第 43 卷 第 2 期/2023 年 1 月/光学学报 研究论文

光学学报

# 基于FPGA异构计算的数据协调系统设计

刘佳森[1]，郭大波[1,2*]，郭天昊[1]，李仙钟[1]，王玉杰[1]，孟颖岫[1]

[1]山西大学物理电子工程学院，山西 太原 030006；

[2]三亚学院信息与智能工程学院，海南 三亚 572022

**摘要** 针对当前连续变量量子密钥分发系统数据协调运算速度慢的问题，采用高性能FPGA板为加速设备，在OpenCL异构计算框架上实现了八维数据协调算法的并行加速运算。针对FPGA的特点，所提算法进行了如下优化：1)优化for循环表达方式，使OpenCL编译器能更好地理解设计意图，以生成有效的FPGA硬件结构，速度提高50%以上；2)内存优化，根据LDPC解码置信传播算法的特点，设计了一种哑铃式内核架构和核内、核间信息传播方式，速度提高了近1倍；3)使用聚合访问的数据读取模式减少并行工作项数量，速度提高了1倍多。仿真结果显示，在码长为$2\times10^5$ bit的情况下，代码优化后的协调速率为优化前的2.17倍，采用OpenCL/FPGA异构平台并行加速的协调速率是单一CPU平台的4倍以上。

**关键词** 量子光学；量子密钥分发；多维数据协调；OpenCL；FPGA；LDPC码

**中图分类号** O431.2 **文献标志码** A **DOI**：10.3788/AOS0227001

## 1 引 言

随着通信技术的快速发展，保密通信在信息安全特别是军事通信安全方面显得至关重要。量子密钥分发(QKD)是量子通信领域最早可实用化的一项技术，具有理论上的绝对安全性[1]。自1984年Bennett等提出BB84协议[2]以来，现在已经发展出许多新的QKD协议[3]。按照信源端编码空间的维度可以分为离散变量QKD(DV-QKD)和连续变量QKD(CV-QKD)。

由于通信双方最终建立的密钥必须是一致的，为了排除各种因素造成的误码，需要通过一个环节从双方相关的序列中提取出一致的信息[4]，称为数据协调。DV-QKD的数据协调相对简单，但DV量子态的产生和探测装置却非常复杂；相比之下，CV量子态的调制和解码不需要专门的设备，可由当前广泛使用的标准电信网络实现，且常温下CV-QKD使用的零差探测器[5]的效率要高于DV-QKD使用的单光子探测器[6]。研究者们发现，CV-QKD的数据协调环节是CV-QKD系统远程化和高速化发展的瓶颈[7]，原因在于CV-QKD的传输信道是高斯信道而非二进制纠错信道(如BEC、BSC及BI-AWGNC)，渐近收敛信噪比较DV-QKD高，从而缩短了这一系统的安全密钥传输距离。根据香农信道编码定理，编码码组越长，信道的渐近收敛信噪比越低。CV-QKD系统普遍采用大码长，参与数据协调的光脉冲数据达$10^5$量级，明显增加了系统的计算量，导致通信时延变长。基于FPGA[8]和芯片[9]硬件加速的CV-QKD技术有望解决这一关键技术问题，因此得到快速发展。

针对CV-QKD的数据协调主要包括3种协议：切片纠错(SEC)协议[10]、使用连续变量的符号进行编码的符号协调算法[11]、由Leverrier等[12]提出的多维数据协调算法。切片纠错协议破坏了高斯对称性[13]，因此收敛信噪比较高，传输距离被限制在25 km[14]。使用符号协调算法时，由于高斯调制的大部分随机数分布在0附近，在信噪比较低时难以区分信号。多维数据协调算法保留了高斯对称性，并通过旋转操作使得各个状态点之间的距离最大化，这就使得信噪比较低时也可以协调成功，从而较大地提升了安全传输距离。

Lin等[15]利用CUDA(compute unified device architecture)并行语言在GPU上加速数据协调，协调速率可达到25 Mbit/s，但他们的分组码长只有$10^4$ bit，因此数据协调效率低，密钥分发距离短。从目前的研究结果来看，初始密钥码字长度达到$2\times10^5$ bit左右为最佳[16]，这导致系统的计算量大，解码时间长。

基于上述问题，本文采用OpenCL(open computing language)异构框架来加速数据协调，选择具有高度并行性的FPGA加速板来执行繁重的解码任务，在根据FPGA的特点对OpenCL代码进行优化

**收稿日期**：2022-06-09；**修回日期**：2022-07-03；**录用日期**：2022-07-21；**网络首发日期**：2022-07-30

**基金项目**：山西省基础研究项目(201801D121118)、三亚学院人才引进项目(USYRC22-14)

**通信作者**：*dabo_guo@sxu.edu.cn

后，较大地提升了解码速度，超过了同样采用OpenCL框架的CPU/GPU异构平台。

## 2　CV-QKD多维数据协调方案

### 2.1　八维数据协调方案

由于高斯调制的随机变量集中在0附近，且大部分数据的绝对值都很小，当信噪比较低时，很难对信号进行有效区分。为解决这个问题，研究人员提出了多维数据协调方案，即将多个数据组合为向量，利用多维空间的旋转操作将这些向量投射到球面上并最大化分开。本文采用八维数据协调方案，过程如下：

1）Alice和Bob通过量子信道完成信息传输后，分别按顺序将每8个连续变量组合成1个8维向量，设Alice发送的8维向量为$\boldsymbol{X}$，Bob接收到的8维向量为$\boldsymbol{Y}$，$\boldsymbol{Z}$为信道噪声，则有

$$\boldsymbol{Y}=\boldsymbol{X}+\boldsymbol{Z},\boldsymbol{X}\sim N\left(0,\Sigma^2\right)^8,\boldsymbol{Z}\sim N\left(0,\sigma^2\right)^8,\quad(1)$$

式中：$\Sigma^2$为Alice端调制信号方差；$\sigma^2$为信道噪声方差。

2）对$\boldsymbol{X}$和$\boldsymbol{Y}$进行归一化处理，将向量由欧氏空间映射到单位球面空间，即

$$\begin{cases}\boldsymbol{x}=\boldsymbol{X}/\|\boldsymbol{X}\|\\ \boldsymbol{y}=\boldsymbol{Y}/\|\boldsymbol{Y}\|\end{cases},\quad(2)$$

式中：$\|\boldsymbol{X}\|=\sqrt{\langle\boldsymbol{X},\boldsymbol{X}\rangle}$；$\|\boldsymbol{Y}\|=\sqrt{\langle\boldsymbol{Y},\boldsymbol{Y}\rangle}$。归一化后的随机向量从非均匀的高斯分布转化为均匀的高斯分布，但是信号点的分布仍然较密集，这就给信号点分辨带来很大困难。

