

# 基于 FPGA 的分子动力学模拟匹配单元设计

王鑫<sup>1,2</sup>, 刘炫<sup>1,2</sup>

(1. 江南大学 教育部轻工过程先进控制重点实验室, 江苏 无锡 214122;

2. 江苏省未来网络创新研究院, 南京 211111)

**摘要:** 为减少分子动力学模拟中短程力计算的时间消耗, 设计并实现基于现场可编程门阵列的分子动力学模拟匹配单元; 理论上, 分析了分子动力学模拟中粒子间作用力的物理规律, 提出两种满足短程力计算要求的粒子对的筛选方法: 偏序法和平面法; 在技术上使用新兴的硬件描述语言 SpinalHDL, 在 Xilinx VirtexUltraScale+ HBM VCU128 FPGA 板卡上实现了匹配单元; 最后, 将硬件测试结果与理论结果进行对比, 验证了匹配单元可以有效过滤掉对短程力计算没有贡献的粒子对; 同时对使用偏序法、平面法和直接使用算法两种不同情况下的资源消耗进行对比分析, 表明使用偏序法、平面法可以节省系统 70% 的 DSP 资源。

**关键词:** 分子动力学; 现场可编程门阵列; 匹配单元; 并行计算; 短程力

## Design of Match Units for Molecular Dynamics Simulations Based on FPGA

WANG Xin<sup>1,2</sup>, LIU Xuan<sup>1,2</sup>

(1. Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University,

Wuxi 214122, China; 2. Jiangsu Future Networks Innovation Institute, Nanjing 211111, China)

**Abstract:** In order to reduce the time consumption of rang-limited force calculations in molecular dynamics simulations, a field-programmable gate array-based molecular dynamics simulation matching unit is designed and implemented. Theoretically, the physical laws of the inter-particle forces in the molecular dynamics simulations are analyzed, and two filtering methods for particle pairs that meet the requirements of the rang-limited force calculations are proposed: the partial-order method and the planar method. Technically, the matching unit was implemented on a Xilinx VirtexUltraScale+ HBM VCU128 FPGA board using SpinalHDL, which is an emerging hardware description language. Finally, the hardware test results are compared with the theoretical results to verify that the matching unit can effectively filter out the particle pairs that do not contribute to the rang-limited force calculation. At the same time, a comparative analysis of the resource consumption of using the partial order method, the plane method and the direct calculation method shows that the use of the partial order method and the plane method can save 70% of the DSP resources of the system.

**Keywords:** molecular dynamics; FPGA; match unit; parallel computing; rang-limited force

## 0 引言

分子动力学模拟利用计算机求解生物分子体系内原子和分子的运动方程来模拟这些粒子的运动轨迹, 从而获得系统的温度、体积、压力等宏观和微观过程量。分子动力学模拟在计算物理、计算化学、材料科学、生命科学、生物医学等多个领域有着广泛的应用, 已成为与实验同等重要的科学研究方法。

分子动力学模拟过程中, 需要花费大量时间用于计算范德华相互作用和静电相互作用, 而模拟具有现实意义的生物大分子体系动力学过程往往需要几个月甚至几年时间。因此, 针对生物大分子进行微秒甚至毫秒级别的模拟, 对范德华相互作用和静电相互作用的优化加速将是首要任务,

其中进行硬件加速是提高计算速度的重要方法。近年来, 随着集成电路技术的快速发展, 分子动力学模拟在专用集成电路 (ASIC, application specific integrated circuit) 以及现场可编程门阵列 (FPGA, field programmable gate array) 上都得到了很好的实现, 并在并行化、算法进展和硬件专业化方面进行了不同程度的改进。这些改进大大提升了系统的性能。需要指出的是, 相比于 ASIC, FPGA 设计灵活, 功能强大, 风险低。因此, FPGA 软硬件协同设计环境下的硬件加速技术具有重要的研究意义。

在分子动力学模拟中, 粒子间需要计算的力包括键合力和非键合力<sup>[15]</sup>, 其中非键合力包括范德华力和静电力。为了降低非键合力的计算复杂度, 将非键合力分解为短程

收稿日期: 2021-12-02; 修回日期: 2021-12-31。

基金项目: 国家自然科学基金(61703185); 高等学校学科创新引智计划项目(B12018); 未来网络科研基金项目(FNSRFP-2021-YB-11)。

作者简介: 王鑫(1981-), 男, 山西忻州人, 工学博士, 讲师, 主要从事信号处理算法的设计与加速方向的研究。

刘炫(1995-), 女, 安徽安庆人, 硕士研究生, 主要从事基于 FPGA 的算法加速方向的研究。

引用格式: 王鑫, 刘炫. 基于 FPGA 的分子动力学模拟匹配单元设计[J]. 计算机测量与控制, 2022, 30(6): 203-209.

