottky FET modelling / E.E. Asanov, S.A. Zuev, G.V. Kilessa, N.I. Slipchenko // Applied Radio Electronics: Sci. Journ. – 2012. Vol. 11. № 3. – P. 431–434.

The basic theoretical provisions on the basis of which the Schottky GaAs FET numerical model has been constructed are presented. The researches of Schottky FET characteristics are carried out. To increase the productivity of the model separate calculation stages have been carried out on GPU with NVIDIA CUDA technology use.

Keywords: field-effect transistor with Schottky gate, modelling by macroparticles method, parallel calculations on the basis of CUDA.

Fig. 06. Ref.: 06 items.