3）Bob产生一串随机二进制数$L$并按8位分为一组，在球面上建立如下映射关系

$$\{b_1,b_2,b_3,\cdots,b_8\}\to\left\{\frac{(-1)^{b_1}}{2\sqrt{2}},\frac{(-1)^{b_2}}{2\sqrt{2}},\frac{(-1)^{b_3}}{2\sqrt{2}},\cdots,\frac{(-1)^{b_8}}{2\sqrt{2}}\right\},\quad(3)$$

式中：$\{b_1,b_2,b_3,\cdots,b_8\}$为8位随机二进制数。

4）令$\boldsymbol{u}=\left\{\frac{(-1)^{b_1}}{2\sqrt{2}},\frac{(-1)^{b_2}}{2\sqrt{2}},\frac{(-1)^{b_3}}{2\sqrt{2}},\cdots,\frac{(-1)^{b_8}}{2\sqrt{2}}\right\}$，在Bob端计算旋转矩阵$\boldsymbol{M}(\boldsymbol{y},\boldsymbol{u})$，使其满足以下关系：

$$\boldsymbol{M}(\boldsymbol{y},\boldsymbol{u})\boldsymbol{y}=\boldsymbol{u}。\quad(4)$$

5）将$\boldsymbol{M}(\boldsymbol{y},\boldsymbol{u})$发给Alice，Alice利用旋转矩阵计算得到$\boldsymbol{v}$：

$$\boldsymbol{M}(\boldsymbol{y},\boldsymbol{u})\boldsymbol{x}=\boldsymbol{v},\quad(5)$$

由此完成旋转。$\boldsymbol{v}=\boldsymbol{u}+\boldsymbol{z}$，而噪声$\boldsymbol{z}$是$\boldsymbol{Z}$的旋转版本，两种噪声具有相同的方差。通过上述旋转，可避免高斯分布的大部分数据接近于0，从而出现不易分辨的问题。Alice在Bob发送的校验子$\boldsymbol{S}$的辅助下，使用SW（Slepian-Wolf）[17]解码器进行解码。该算法的整体方案如图1所示。

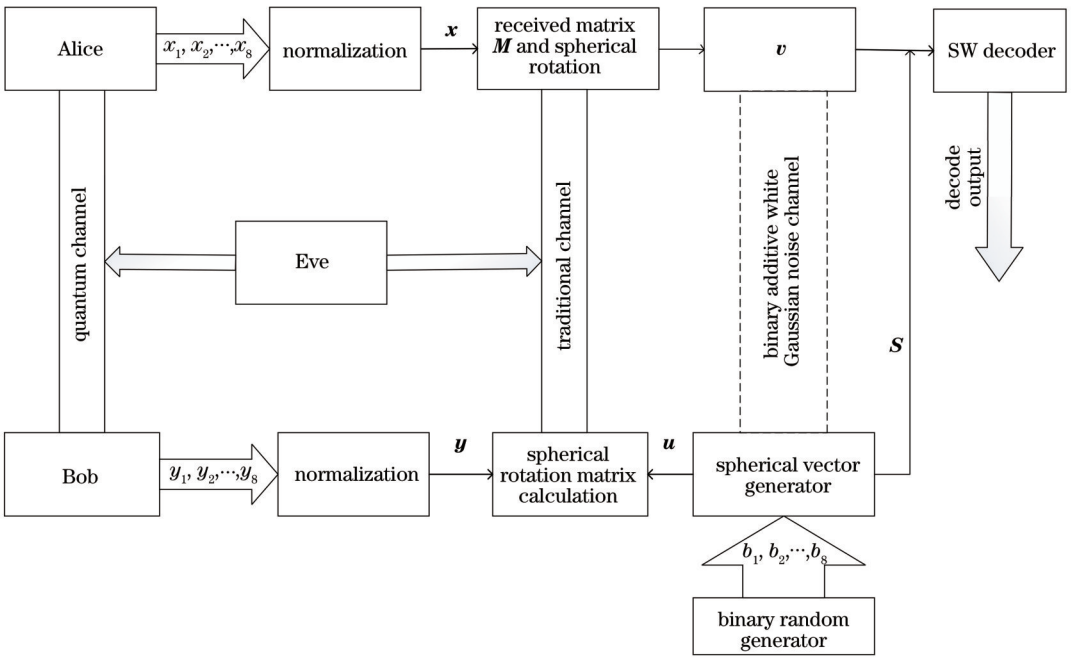


图1　八维数据协调算法示意图

Fig. 1　Schematic of eight-dimensional reconciliation

### 2.2　安全性分析

在CV-QKD中，Alice和Bob通过经典信道从双方相关的密钥信息中提取出一致的密钥，并保证泄漏给Eve的信息尽可能少。由于本文采取逆向协调，理论

上的安全密钥速率[18]为

$$K_{\mathrm{th}}=I_{\mathrm{A\text{-}B}}-\chi_{\mathrm{B\text{-}E}},\tag{6}$$

式中：$I_{\mathrm{A\text{-}B}}$为 Alice 发送序列和 Bob 接收序列之间的互信息量；$\chi_{\mathrm{B\text{-}E}}$为窃听者 Eve 窃听到的序列和 Bob 接收序列之间的 Holevo 信息量。由于实际协调过程中数据协调效率不可能达到 100%[19]，因此实际上的安全密钥速率为

$$K_{\mathrm{real}}=\beta I_{\mathrm{A\text{-}B}}-\chi_{\mathrm{B\text{-}E}},\tag{7}$$

$$\beta=R\Big/\left[\frac{1}{2}\log_2(1+R_{\mathrm{sn}})\right],\tag{8}$$

式中：$\beta$为数据协调效率；$R$为码率；$R_{\mathrm{sn}}$为收敛信噪比。

数据协调过程中，$K_{\mathrm{real}}>0$表示通信过程安全，这对协调效率有很高的要求。实验参数取 Alice 端信号功率 $V_{\mathrm{A}}=18.5$，信道输入的过量噪声 $\varepsilon=0.01$，信道透射率 $T=0.151$，Bob 端探测器的效率 $\eta=0.606$，由平衡探测器引入的电子噪声 $V_{\mathrm{elc}}=0.041$。系统脉冲重复频率为 500 kHz/s。本研究以真空散粒噪声 $N_0$ 为功率单位，上述功率均是以真空散粒噪声[20]功率为单位。利用上述参数可算得 $I_{\mathrm{A\text{-}B}}=243.79$ kbit/s，$\chi_{\mathrm{B\text{-}E}}=222.73$ kbit/s[16]。本实验的量子信道采用 1550 nm 光通信波段光纤，传输损耗为 0.2 dB/km，因此传输距离的计算公式为

$$D_{\mathrm{is}}=\frac{10-R_{\mathrm{sn}}}{0.2}。\tag{9}$$

## 3　基于二进制 LDPC 码的异构计算方案

整个数据协调过程中 Bob 端负责旋转矩阵的计算以及校验子 $\boldsymbol{S}$ 的计算，这些计算只进行一次，计算量较小，因此将这些计算部署在主机(CPU)上。Alice 需要计算初始概率并运行置信传播算法[21]进行解码：初始概率只计算一次，也可由 CPU 执行；解码算法包含大量的迭代运算，这些运算任务对于串行计算的 CPU 处理器来说较为繁重，耗时较长，因此将这个任务分配给 FPGA，利用其拥有的大量并行内核来执行运算。