力与长程力，并对两者分别设计专用的低复杂度计算方法。具体来说，对固定的一个粒子，可以定义一个截止半径。这个粒子与范围内的其它粒子之间的非键合力就称为短程力。如何有效地从大量粒子中匹配出满足短程力计算要求的粒子对，是本文要解决的问题。

为了提升分子动力学模拟中粒子间非键合力的计算效率，本文提出了匹配单元的设计方案，用以高效地筛选出满足短程力计算条件的粒子对。

### 1 系统结构及原理

图 1 是本文研究的短程力计算系统的总体框图。首先，上位机通过 PCIe (peripheral component interconnect express) 总线将二进制的粒子数据传递到 FPGA 加速卡上的 BRAM (Block RAM)。当数据预处理单元接收到开始读取数据的指令时，就从 BRAM 中读取数据。读取完成后，数据预处理单元将读取的数据对应到各个粒子，再将粒子数据送入匹配单元进行匹配，匹配成功的粒子对进入到短程力计算单元。最后，计算得到的力将在力累加器中进行累加，并通过力回写单元将累加结果写回 BRAM，供上位机取回。

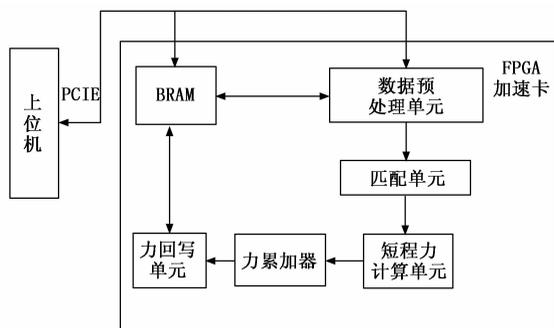


图 1 短程力计算系统总体框图

综上所述，短程力计算单元是短程力计算系统的核心部分。为了保证短程力计算单元的利用率，就需要从预处理后的粒子数据中准确快速地匹配出符合短程力计算条件的粒子对。针对该问题，本文设计了匹配单元。

本文设计的匹配单元是根据截止半径来筛选粒子对的。在半径为  $r_c$  的圆球内，以球心处粒子为参考粒子，匹配单元将滤除圆球外的粒子。与此同时，将圆内的粒子依次和参考粒子配对送入短程力计算单元。如果匹配单元每次只对一个参考粒子进行匹配，匹配效率不高。若多个参考粒子和周围粒子同时进行匹配，就能大幅度提升匹配效率。此外，分子动力学模拟是通过离散时间步长来模拟化学系统中单个粒子的运动<sup>[21]</sup>，而粒子在化学系统中的空间分布具有不确定性，因此，参考粒子的选取方式极大程度上影响着匹配单元的工作效率。

为保证匹配单元的工作效率，避免参考粒子的重复选取，需要对分子系统进行区域划分。因为化学系统是在一个(非物理)周期性边界条件盒子中的，所以，实际模拟

的系统相当于将一个无限的空间在三维坐标轴上平铺为大小相同的盒子。粒子就分布在这些盒子内。盒子的大小和粒子大小有关，每个盒子都有对应的盒子编号。除此之外，每个盒子内的粒子还有相应的盒内编号，并且每个粒子都有对应的位置坐标。为了便于遍历计算粒子对之间的相互作用，将其中的某个粒子作为参考粒子，称为静态粒子。静态粒子周围的粒子则称为动态粒子。由于化学系统内的粒子是随机分布的，故盒内粒子数具有不确定性。

在设计中，本文每个盒子最多可容纳 64 个粒子。为了补足盒内粒子数，使动态粒子必有静态粒子与其进行相互作用，本文又将静态粒子分为有效静态粒子和无效静态粒子(即空粒子)，空粒子是用来补足盒内粒子数。与此同时，为了方便判断静态粒子是不是空粒子，本文引入了空粒子标志位。

当对化学系统进行盒子划分后，结合 FPGA 的 DSP (digital signal processor) 资源的限制，匹配单元的匹配原则就不单单是在截止半径范围内，就需要更细致的匹配原则。匹配原则的内容主要有以下两个方面：

1) 偏序法：偏序法主要是为了防止同一盒子内的两个粒子重复计算短程力。具体而言，如果两个粒子在同一个盒子内，若将其中一个粒子作为参考粒子，参考粒子同周围粒子进行短程力的计算，同时盒内每个粒子都将轮流作为参考粒子，此时两个粒子之间就进行了两次短程力的计算。根据牛顿第三运动定律，两次计算的短程力大小相等，完全不需要计算两次。因此，参考粒子只需和盒内编号比自己大的粒子进行短程力的计算，盒内编号比自己小的已经计算过了，不需要再次计算。

2) 平面法：传统的直接计算方法<sup>[21]</sup>是使用 8 个 DSP 单元来计算浮点数  $r^2$ ，并与  $r_c^2$  进行比较。如果距离在截止范围之内，则  $r^2$  的值将会被短程力计算重复使用。基于 FPGA 的运算需要使用到 6 个 DSP 单元，如果有 8 个距离计算单元，则直接实现会消耗 48 个 DSP 单元，这样会限制每个芯片的流水线数量，此时可以使用平面法来解决这个问题。