### 3.1　Bob 端主机程序

旋转矩阵的计算公式为

$$\boldsymbol{M}(\boldsymbol{y},\boldsymbol{u})=\sum_{i=1}^{8}a_i(\boldsymbol{y},\boldsymbol{u})\boldsymbol{A}_i,\tag{10}$$

$$\left[a_1(\boldsymbol{y},\boldsymbol{u}),\cdots,a_8(\boldsymbol{y},\boldsymbol{u})\right]^{\mathrm{T}}=\left[\boldsymbol{A}_1\boldsymbol{y},\cdots,\boldsymbol{A}_8\boldsymbol{y}\right]^{-1}\boldsymbol{u},\tag{11}$$

式中：$\boldsymbol{A}_i$为由 Leverrier 等[12]构造的单位矩阵；$\left[a_1(\boldsymbol{y},\boldsymbol{u}),\cdots,a_8(\boldsymbol{y},\boldsymbol{u})\right]$为 $\boldsymbol{u}$ 在正交基$\left(\boldsymbol{A}_1\boldsymbol{y},\cdots,\boldsymbol{A}_8\boldsymbol{y}\right)$上的坐标。

校验子计算公式为

$$\boldsymbol{S}=\boldsymbol{H}\times\boldsymbol{L},\tag{12}$$

式中：$\boldsymbol{H}$为 LDPC(low density parity check code)码对应的校验矩阵；$\boldsymbol{L}$为 Bob 端产生的码字。

### 3.2　Alice 端 OpenCL 并行架构

Alice 执行 LDPC 置信传播解码运算，不断更新变量节点和校验节点之间的消息[21]，直到收敛。图 2 以 Tanner 图的形式展现了迭代中的信息流动。假设当前校验节点为 $i$，其连接的变量节点集为 $N(i)$，$j$ 为变量节点集中的一个节点，其连接的校验节点集为 $M(j)$。$P_{ij}$为校验节点 $i$ 向变量节点 $j$ 传递的信息，$Q_{ji}$为变量节点 $j$ 向校验节点 $i$ 发送的信息。$O_j$为变量节点 $j$ 的初始概率信息，$S_i$为校验节点 $i$ 接收到的检验子信息。图 2 中的箭头指明了节点之间信息的流动方向。为实现上述功能，FPGA 设备端包含 3 个内核函数，一个内核用来初始化信息，另外两个内核分别用于处理校验节点信息和变量节点信息。

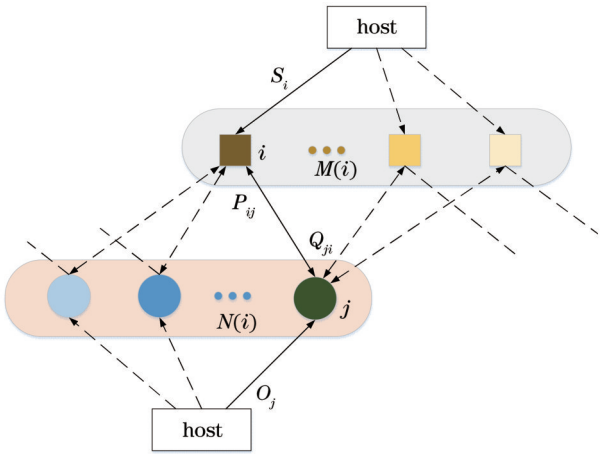


图 2　信息流动的 Tanner 图

Fig. 2　Tanner graph of information flow

#### 3.2.1　初始化内核

初始概率是要在向量 $\boldsymbol{v}$ 已知的情况下计算向量 $\boldsymbol{u}$ 中对应位的概率，这是 LDPC 译码的第一步，也是最关键的步骤，很大程度上影响了译码时所需要的迭代次数。林毅等[22]给出了 $\boldsymbol{u}$ 和 $\boldsymbol{v}$ 之间的计算关系，经过实验优化，初始概率的计算公式为

$$O_j=\exp\left\{-2\cdot\left\|\boldsymbol{X}_j\right\|\cdot v_j\right\},\tag{13}$$

式中：$\left\|\boldsymbol{X}_j\right\|$为变量 $j$ 对应的向量内积；$v_j$为变量 $j$ 对应的向量中该维度的值。初始化内核负责将计算好的 $O_j$ 值作为第一次迭代时所需要的 $Q_{ji}$ 值传给校验节点内核。

#### 3.2.2　校验节点内核

校验节点 $i$ 负责接收与其连接的所有变量节点传来的消息 $Q_{ji}[j\in N(i)]$，并计算出对应的 $P_{ij}[j\in N(i)]$信息发送给变量节点，计算公式为

$$\delta r_{ij}=\prod_{j'\in N(i)\backslash j}\left(\frac{2}{1+Q_{j'i}}-1\right),\tag{14}$$

式中：$\delta r_{ij}$为计算迭代中校验节点传递给变量节点的信息；$\backslash$表示 $j'$是 $N(i)$集合中除去 $j$ 之外的节点。然后根据

接收到的校验子$S_i$进行下一步计算：

$$P_{ij}=\begin{cases}\dfrac{1+\delta r_{ij}}{1-\delta r_{ij}}, S_i=1\\ \dfrac{1-\delta r_{ij}}{1+\delta r_{ij}}, S_i=0\end{cases}。\tag{15}$$

所有的校验节点并行完成上述计算后，任务完成。

3.2.3 变量节点内核

变量节点$j$负责接收与其连接的所有校验节点传来的信息$P_{ij}\left[i\in M(j)\right]$并计算出对应的$Q_{ji}[i\in M(j)]$信息发送给校验节点，计算公式为

$$Q_{ji}=O_j\times\prod_{i'\in M(j)\backslash i}P_{i'j}。\tag{16}$$

所有的变量节点并行完成上述计算后，完成一次迭代。

3.2.4 硬判决译码

在完成给定次数的迭代之后，根据式(17)、(18)进行译码判决：

$$\lambda_j=O_j\times\prod_{i\in M(j)}P_{ij},\tag{17}$$

$$L_j=\begin{cases}1, \lambda_j\geqslant 1\\ 0, \lambda_j<1\end{cases}。\tag{18}$$

将得到的码字与Bob产生的码字$L$进行对比，判断译码成功与否。如不成功，校验节点内核和变量节点内核继续并行运算，并按图2所示方式交换信息继续迭代。

## 4 基于FPGA的OpenCL程序优化

OpenCL异构计算框架是基于主机和加速设备来实现的，其中主机只有CPU这一种设备类型，加速设备则可以采用GPU、FPGA、ASIC芯片等类型的设备。选择FPGA作为加速设备时，实现的难度要远大于GPU。这是因为FPGA是一种非冯·诺依曼结构，内部都是独立的基本逻辑单元和连线逻辑，一般只能通过硬件描述语言进行开发。GPU的架构与CPU比较类似，都属于冯·诺依曼结构，内部集成了大量计算单元，可以直接采用高级语言进行开发。因此，FPGA实现OpenCL代码的机制与GPU完全不同：FPGA是将代码转换成特定的逻辑结构，而非采用处理器来执行代码。这些区别导致了面向GPU的OpenCL代码被直接移植到FPGA上会使得执行效率大幅下降。

选用码率为0.4、码长为$2\times10^5$ bit的码字进行实验，评估程序优化后的提升效果。将GPU上的代码直接移植到FPGA上，测得的解码时间为4.44 s，解码速度为45.03 kbit/s。结合FPGA的特点对程序进行以下优化。

### 4.1 循环优化

每个节点要处理其连接的多个节点的信息，不可避免地要用到循环结构。对FPGA而言，构造格式良好的循环非常重要，这样编译器才能有效地分析这些循环。这就要求循环的退出条件是和一个整数界限作比较，并且每次迭代都有一个简单的归纳增量。以校验节点$i$处理其连接的变量节点集$N(i)$为例，原代码使用的for循环如下：

```
for(ex=&first_in_row(H,i); node_in_row(ex);
ex=&next_in_row(ex)) {……;}
```

其中参数ex为指针，将校验矩阵$\boldsymbol{H}$的第$i$行第一个节点的位置赋予ex作为初值，完成一次循环后指针指向下一个节点，直到ex指向的节点不属于第$i$行时退出循环。这种方式下编译器每次循环都需要访问该节点的行列信息来判断是否满足退出条件，后续迭代在判断完成之前无法启动，从而导致了总体循环性能下降。

给校验节点内核增加一个数组参数$M[n]$，$n$为校验节点的数量，$M[i]$的值为校验矩阵$\boldsymbol{H}$第$i$行的节点总数。新的for循环如下：

```
ex=&first_in_row(H,i);
for(int k=0; k<M[i]; k++)
{……; ex=&next_in_row(ex);}
```

新构造的循环满足退出条件是和一个整数作比较，并且每次迭代后变量$k$加1，因此该循环是一个可以被编译器有效分析且格式良好的循环。

当循环次数为定值时，OpenCL允许通过循环展开来提高性能。OpenCL内核在每个时钟周期对每个工作项执行一次循环迭代，循环展开是以增加FPGA的资源消耗为代价来增加内核在每个时钟周期的工作量，减少内核执行的迭代次数。循环展开后FPGA的资源利用率如表1所示。

由表1可知，循环展开后可更充分地利用FPGA的硬件资源。经过循环优化，测得的解码时间减少到2.84 s；解码速度达到70.45 kbit/s，比优化前提高了56.45%。

表1 FPGA循环展开资源利用率

Table 1 FPGA resource utilization of unroll loops

| Programmable logic resource | Logic | ALUTs | FFs | RAMs | DSPs |
|---|---|---|---|---|---|
| Original resource utilization /% | 33 | 17 | 17 | 28 | 10 |
| Resource utilization of unroll loops /% | 61 | 31 | 32 | 44 | 49 |

### 4.2 内存优化

OpenCL设备的内存模型共有4层结构，分别为全局内存、常量内存、局部内存和私有内存。全局内存允许所有工作组的所有工作项进行读写，常量内存属于全局内存的一部分，在执行一个内核期间保持不变。局部内存只对同一个工作组里的所有工作项可见，私

有内存是一个工作项私有的空间，对其余工作项不可见。所用的高性能FPGA实验板是Intel生产的det5a-net，其内外部资源对应的内存模型如表2所示。

表2　FPGA资源对应的内存模型

Table 2　Memory models corresponding to FPGA resources

| FPGA resource | Arguments | Corresponding memory model |
|---|---|---|
| External memory | 8 GB DDR3 | Global memory |
| Part of external memory | 16 kB | Constant memory |
| M9K memory | 2.5 MB | Local memory |
| Register | 938880 | Private memory |

对于FPGA而言，全局内存的访问代价较大。如果对其访问过于频繁，会造成相当大的性能损失。而局部内存虽然容量比全局内存小得多，但其存储带宽比全局内存大得多，可以达到2.3 TB/s。因此，尽可能地充分利用局部内存访存，并减少全局内存访存成为提升算法性能的关键。借鉴CPU体系结构中高速缓存的思想，又基于LDPC置信传播的特点，设计了哑铃式的内核架构(图3)。