平面法的准则可见式 (1) ~ (7)，其中， $X_1$ 、 $Y_1$ 、 $Z_1$ 、 $X_2$ 、 $Y_2$ 、 $Z_2$  分别是粒子对在三维坐标系中所对应的 X 轴、Y 轴和 Z 轴方向的坐标值，截止半径  $r_c$  是常数项，公式右边的数值都可以将其视作一个新的常量，这种方式能够避免产生更消耗资源的乘法的出现。平面法将两个粒子之间距离的高阶的比较运算拆分为简单的低阶比较运算。首先，两个粒子 X 轴、Y 轴和 Z 轴方向的距离要在截止半径  $r_c$  范围内；其次任意两个坐标轴方向差值的和要小于临界值  $\sqrt{2}r_c$ ，式 (7) 也是类似。

$$|X_1 - X_2| < r_c \tag{1}$$

$$|Y_1 - Y_2| < r_c \tag{2}$$

$$|Z_1 - Z_2| < r_c \tag{3}$$

$$|X_1 - X_2| + |Y_1 - Y_2| < \sqrt{2}r_c \tag{4}$$

$$|X_1 - X_2| + |Z_1 - Z_2| < \sqrt{2}r_c \quad (5)$$

$$|Z_1 - Z_2| + |Y_1 - Y_2| < \sqrt{2}r_c \quad (6)$$

$$|X_1 - X_2| + |Y_1 - Y_2| + |Z_1 - Z_2| < \sqrt{3}r_c \quad (7)$$

## 2 硬件设计

匹配单元的设计如图 2 所示。图中的匹配单元有 8 个距离计算器, 可实现 1 个动态粒子分别和 8 个静态粒子进行配对。首先将盒子中的静态粒子和动态粒子送入 FIFO (first input first output) 进行暂存, 当匹配单元接收到数据预处理完成的信号后, FIFO 中静态粒子和动态粒子进入流寄存器中, 以粒子对形式暂存, 共计 8 组粒子对。一旦粒子对发送到距离计算器, 就将粒子之间的距离与截止半径进行比较, 最后输出粒子对匹配结果。距离计算器的匹配结果存放在向量寄存器 (向量寄存器本质上是多个寄存器进行捆绑) 中。由于可能存在空粒子, 所以, 需要排除空粒子与动态粒子生成的粒子对。于是, 通过将距离计算器的匹配结果和流寄存器中的空粒子标志位进行与运算, 经过与运算后的 8 组粒子对流向仲裁器, 最后输出满足短程力计算条件的粒子对。

值得注意的是, 数据预处理单元的工作速度也限制匹配单元的工作效率。由于数据预处理的速度是未知的, 因此需要一个 FIFO, 通过缓存数据来控制进入匹配单元的粒子。除此之外, 粒子对在进行计算的时候需要的时间比较长, 为了节约时间, 将短程力计算单元设计成流水的, 也就是说第一组粒子对进入计算单元计算后, 第二组能马上跟上。在本文系统中, 短程力计算单元只有一个输入口。由于匹配单元匹配成功的粒子对个数是大于或等于 1 的, 所以, 需要一个仲裁器对匹配单元输出粒子对进行仲裁, 使匹配成功的粒子对流水输出到短程力计算单元。

匹配单元的设计, 重点在于粒子对的生成、过滤和仲裁。

### 2.1 粒子对生成

为了使进入匹配单元的粒子能够高效有序配对, 本文系统的匹配单元设计了粒子对生成, 如图 2 所示粒子对生成主要包括 FIFO 和流寄存器部分。匹配单元每轮将有 8 组粒子对进行匹配。当本轮匹配结束后, 就需要新一轮的粒子进入到匹配单元, 此时, 需要通过 FIFO 使暂存的粒子进入到匹配单元。此外, 进入匹配单元的粒子将开始进行匹配, 但数据预处理可能依旧有粒子输出, 因此, 需要 FIFO 将它们暂存, 等下一轮再输出。FIFO 中输出的粒子在进入流寄存器中时, 为使静态粒子和动态粒子进行高效合理地配对, 同时满足时序要求, 本文使用流寄存器, 将静态粒子和动态粒子以粒子对的形式暂存起来。

### 2.2 粒子对过滤

根据本文第 1 节, 实际模拟的系统相当于将一个无限的空间在三维坐标轴上平铺为大小相同的盒子, 图 3 是粒子 P 的模拟空间, 其中以粒子 P 所在的盒子为主盒, 主盒的邻域以非灰色表示, 盒子边缘大小为截止半径  $r_c$ , 则三