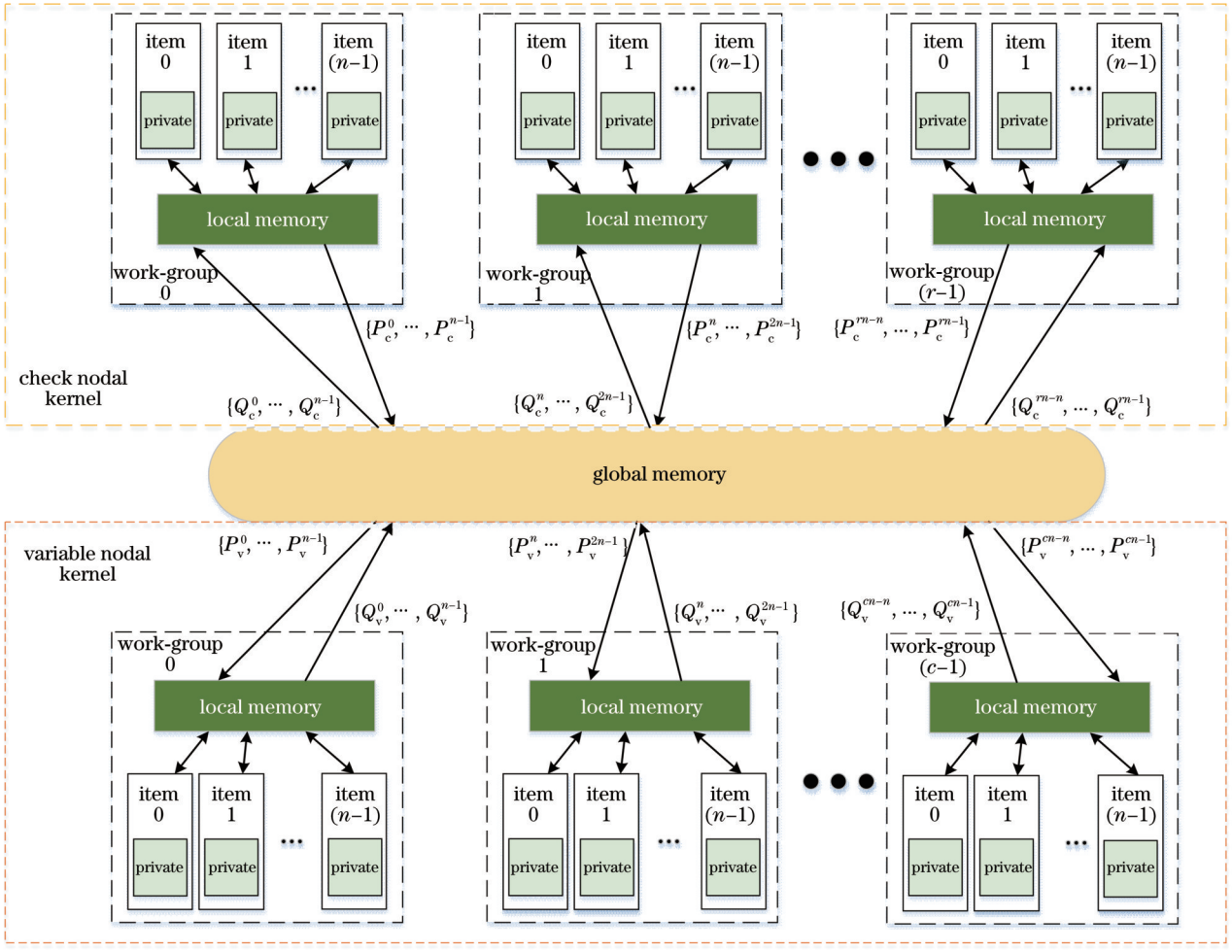


图3　哑铃式内核架构及核内、核间信息流交换示意图

Fig. 3　Dumbbell kernel architecture and internal and inter-core information flow exchange diagram

哑铃结构上部为校验节点内核，中部为全局内存，全局内存中的校验信息按工作组进行剖分并下传，校验节点内核共有$r$个工作组，每个工作组包含$n$个工作项，每个工作项都对应一个校验节点，工作组的大小$n$乘以数量$r$等于校验矩阵$\boldsymbol{H}$的行数。在工作组内，$Q_c^i$是指校验节点$i$从连接的变量节点集接收的信息集$\left\{Q_{ji}, j\in N(i)\right\}$，由式(16)算得；$P_c^i$是指校验节点$i$向连接的变量节点集发送的信息集$\left\{P_{ij}, j\in N(i)\right\}$，由式(15)算得。运算时每个工作组从全局内存中下载需要的信息$\left\{Q_c^i\right\}$进入局部内存，工作组内的每个工作项利用对应的$Q_c^i$算出$P_c^i$，所有工作项运算完成后得到$\left\{P_c^i\right\}$，并将其上传到全局内存，以供变量节点内核读取。

哑铃结构下部为变量节点内核，全局内存中的变量信息按工作组进行剖分并下传，变量节点内核共有$c$个工作组，每个工作组包含$n$个工作项，每个工作项都对应一个变量节点，工作组的大小$c$乘以数量$n$等于校验矩阵$\boldsymbol{H}$的列数。在工作组内，$P_v^j$是指变量节点$j$从

连接的校验节点集接收的信息集$\{P_{ij}, i\in M(j)\}$，由式(15)算得；$Q_v^j$是指变量节点$j$向连接的校验节点集发送的信息集$\{Q_{ji}, i\in M(j)\}$，由式(16)算得。运算时所有工作组从全局内存下载更新后的$\{P_v^j\}$信息到局部内存，工作项运算出结果$\{Q_v^j\}$后上传到全局内存，以供下一次迭代时校验节点内核进行读取。

实验中发现设置不同大小的工作组会影响解码的性能，实验结果如表3所示，其中$A_{\text{ve-time}}$为平均协调时间，$S_{\text{peed}}$为协调速率。由表3可知，工作组变大时，平均解码时间缩短，但当工作组设置过大时性能反而会下降。工作组大小为500时性能达到最佳，解码时间为2.24 s；速度为89.29 kbit/s，比优化前提高了98.29%。

### 4.3　用聚合访问模式增加内核工作量

对于码长为$2\times10^5$ bit的码字，如果每个工作项负责一个变量节点，那么需要有20万个工作项，数量很大，对于设备要求太高。可以通过让每个工作项负责多个节点来减少工作项的数量。

表3　不同大小的工作组性能对比

Table 3　Performance comparison of work-groups of different sizes

| Work-group size | 50 | 100 | 200 | 500 | 800 | 1000 |
|---|---|---|---|---|---|---|
| $A_{\text{ve-time}}$ /s | 4.32 | 3.05 | 2.48 | 2.24 | 2.32 | 2.39 |
| $S_{\text{peed}}$ /(kbit·s$^{-1}$) | 46.35 | 65.48 | 80.63 | 89.29 | 86.21 | 83.68 |

设Addr[$n$]为节点$n$在内存中的地址，相邻的节点在内存中的地址是连续的。由于FPGA的工作项是并行运行的，如果让每个工作项负责相邻的$m$个节点，那么工作项0处理节点0时，工作项1和2也会并行处理节点$m$和$2m$，这会导致同时访问不相邻的内存区域Addr[0]、Addr[$m$]、Addr[$2m$]，破坏了数据访问局部性的要求。本研究把20万个节点按顺序均分为8组，由25000个工作项按组执行，这样既能减少工作项的数量，又能使得并行运算访问相邻的内存区域，这种数据读取模式被称为聚合访问，如图4所示。优化后，测得的解码时间减少到2.05 s；解码速度达到97.59 kbit/s，比优化前提高了116.72%。

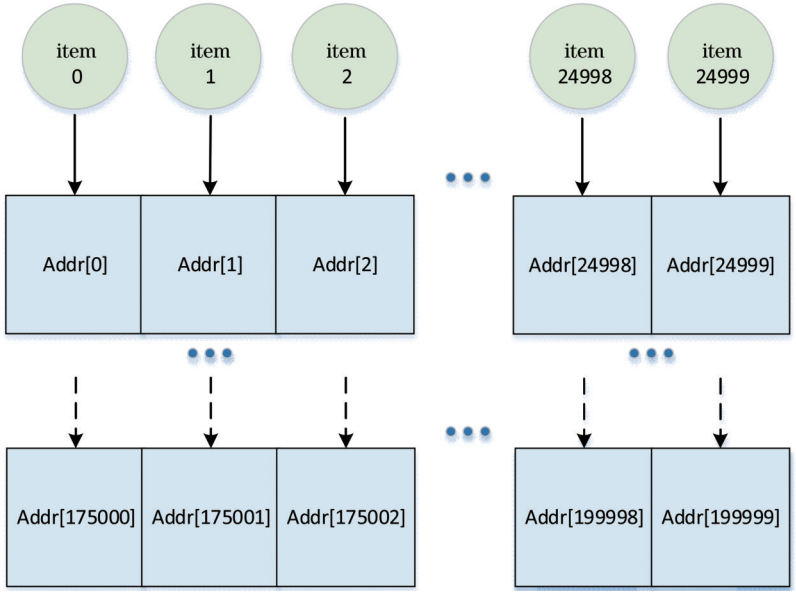


图4　聚合访问模式

Fig. 4　Aggregated access pattern

## 5　仿真结果及分析

经过上述优化后OpenCL在FPGA上也取得了良好的表现。马识途等[16]的研究结果已经表明当码长为$2\times10^5$ bit时协调效率最高，因此本研究对相同码长、不同码率的LDPC码在不同设备上的数据协调性能进行仿真。硬件平台选用内存为32 GB，CPU型号为Intel(R)Xeon(R)E5620的主机设备，将DE5a-Net开发板作为加速设备。主机开发软件为Visual Studio 2010 Professional，FPGA开发软件为Altera Quartus Ⅱ 16.1，内核编译软件采用Altera SDK for OpenCL 16.1。所用校验矩阵由Mackay随机构造法构成，采用八维数据协调算法。表4对比了CPU平台和CPU/FPGA异构平台在不同码率时的数据协调速度，其中$R_{\text{ate}}$为码率，$R_{\text{SN}}$为收敛信噪比，$A_{\text{ve-time}}$为平均协调时间，$S_{\text{peed}}$为协调速率，由码长除以平均协调时间得到。从表4可以看出，随着码率的增加，收敛信噪比增大，协调速率降低。这是因为码率增加后校验矩阵的行数减少，即校验节点减少了，想要完成译码则会对信噪比有更高的要求；同样的原因导致码率越高所需要的迭代次数越多，平均协调时间变长，协调速率降低。从表4还可看出，异构平台的协调速率是CPU平台的4倍

以上。

随着码率的增加，总体的运算量也在增大，此时异构平台的协调速率可以达到CPU平台的5倍以上。表5对比了不同码率时两个平台的安全密钥量，其中：$\beta$为协调效率，由式(8)计算得出；$R_{\text{ate-bit}}$为安全密钥率，由式(7)计算得出；$D_{\text{istance}}$为传输距离，由式(9)计算得出。安全密钥率$R_{\text{ate-bit}}$为负值，表示此次密钥分发失败，不能得到安全密钥。只有在确保能获取到安全密钥的前提下，协调速率的提升才有意义。由表5可知，当码率为0.1～0.4时，协调效率高于95%，可以获得安全密钥。采用异构方案后部分码率的收敛信噪比升高，协调效率降低，这是因为多维协调算法对浮点数的精度要求很高，而FPGA在实现过程中使用了多种近似手段，比如定点数算法等，降低了实现的复杂度，但代价是牺牲了计算的准确度。可以在编译时加入一些参数来尽可能去除运算过程中四舍五入的截位操作，保留多余的小数位，这样可以得到更高精度的计算结果。

表4 码率不同时CPU和CPU/FPGA异构平台的实验结果

Table 4 Experimental results of CPU platform and CPU/FPGA heterogeneous platform at different code rates

| $R_{\text{ate}}$ | CPU | | | CPU/FPGA | | |
|---|---|---|---|---|---|---|
| | $R_{\text{SN}}$ /dB | $A_{\text{ve-time}}$ /s | $S_{\text{peed}}$ /(kbit·s$^{-1}$) | $R_{\text{SN}}$ /dB | $A_{\text{ve-time}}$ /s | $S_{\text{peed}}$ /(kbit·s$^{-1}$) |
| 0.1 | 0.15 | 3.9544 | 50.58 | 0.15 | 0.9858 | 202.88 |
| 0.2 | 0.33 | 4.4656 | 44.79 | 0.33 | 1.1108 | 180.05 |
| 0.3 | 0.54 | 5.8448 | 34.22 | 0.55 | 1.3223 | 151.25 |
| 0.4 | 0.79 | 9.9538 | 20.09 | 0.81 | 2.0494 | 97.59 |
| 0.5 | 1.38 | 14.1742 | 14.11 | 1.43 | 2.8578 | 69.98 |
| 0.6 | 2.42 | 17.7690 | 11.26 | 2.49 | 3.4542 | 57.90 |
| 0.7 | 4.67 | 18.2994 | 10.93 | 4.80 | 3.5741 | 55.96 |

表5 码率不同时CPU和CPU/FPGA平台的密钥量对比

Table 5 Comparison of key quantity between CPU platform and CPU/FPGA platform at different code rates

| $R_{\text{ate}}$ | CPU | | | | CPU/FPGA | | | |
|---|---|---|---|---|---|---|---|---|
| | $R_{\text{SN}}$ /dB | $\beta$ /% | $R_{\text{ate-bit}}$ /(kbit·s$^{-1}$) | $D_{\text{istance}}$ /km | $R_{\text{SN}}$ /dB | $\beta$ /% | $R_{\text{ate-bit}}$ /(kbit·s$^{-1}$) | $D_{\text{istance}}$ /km |
| 0.1 | 0.15 | 99.19 | 19.08 | 49.25 | 0.15 | 99.19 | 19.08 | 49.25 |
| 0.2 | 0.33 | 97.22 | 14.29 | 48.35 | 0.33 | 97.22 | 14.29 | 48.35 |
| 0.3 | 0.54 | 96.32 | 12.09 | 47.30 | 0.55 | 94.90 | 8.62 | 47.25 |
| 0.4 | 0.79 | 95.24 | 9.46 | 46.05 | 0.81 | 93.46 | 5.11 | 45.95 |
| 0.5 | 1.38 | 79.94 | −27.85 | 43.10 | 1.43 | 78.07 | −32.41 | 42.85 |
| 0.6 | 2.42 | 67.64 | −57.82 | 37.90 | 2.49 | 66.55 | −60.49 | 37.55 |
| 0.7 | 4.67 | 55.93 | −86.39 | 26.65 | 4.80 | 55.20 | −88.15 | 26.00 |

表6对比了同样采用OpenCL异构框架，以GPU为加速设备进行数据协调[23]的实验结果。由表6可知，在可获得安全密钥的前提下，以FPGA为加速设备的异构平台在优化算法之后协调速率已经超过了以GPU为加速设备的异构平台，当码率为0.3时，其协调速率达到GPU设备的2倍以上。这是因为循环优化和内存优化可显著提升FPGA译码性能。由于文献[23]的码率最低只到0.3，无法对比码率更低情况下的协调速率。