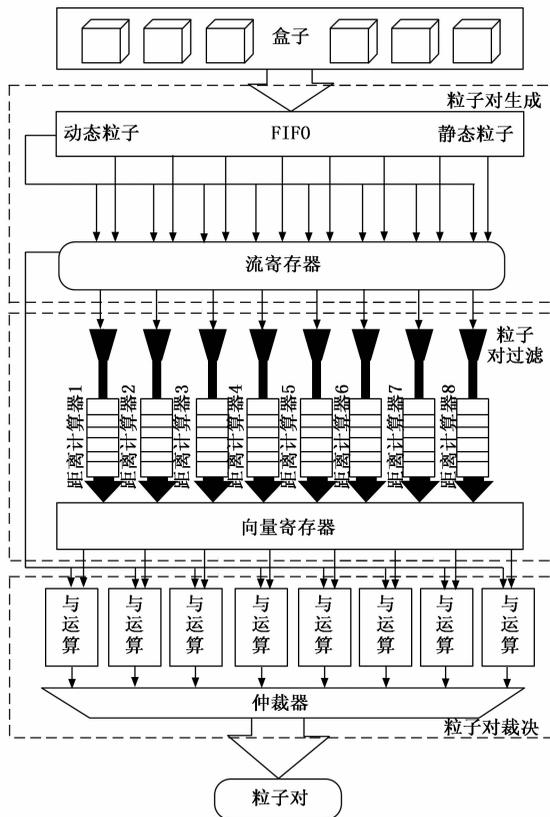


图 2 匹配单元设计框图

维坐标系中可能与粒子 P 进行匹配的其他粒子在相邻的 27 个盒子内。

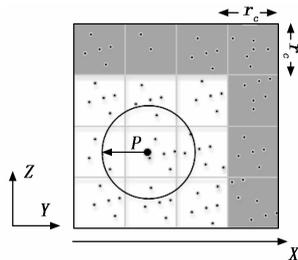


图 3 粒子 P 的模拟空间

平均配对成功率见式 (8):

$$Average_{PassRate} = \frac{3}{4} \frac{\pi r_c^3}{27 \times r_c^3} = 15.5\% \quad (8)$$

从式 (8) 可以看出, 平均配对成功率不高, 只有 15.5%。为了尽可能提高短程力计算的利用率 (尽量达到不停计算, 而不产生空转), 使每次短程力计算至少有一个有效输入就需要 7 个以上距离计算器, 理论上这就可以达到使短程力计算流水线充分运转。所以, 本文系统的匹配单元使用了 8 个距离计算器。

如图 2 所示, 粒子对过滤主要包括距离计算器和向量寄存器部分。当流寄存器中的粒子对在进入到 8 个距离计算器后, 将根据第 1 节的匹配原则对粒子对进行匹配。

图 4 的状态转移图详细介绍了距离计算器的工作过程。当距离计算器收到粒子对信息后，首先对盒子号和盒内编号进行比较，其次计算粒子间距离，最后输出匹配成功的粒子对。如果静态粒子和动态粒子不在同一个盒子内，就根据式 (1) ~ (7) 对粒子间距离进行计算。如果静态粒子和动态粒子在同一个盒子内，比较静态粒子和动态粒子的盒内编号，如果动态粒子盒内编号大于静态粒子盒内编号，则进行粒子间距离的计算，反之则不需要进行粒子间距离的计算。如果粒子间距离都满足式 (1) ~ (7)，距离计算器的匹配结果为 1，反之为 0。

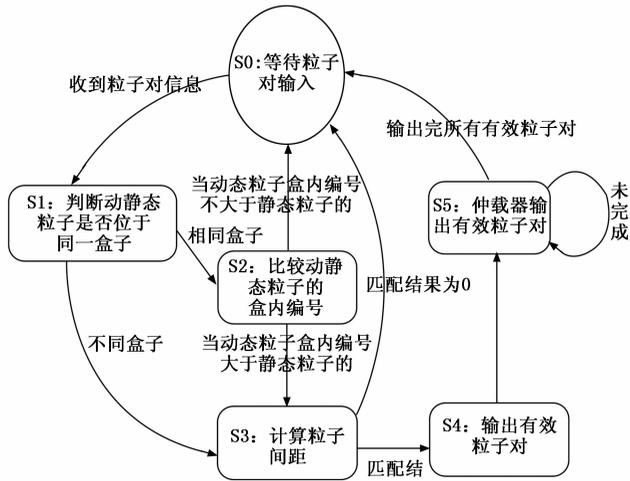


图 4 距离计算器状态图

### 2.3 粒子对仲裁

由于本文系统的短程力计算单元只有一个输入端口，并且为了排除空粒子，本文系统的匹配单元设计了粒子对仲裁，粒子对仲裁主要包括与运算和仲裁器部分，如图 2 所示。当距离计算器的匹配结果输出时，由于存在空粒子，故还需要对图 2 中暂存在向量寄存器中的 8 组距离计算器匹配结果和流寄存器中的空粒子标志位进行与运算，避免空粒子进入短程力计算单元。

由于 8 组与运算结果同时输出，并且 8 组结果中含有无效粒子对，此时，需要流水输出有效粒子对，故本文系统的匹配单元使用了仲裁器，即可以判断粒子对的有效性又可以流水输出。同时考虑到仲裁器有多个输入端口和一个输出端口，并且每个有效的粒子对请求输出的机会是平等的，在这种情况下，存在粒子对输出先后问题。为了在保证公平的基础上解决这个问题，采用了仲裁器的轮询仲裁策略。通过轮询仲裁依次输出多个匹配成功的粒子对，为短程力计算单元实现流水供数。下面对轮询仲裁进行详细介绍。

轮询仲裁策略<sup>[22]</sup>的基本思路是，当请求者发出一个请求并获得许可后，它的优先级在接下来的仲裁中就变成了最低。简而言之，就是每个请求者的优先级不是固定不变的，而是在优先级达到最高之后（即获得许可后）变成最低，并且还会根据其他请求者获得许可的情况进行相应的

调整。这样当有多个请求者时，每个请求者将依次获得许可，即便之前高优先级的请求者重新发出新的请求，也会等前面的请求者都获得许可后再轮到它。关于轮询仲裁策略的硬件实现，就必须提到固定优先级仲裁策略，固定优先级，顾名思义，就是每个模块的优先级是固定的，提前分配好的，如果有两个模块同时发出请求，则优先级高的模块可以获得许可。固定优先级的硬件设计是通过输入一个多位的请求信号，其中每一位代表一个模块的请求，输出也是多位的许可信号，每一位代表对应模块的许可信号。默认最低位优先级最高，最高位优先级最低。从而实现固定优先级仲裁。轮询仲裁的硬件设计是在固定优先级仲裁的基础上进行的，轮询仲裁硬件设计需要两个并行的固定优先级仲裁，还需要一个屏蔽信号。屏蔽信号原理是将一个多位的请求信号之前获得许可的位以及之前的位都屏蔽掉，允许通过的是之前优先级更低的位。如果那些位上之前有发出请求但没获得许可，在经过屏蔽信号后就可以轮到它们。每次获得许可之后，屏蔽信号里 0 的位数会变多，从而屏蔽掉更多位，直到所有低优先级发出的请求均都获得过一次许可后，请求信号和屏蔽信号逻辑与的结果变成全 0，此时表明轮询结束，需要进行新一轮的轮询。图 5 是轮询仲裁的硬件设计图，屏蔽信号和请求信号在逻辑与的结果不全为 0 时，即存在没被屏蔽掉的请求，此时优先获得许可。当屏蔽信号和请求信号逻辑与结果全为 0 时，即没有请求需要屏蔽。此时需要将逻辑与的结果和请求信号进行逻辑与。最后将逻辑与的结果和之前的结果进行逻辑或，最后输出的即获得许可。

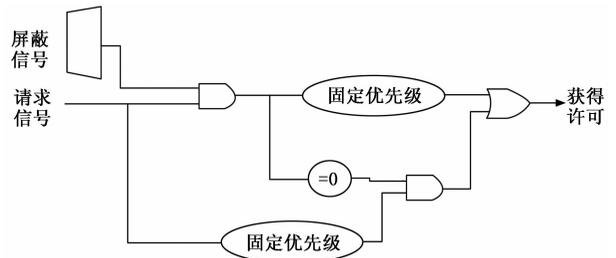


图 5 轮询仲裁硬件设计图

### 3 软件设计

系统实现过程中，一方面需要不断将硬件实现的结果和软件的结果进行对比，以确保硬件设计实现的正确性。另一方面，需要不断调整优化算法，实现对硬件资源的高效利用。系统的软件设计是基于 C 语言实现，系统的开发流程包括以下几点：

- 1) 获取软件代码和输入数据，生成最终的输出数据（标准结果，作为参照）；
- 2) 根据硬件顶层设计，重新划分模块，每个模块都有软硬件两种实现；
- 3) 运行软件模块获得软件结果文件，软件结果文件经过对比后，可以作为回归测试的标准结果文件使用；
- 4) 运行硬件模块获得硬件结果文件；

5) 对软硬件结果文件进行比对。

软件主要使用浮点数据作为输入输出, 同时软硬件输入数据, 均由软件统一提供。匹配单元软件设计的主要思路是判断两个粒子之间的距离是否在截止半径范围内。根据第 1 节引入的偏序法和平面法, 实现对粒子对的筛选, 将不满足短程力计算的粒子对过滤, 其软件流程如图 6 所示。

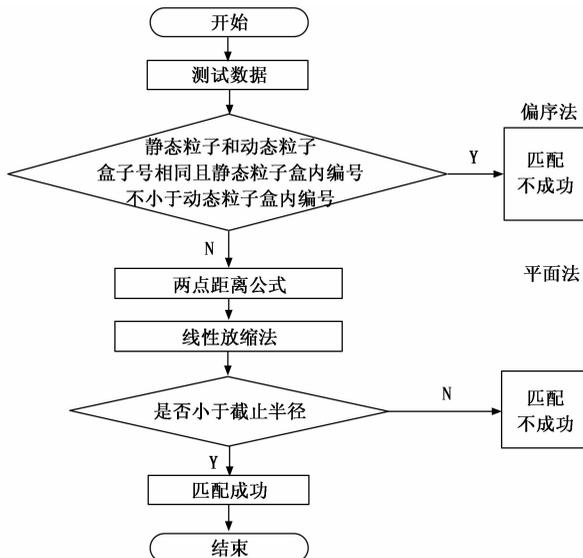


图 6 软件流程图

如图 6 所示, 首先通过偏序法判断静态粒子和动态粒子的盒子号和盒内编号来排除粒子对重复计算的可能性。如果静态粒子和动态粒子在同一盒子内, 并且静态粒子的盒内编号大于等于动态粒子的盒内编号, 则直接排除掉该粒子对。当且仅当静态粒子和动态粒子不在同一盒子内或者静态粒子的盒内编号小于动态粒子的盒内编号, 才可以进一步进行筛选。下一步通过平面法将满足偏序法的粒子对进行距离比较, 根据线性放缩法, 将两点距离公式从高阶降为低阶, 使粒子到球面的距离简化为到平面的距离。在对粒子进行距离比较时, 根据式 (1) ~ (7), 将公式左边数据进行运算处理, 此过程中不涉及高阶运算, 全部运算都停留在低阶, 通过使用 c 语言库中自带函数, 使整个运算过程简单易处理, 相比于硬件这部分运算内容更易实现, 运算结束再进行距离比较得到同时满足偏序法和平面法的粒子对组合。最后, 匹配成功的粒子对即满足短程力计算的粒子对。据此, 实现对粒子对的过滤。