表6 码率不同时CPU/GPU和CPU/FPGA异构平台的实验结果

Table 6 Experimental results of CPU/GPU and CPU/FPGA heterogeneous platforms at different code rates

| $R_{\text{ate}}$ | CPU/GPU | | | CPU/FPGA | | |
|---|---|---|---|---|---|---|
| | $R_{\text{SN}}$ /dB | $A_{\text{ve-time}}$ /s | $S_{\text{peed}}$ /(kbit·s$^{-1}$) | $R_{\text{SN}}$ /dB | $A_{\text{ve-time}}$ /s | $S_{\text{peed}}$ /(kbit·s$^{-1}$) |
| 0.3 | 0.55 | 2.84 | 70.42 | 0.55 | 1.3223 | 151.25 |
| 0.4 | 1.00 | 2.90 | 68.97 | 0.81 | 2.0494 | 97.59 |

## 6 结 论

针对CV-QKD数据协调通信时延长的问题，采用异构计算的方式加速解码。协调系统采用接近香农限的LDPC码作为纠错码，八维数据协调算法为协调方案，用OpenCL框架搭建了CPU/FPGA异构平台。在针对FPGA的特点对OpenCL代码进行了一系列优化后，协调速率提升了116.72%，达到97.59 kbit·s$^{-1}$，其中循环优化的提升最大，可达56%；哑铃式内核架构通过减少访存消耗也可提升近42%的性能。在相同

的码长、码率下，所提方法的协调速率是同样采用八维数据协调算法的文献[24]方法的8.2倍。利用优化后的代码对相同码长、不同码率的码字在CPU和FPGA异构平台分别进行仿真，结果显示采用异构平台并行加速的协调速率是单一CPU平台的4倍以上。在确保能获取安全密钥的前提下所使用的CPU/FPGA异构平台的协调速率已经超过了贺超等[23]使用的CPU/GPU异构平台。

此外，还可以通过优化LDPC码的度分布或构造多边类型的LDPC码[25]来提高协调性能。在多维数据协调中，这种特殊的码可以取得更低的收敛信噪比，从而增大传输距离。

## 参 考 文 献


[1] Bennett C H, Brassard G. Quantum cryptography: public key distribution and coin tossing[J]. Theoretical Computer Science, 2014, 560: 7-11.

[2] Bennett C H, Brassard G. Quantum cryptography: public key distribution and coin tossing[EB/OL]. (2020-03-14) [2021-05-06]. https://arxiv.org/abs/2003.06557.

[3] Gyongyosi L, Bacsardi L, Imre S. A survey on quantum key distribution[J]. Infocommunications Journal, 2019(2): 14-21.

[4] Grosshans F, Grangier P. Continuous variable quantum cryptography using coherent states[J]. Physical Review Letters, 2002, 88(5): 057902.

[5] 靳晓丽. ~kHz量级平衡零拍探测器的研究[J]. 激光与光电子学进展, 2021, 58(9): 0927003.
Jin X L. Study on kHz balanced homodyne detector[J]. Laser & Optoelectronics Progress, 2021, 58(9): 0927003.

[6] 刘俊良, 许伊宁, 董亚魁, 等. 集成型快速主动淬灭InGaAsP近红外单光子探测器[J]. 中国激光, 2021, 48(12): 1212002.
Liu J L, Xu Y N, Dong Y K, et al. Integrated InGaAsP near-infrared single-photon detector with fast active quenching[J]. Chinese Journal of Lasers, 2021, 48(12): 1212002.

[7] Guo D B, He C, Guo T H, et al. Comprehensive high-speed reconciliation for continuous-variable quantum key distribution [J]. Quantum Information Processing, 2020, 19(9): 320.

[8] Yang S S, Liu J Q, Lu Z G, et al. An FPGA-based LDPC decoder with ultra-long codes for continuous-variable quantum key distribution[J]. IEEE Access, 2021, 9: 47687-47697.

[9] Zhang G, Haw J Y, Cai H, et al. An integrated silicon photonic chip platform for continuous-variable quantum key distribution [J]. Nature Photonics, 2019, 13(12): 839-842.

[10] Wen X, Li Q, Mao H K, et al. An improved slice reconciliation protocol for continuous-variable quantum key distribution[J]. Entropy, 2021, 23(10): 1317.

[11] Silberhorn C, Ralph T C, Lütkenhaus N, et al. Continuous variable quantum cryptography: beating the 3 dB loss limit[J]. Physical Review Letters, 2002, 89(16): 167901.

[12] Leverrier A, Alléaume R, Boutros J, et al. Multidimensional reconciliation for a continuous-variable quantum key distribution [J]. Physical Review A, 2008, 77(4): 042325.

[13] Bloch M, Thangaraj A, McLaughlin S W, et al. LDPC-based Gaussian key reconciliation[C]//2006 IEEE Information Theory Workshop-ITW '06 Punta del Este, March 13-17, 2006, Punta del Este, Uruguay. New York: IEEE Press, 2006: 116-120.

[14] 郭大波, 张彦煌, 王云艳. 高斯量子密钥分发数据协调的性能优化[J]. 光学学报, 2014, 34(1): 0127001.
Guo D B, Zhang Y H, Wang Y Y. Performance optimization for the reconciliation of Gaussian quantum key distribution[J]. Acta Optica Sinica, 2014, 34(1): 0127001.

[15] Lin D K, Huang D, Huang P, et al. High performance reconciliation for continuous-variable quantum key distribution with LDPC code[J]. International Journal of Quantum Information, 2015, 13(2): 1550010.

[16] 马识途, 郭大波, 薛哲, 等. 基于双边类型低密度奇偶校验码的连续变量量子密钥分发多维数据协调[J]. 光学学报, 2019, 39(5): 0527001.
Ma S T, Guo D B, Xue Z, et al. Multidimensional reconciliation for continuous-variable quantum key distribution based on two-edge type low-density parity-check codes[J]. Acta Optica Sinica, 2019, 39(5): 0527001.

[17] Slepian D, Wolf J. Noiseless coding of correlated information sources[J]. IEEE Transactions on Information Theory, 1973, 19 (4): 471-480.

[18] Leverrier A, Grangier P. Continuous-variable quantum key distribution protocols with a discrete modulation[EB/OL]. (2010-02-22)[2021-08-09]. https://arxiv.org/abs/1002.4083.

[19] Jouguet P, Kunz-Jacques S, Leverrier A. Long-distance continuous-variable quantum key distribution with a Gaussian modulation[J]. Physical Review A, 2011, 84(6): 062317.