## 4 实验结果与分析

### 4.1 软件测试

测试使用的粒子数据集 (7 051 个有效粒子数据) 是来自蛋白质晶体结构资料数据库 (PDB, protein data bank) 的多聚泛素酶基因 (UBQ) 数据集 (<https://www.rcsb.org/structure/1UBQ>)。由于数据过多只说明部分数据, 表 1 列出了该数据集中的部分粒子数据。输入的数据主要是 0 号盒子和 5 号盒子内的粒子。前期工作准备完毕后, 就根

据第 3 节的匹配原则对粒子进行匹配。其中截止半径为 12Å。通过软件编译得到的匹配结果如表 2 所示。

表 1 粒子数据

粒子编号	粒子属性	盒子号	盒内编号	X 轴坐标	Y 轴坐标	Z 轴坐标
1	动态粒子	0	1	-17.541 99	-19.848 99	-20.070 99
2	动态粒子	5	1	-16.055 99	-15.778 99	-11.898 99
3	静态粒子	0	0	-17.239 99	-19.672 99	-19.172 99
4	静态粒子	0	1	-17.541 99	-19.848 99	-20.070 99
5	静态粒子	0	7	-15.418 99	-19.500 99	-22.366 99

表 2 匹配结果

粒子编号	粒子编号	匹配结果
1	3	1
1	4	0
1	5	0
2	3	1

### 4.2 硬件测试

本文使用 SpinalHDL 实现前述匹配单元的硬件设计, SpinalHDL 是一种开源的高级硬件描述语言, 是基于 Scala 的全新硬件设计语言, 更准确的说是一个基于 scala 的 HDL 开发框架, 不同于以前的 HDL, 它解决了 Verilog 的例化不方便, 大量重复声明, 函数不能带参, 参数化笨拙, 错误检测弱等问题。并且和传统的集成电路流和谐共存, 适合大规模的系统级芯片 (Soc, system on chip) 的开发。在系统通过 SpinalHDL 实现硬件设计之后, 还需对其进行板级测试验证, 验证所设计的逻辑代码是否符合预期的要求, 测试结果是否与软件仿真结果一致, 是否满足时序要求。在现在的集成电路的验证中, 主要有两种方法, 一种是随机验证, 另外一种形式验证 (formal verification)。随机验证就是通过大量的随机向量来验证测试是否符合要求 (通常是将硬件的结果和软件的结果进行比较, 或者通过断言比较)。形式验证的主要思想是通过使用形式证明的方式来验证一个设计的功能是否正确。形式验证可以分为 3 大类: 等价性检查 (equivalence checking)、模型检查 (model checking) 和定理证明 (theory prover)。本文所使用的方法是基于 FPGA 开发平台的随机验证。基本方法是: 在 FPGA 中调用生成的设计模块 (以 IP 形式), 添加一些外围逻辑, 来给此设计模块提供测试输入; 通过主机发送控制命令, 对设计模块进行测试; 回收测试结果, 将测试结果与预期的软件仿真结果比较。使用的 FPGA 开发平台是 Xilinx 公司的 VCU128 开发套件。VCU128 开发板采用全新 Xilinx VU37P HBM FPGA, 利用堆叠芯片互连将 8 GB HBM DRAM 裸片添加到封装基板上的 FPGA 裸片旁边。支持高带宽存储器 (HBM) 的 Xilinx FPGA 是计算带宽问题 (与在 PCB 上使用 DDR4 等并行内存相关) 的最优解决方案。VCU128 提供 PCIe Gen3 x16 接口, 插到一台服务器主机上。主机平台安装 Ubuntu18.04 Linux 操作系统, 在

其上运行驱动程序和测试程序。测试框架如图 7 所示。

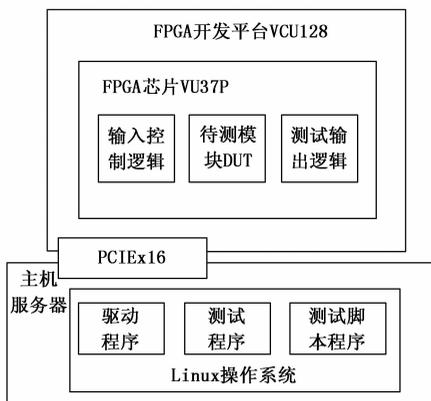


图 7 验证平台框架示意图

图 7 中，测试脚本程序调用测试程序，测试程序使用驱动程序中的读写函数，跟 VCU128 开发平台上的 FPGA 芯片通讯。FPGA 芯片接收命令，输入控制逻辑从存储器指定位置读取测试输入数据，提供给待测模块进行测试，测试结果通过测试输出逻辑写到指定存储器位置。测试程序从指定存储器读取结果数据，跟期望值比较，输出比较结果。