[20] 刘勇飞, 杨春燕, 赵露涵, 等. 基于传送带量子纠缠光的卫星钟差测量[J]. 光学学报, 2022, 42(4): 0427001.
Liu Y F, Yang C Y, Zhao L H, et al. Satellite clock offset measurement based on conveyor belt quantum entangled light[J]. Acta Optica Sinica, 2022, 42(4): 0427001.

[21] Gallager R. Low-density parity-check codes[J]. IEEE Transactions on Information Theory, 1962, 8(1): 21-28.

[22] 林毅, 何广强, 曾贵华. LDPC码在量子密钥分配多维协商算法中的应用[J]. 量子光学学报, 2013, 19(2): 116-121.
Lin Y, He G Q, Zeng G H. The application of LDPC codes in the multidimensional reconciliation of quantum key distribution [J]. Acta Sinica Quantum Optica, 2013, 19(2): 116-121.

[23] 贺超, 郭大波, 穆健健, 等. 基于OpenCL/GPU异构计算的高速数据协调系统设计[J]. 量子光学学报, 2019, 25(3): 273-281.
He C, Guo D B, Mu J J, et al. A system design for high-speed data reconciliation based on OpenCL/GPU heterogeneous computing[J]. Journal of Quantum Optics, 2019, 25(3): 273-281.

[24] 王云艳, 郭大波, 张彦煌, 等. 连续变量量子密钥分发多维数据协调算法[J]. 光学学报, 2014, 34(8): 0827002.
Wang Y Y, Guo D B, Zhang Y H, et al. Algorithm of multidimensional reconciliation for continuous-variable quantum key distribution[J]. Acta Optica Sinica, 2014, 34(8): 0827002.

[25] Richardson T J, Urbanke R L. Modern coding theory[M]. Cambridge: Cambridge University Press, 2008.

# Design of Data Reconciliation System Based on FPGA Heterogeneous Computing

Liu Jiasen[1], Guo Dabo[1,2*], Guo Tianhao[1], Li Xianzhong[1], Wang Yujie[1], Meng Yingxiu[1]

[1]*College of Physics and Electronic Engineering, Shanxi University, Taiyuan* 030006, *Shanxi, China*;

[2]*School of Information and Intelligent Engineering, University of Sanya, Sanya* 572022, *Hainan, China*

## Abstract

**Objective** Quantum key distribution (QKD) is the earliest practical technology in the field of quantum communication, which has absolute security in theory. There are two kinds of QKD systems according to their source encoding dimensions: continuous-variable QKD (CV-QKD) and discrete-variable QKD (DV-QKD). Compared with DV-QKD, CV-QKD systems have such advantages: 1) modulation and decoding of CVs do not require special devices and can be implemented effectively by standard telecommunication networks; 2) the detection efficiency of homodyne or heterodyne detector used by CV-QKD is higher than that of the single-photon detector used by DV-QKD at room temperatures. Shannon′s theorem suggests that the longer code length suffices for a more stable performance. Therefore, the CV-QKD system generally adopts great code length, and the number of optical pulses involved in data reconciliation reaches $10^5$. However, such a long block length brings about a much high calculated quantity. This inevitably results in a low speed of data reconciliation, which restricts the throughput and the key rates of the CV-QKD system. Given this, the paper adopts hardware devices to accelerate the decoding process. Open computing language (OpenCL) can process data at high speed by means of parallel computation. FPGA is highly parallel and can achieve high performance with ultra-low power consumption. Therefore, the combination of OpenCL and FPGA for accelerated computing becomes a good solution.

**Methods** To tackle the problem of low computing speed of data reconciliation in the current continuous variable quantum key distribution system, this paper proposes an eight-dimensional data reconciliation algorithm by adopting a high-performance FPGA board as the acceleration device on the OpenCL heterogeneous computing framework. According to the characteristics of FPGA, the algorithm is optimized as follows. 1) The expression of for loops is optimized so that the OpenCL compiler can better understand the intention of the designer to generate a more effective FPGA hardware structure. 2) Memory optimization: according to the characteristics of the belief propagation algorithm for low-density parity-check code (LDPC) decoding, a dumbbell-type core architecture and information transmission mode within and between cores is designed. 3) Aggregated access data read pattern is applied to reduce the number of parallel work items. The program written in OpenCL after the above optimization has also achieved good performance. Then, simulations are performed with the optimized algorithm on a CPU/FPGA heterogeneous platform. The results are compared with the experimental results of the CPU platform.

**Results and Discussions** The simulation results show that when the code length is $2\times10^5$ bit, the reconciliation speed after optimization is 2.17 times that before optimization. The reconciliation speed using parallel acceleration in OpenCL/FPGA heterogeneous platform is more than 4 times that in the single CPU platform (Fig. 4). As the code rate increases, the signal noise ratio (SNR) asymptotic threshold RSN increases and the reconciliation speed decreases (Fig. 4). This is because the number of rows of the check matrix declines after the code rate increases, namely that the number of check nodes reduces, and a higher SNR is required to complete the decoding. The same reason leads to more iterations required for higher code rates, longer average reconciliation time, and lower reconciliation speed. The increase in reconciliation speed only makes sense if it ensures that the security key can be obtained. When the code rate is 0.1-0.4, the reconciliation efficiency $\beta$ is higher than 95% and the security key can be obtained (Fig. 5). The multi-dimensional reconciliation algorithm requires high precision for floating-point numbers, and FPGA uses a variety of approximations in its implementation, which reducing the complexity of the implementation at the expense of computational accuracy. This leads to a higher SNR asymptotic threshold and lower reconciliation efficiency with the heterogeneous scheme. After optimizing the algorithm, the reconciliation speed of the FPGA-accelerated device already exceeds that of the GPU-accelerated device (Fig. 6). This is because loop optimization and memory optimization can significantly improve FPGA decoding performance.

**Conclusions** To address the problem of slow computing speed of data reconciliation in CV-QKD systems, heterogeneous computing is adopted to accelerate decoding. The reconciliation system takes LDPC codes close to the Shannon limit as the error correction codes and the eight-dimensional data reconciliation algorithm as the reconciliation scheme. A CPU/

FPGA heterogeneous platform is built with the OpenCL framework. After a series of optimizations of the OpenCL code based on the characteristics of the FPGA, the reconciliation speed is increased by 116.72% to 97.59 kbit/s. The loop optimization provides the largest improvement of up to 56%, while the dumbbell-type core architecture can also improve performance by nearly 42% by reducing access memory consumption. Simulations are performed at different code rates on CPU and FPGA heterogeneous platforms separately. The results show that the reconciliation speed using the parallel acceleration of heterogeneous platforms is more than 4 times that of single CPU platforms. The reconciliation speed of CPU/FPGA heterogeneous platforms has exceeded that of CPU/GPU heterogeneous platforms while ensuring that a security key can be obtained.

**Key words** quantum optics; quantum key distribution; multidimensional reconciliation; OpenCL; FPGA; LDPC code