因数据量较大，本节主要说明基于部分数据的硬件测试结果。为了便于计算，将表 1 中的浮点数据都转换成了 16 进制的定点数（22 位整数，42 位小数）。图 8 是 0 号盒子内粒子的匹配结果，从图的最左边可以看到匹配单元的相关信息。其中重要信息有 static\_vec（静态粒子向量）、is\_real（空粒子标志）、dynamic（动态粒子）、number（盒子号）、idx（盒内编号）、loc\_x（粒子 x 轴坐标）、loc\_y（粒子 y 轴坐标）、loc\_z（粒子 z 轴坐标）、match\_cal\_io\_count（距离计算器匹配结果）、pairs\_out（最后输出粒子对）、pairs\_out\_valid（输出粒子对有效标志）。其中静态粒子和动态粒子都在 0 号盒子中，且动态粒子盒内编号是 1。因为静态粒子个数比较多，所以只展示盒内编号为 0 的粒子。从图中可以看到距离计算器只有一对粒子匹配成功（0 表示匹配不成功，1 表示匹配成功），最后经过仲裁器输出的粒子对也仅有一对。

图 9 是 8 组粒子对都匹配成功情况。可以看出静态粒子在 0 号盒子中，动态粒子在 5 号盒子中，由于共有 8 个静态粒子，导致粒子数据信息比较多，故只展示了第 1 个静态粒子的数据信息，经过距离计算器进行匹配，发现距离计算器的结果均为 1，即所有粒子对都匹配成功，最后匹配成功的粒子对全部通过仲裁器一一输出。

结果表明，图 8、图 9 硬件测试结果与表 2 的理论匹配结果完全一致。

与此同时，为了说明偏序法和平面法可以节省系统的资源消耗，对使用偏序法和平面法的情况以及只使用直接计算方法的情况分别进行了测试验证。其中使用直接计算方法和使用偏序法、平面法的资源对比如表 3。可以看出，

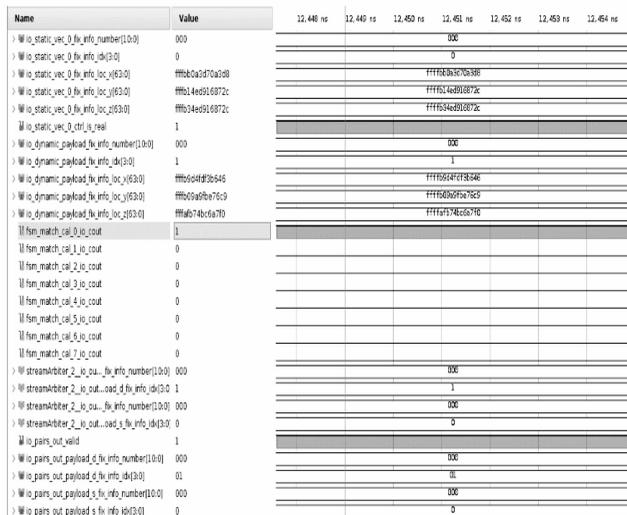


图 8 0 号盒子粒子匹配仿真波形

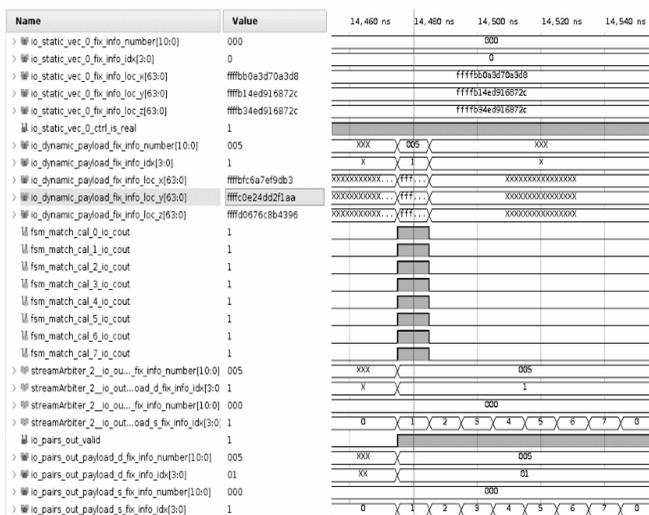


图 9 不同盒子粒子匹配仿真波形

使用偏序法、平面法比使用直接计算方法要多消耗 7.6% 的 LUT (Look-Up-Table)，但却节省了 70% 的 DSP 资源。相比于 LUT 资源，DSP 资源对短程力计算系统更为珍贵。

表 3 不同情况下粒子匹配资源消耗

并行策略	BRAM	DSP	LUT
直接计算方法	12(1%)	57(1%)	6520(0.7%)
偏序法和平面法	11(1%)	17(0.3%)	7030(0.8%)

## 5 结束语

本文针对提升分子动力学模拟中粒子间非键合力的计算效率的问题，提出了盒子划分、静态粒子、动态粒子和空粒子标志位等概念。通过将盒子内的静态粒子和动态粒子进行配对生成粒子对，有效地提高了匹配效率。此外，还提出了粒子对筛选的两种方法：偏序法和平面法，使用偏序法和平面法节省了系统 70% 的 DSP 资源。在分子动力

学模拟中, 短程力计算单元是计算量最大的部分, 而匹配单元过滤掉不需要计算的粒子对减少了整个系统的计算量。从测试结果可知, 匹配单元有效地过滤掉了粒子间距离较大的粒子对, 输出了与理论分析一致的结果。

目前本文系统的匹配单元最高运行频率达到 115 MHz, 下一步可继续优化匹配单元, 如减少组合逻辑内容, 使资源消耗中 LUT 数量减少。LUT 数量的减少有助于提升工作频率, 这对于进一步提升短程力计算系统整体性能具有重要意义。

#### 参考文献:

- [1] DROR R O, DIRKS R M, GROSSMAN J P, et al. Biomolecular simulation: a computational microscope for molecular biology [J]. *Annual review of Biophysics*, 2012, 41: 429 - 452.
- [2] Páll S, ZHUMUROV A, BAUER P, et al. Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS [J]. *The Journal of Chemical Physics*, 2020, 153 (13): 134110.
- [3] DI NATALE F, BHATIA H, CARPENTER T S, et al. A massively parallel infrastructure for adaptive multiscale simulations: modeling RAS initiation pathway for cancer [C] // *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019: 1 - 16.
- [4] YOSHII N, ANDOH Y, FUJIMOTO K, et al. MODYLAS: A highly parallelized general-purpose molecular dynamics simulation program [J]. *International Journal of Quantum Chemistry*, 2015, 115 (5): 342 - 348.
- [5] JAIN N, BOHM E, MIKIDA E, et al. Openatom: Scalable ab-initio molecular dynamics with diverse capabilities [C] // *International Conference on High Performance Computing*. Springer, Cham, 2016: 139 - 158.
- [6] ABRAHAM M J, MURTOLA T, SCHULZ R, et al. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers [J]. *SoftwareX*, 2015, 1: 19 - 25.
- [7] PANG Y T, MIAO Y, WANG Y, et al. Gaussian accelerated molecular dynamics in NAMD [J]. *Journal of Chemical Theory and Computation*, 2017, 13 (1): 9 - 19.
- [8] WANG B, CHEN Y, HOU C. Communication optimization strategy for molecular dynamics simulation on sunway taihu-light [C] // *2020 IEEE 22nd International Conference on High Performance Computing and Communications, IEEE 18th International Conference on Smart City, IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, 2020: 571 - 578.
- [9] JUNG J, NAURSE A, KOBAYASHI C, et al. Graphics processing unit acceleration and parallelization of GENESIS for large-scale molecular dynamics simulations [J]. *Journal of Chemical Theory and Computation*, 2016, 12 (10): 4947 - 4958.
- [10] JUNG J, NISHIMA W, DANIELS M, et al. Scaling molecular dynamics beyond 100, 000 processor cores for large-scale biophysical simulations [J]. *Journal of Computational Chemistry*, 2019, 40 (21): 1919 - 1930.
- [11] HOLLINGSWORTH S A, DROR R O. Molecular dynamics simulation for all [J]. *Neuron*, 2018, 99 (6): 1129 - 1143.
- [12] SIMMONETT A C, BROOKS B R. Analytical Hessians for Ewald and particle mesh Ewald electrostatics [J]. *The Journal of Chemical Physics*, 2021, 154 (10): 104101.
- [13] LÜGHAUSEN F, FAMAHEY B, KROUPA P. Phantom of RAMSES (POR): a new Milgromian dynamics N-body code [J]. *Canadian Journal of Physics*, 2015, 93 (2): 232 - 241.
- [14] KURODA Y, SUENAGA A, SATO Y, et al. All-atom molecular dynamics analysis of multi-peptide systems reproduces peptide solubility in line with experimental observations [J]. *Scientific Reports*, 2016, 6 (1): 1 - 11.
- [15] MORIMOTO G, KOYAMA Y M, ZHANG H, et al. Hardware acceleration of tensor-structured multilevel ewald summation method on MDGRAPE-4A, a special-purpose computer system for molecular dynamics simulations [C] // *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021: 1 - 15.
- [16] BRIONES R, BLAU C, KUTZNER C, et al. GROMaps: A GROMACS-based toolset to analyze density maps derived from molecular dynamics simulations [J]. *Biophysical Journal*, 2019, 116 (1): 4 - 11.
- [17] CONG J, FANG Z, KIANINEJAD H, et al. Revisiting FPGA acceleration of molecular dynamics simulation with dynamic data flow behavior in high-level synthesis [J]. *arXiv preprint arXiv: 1611.04474*, 2016.
- [18] ADAMS P J, BATSON B, BELL A, et al. The ANTON 3 ASIC: a fire-breathing monster for molecular dynamics simulations [C] // *2021 IEEE Hot Chips 33 Symposium (HCS)*, IEEE, 2021: 1 - 22.
- [19] SHAW D E, ADAMS P J, AZARIA A, et al. Anton 3: twenty microseconds of molecular dynamics simulation before lunch [C] // *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2021: 1 - 11.
- [20] LARSON R H, SALMON J K, DROR R O, et al. High-throughput pairwise point interactions in Anton, a specialized machine for molecular dynamics simulation [C] // *2008 IEEE 14th International Symposium on High Performance Computer Architecture*. IEEE, 2008: 331 - 342.
- [21] YANG C, GENG T, WANG T, et al. Fully integrated FPGA molecular dynamics simulations [C] // *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2019: 1 - 31.
- [22] 蔡 升. 一种二维片上网络路由器的设计实现 [J]. *计算机测量与控制*, 2019, 27 (9): 209 - 212.