



西南科技大学

Southwest University of Science and Technology

本科毕业设计（论文） 教学过程记录册

题目名称：快速傅里叶变换的
并行算法研究即实现

学 院 名 称	计 算 机 科 学 与 技 术 学 院
专 业 名 称	软 件 工 程
学 生 姓 名	肖 劲 涛
学 号	5120184509
指 导 教 师	苏 波 讲 师

二〇二二年六月

西南科技大学本科毕业设计（论文）任务书

题 目	快速傅里叶变换的并行算法研究及实现		
学 院	计算机科学与技术学院	专业班级	软件 1804
学 号	5120184509	学生姓名	肖劲涛
题目来源	科研项目（ ） 生产实践（社会实际）（ ） 教师自拟（√） 学生自拟（ ）		请直接在所属项目括号内打“√” （单选）
题目性质	理论研究（ √ ） 应用研究（ ） 技术开发（ ）		
题目类型	设 计（√） 论 文（ ）		
选题意义及目的	<p>傅立叶变换是一种线性积分变换，用于信号在时域（或空域）和频域之间的变换，在物理学和工程学中有许多应用。傅里叶变换源自对傅里叶级数的研究。在对傅里叶级数的研究中，复杂的周期函数可以用一系列简单的正弦、余弦波之和表示。傅里叶变换是对傅里叶级数的扩展，由它表示的函数的周期趋近于无穷。</p> <p>傅立叶变换在许多领域都有广泛的用途，例如在讯号处理中，傅里叶变换的典型用途是将讯号分解成振幅分量和频率分量。基于它的这一特点，我们可以将一个给定的函数的频域变换到时域，或是逆变换。为了探究如何快速完成这一变换，我们将从多种不同的角度入手对比傅立叶变换的效率。</p>		

内容 及 要求	<p>设计要求：</p> <p>（1）实现快速傅立叶变换的 CPU 并行、CPU 串行、GPU 编程，并对比效率；</p> <p>（2）CPU 并行计算使用 C++多线程与 MPI 多进程完成；</p> <p>（3）使用 CUDA 或者 OpenCL 完成 GPU 编程；</p> <p>（4）使用 Unreal 或 Unity 或 OpenGL 完成海洋频率到时域的转换模拟；</p> <p>（5）从运行时间、问题规模、并行系统的性能分析、加速比、效率等进行分析 and 讨论，给出指导性报告。</p> <p>论文要求：</p> <p>（1）撰写不低于 1.5 万字的毕业设计论文。</p> <p>（2）论文参考文献不低于 15 篇，其中期刊文献不低于 10 篇，外文文献不低于 3 篇。</p> <p>（3）论文质量需要符合计算机科学与技术学院该专业的 2016 版人才培养方案。</p>		
时间 安排	<p>1. 开题报告： 2022 年 3 月 5 日至 2022 年 3 月 12 日。</p> <p>2. 完成初稿： 2022 年 3 月 19 日至 2022 年 5 月 15 日。</p> <p>3. 答 辩： 2022 年 6 月 6 日至 2022 年 6 月 7 日。</p>		
以上内容由指导教师填写			
指导教师 签字	<p>教师签名：</p> <p>2022 年 1 月 7 日</p>	<p>学院 审核</p>	<p>审核意见：</p> <p>组长签字： 2022 年 1 月 8 日</p>
接受任务 签字	<p>学生签名：</p> <p>接受任务时间： 2022 年 1 月 9 日</p>		

西南科技大学本科毕业设计（论文）开题报告

学 院	计算机科学与技术学院	专 业	软件工程	班 级	软件 1804
姓 名	肖劲涛	学 号	5120184509	指导教师	苏波
设计（论文）题目	快速傅里叶变换的并行算法研究及实现				
一、选题背景（目的、意义）					
<p>（一）选题背景：</p> <p>离散傅里叶变换（DFT），是傅里叶变换在时域和频域上都呈离散的形式，是科学与工程领域中一个重要的数学方法^[1-3]。给定一个长度为 N 的队列 $x(n)$，它的 DFT 也是一个长度为 N 的队列 $X(k)$，</p> $X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \dots, N-1 \quad (\text{式 1})$ <p>其中，$W_N = e^{-j2\pi/N}$，$j = \sqrt{-1}$。傅里叶变换认为，任何连续的时域信号，都可以表示为不同的正弦波信号的叠加。傅里叶变换算法直接利用时序采样信号，以累加的方式计算时序采样信号中不同正弦波的频率信号的振幅、频率以及相位。通过这种方法，傅里叶变换将原本难以处理的时域信号变换为易于处理的频域信号，可以利用一些工具对其加工、处理。最后可以通过傅里叶逆变换，将频域信号重新变换为时域信号。傅里叶逆变换从本质上来说，也是一种累加处理算法。</p> <p>在不同的研究领域，傅里叶变换具有多种描述形式。用现代数学的观点看，傅里叶变换将满足一定条件的函数表示为积分或正弦函数的线性组合，从这点来看，傅里叶变换可以视作一种积分的特殊形式。</p> <p>DFT 通常使用一种叫快速傅里叶变换（FFT）的技术来实现，即 FFT 是 DFT 的快速算法。同时，FFT 也可用作 DFT 的逆变换。直接计算 DFT 的时间复杂度是 $O(N^2)$，而使用 FFT 的时间复杂度则是 $O(N\log N)$。通常，快速傅里叶变换要求 N 内部因子分解，但不是所有的 FFT 都要求 N 是合数。对于所有的整数 N，都存在复杂度为 $O(N\log N)$ 的 FFT。</p> <p>随着近些年来数字计算机和大规模电路的不断发展，中央处理单元（CPU）与图形处理单元（GPU）的核心数量与核心频率得到了显著提高。如今，我们面临的问题的数据规模日益增加，传统的串行算法在面对大量数据时不能充分利用处理器的多个内核。这要求编写软件程序的思想发生变化，以适应硬件的变化，并充分利用硬件的性能。同时，针对大规模的 FFT 计算，GPU 相比于 CPU 在处理能力和存储带宽上有较为明显的优势。CPU 的设计兼顾了不同任务的需要，其晶体管都用于大量的缓存和复杂的逻辑控制，相对来说运算单元所占不多。多核 CPU 的发展使得其晶体管数量得到了增加，但并没有增加 CPU 的利用率。而 GPU 提供了跟</p>					

多的计算单元和存储控制单元，使得在大规模数据计算以及存储带宽得到提高。CPU 程序通常时单线程编写，如果有需要则可以使用多进程、多线程编程技术，而 GPU 则默认并行计算模式。得益于其结构设计，GPU 使得每次可以同时计算多个数据，而不是像 CPU 一次只能计算一个数据。但与之相对的，则是在复杂计算与指令上，GPU 的计算速度不如 CPU。

(2) 选题意义：

傅里叶变换在信号处理、力学、数学、金融等领域都有着广泛的应用^[4-7]。傅里叶变换的一个典型用途是，将时域信号分解为为频域内振幅分量和频率分量。直接计算长度为 N 的序列卷积的离散傅里叶变换（DFT）的复杂度为 $O(N^2)$ 。Cooley-Tukey 快速傅里叶变换(FFT)^[8]将计算复杂度降低到了 $O(N\log N)$ 。

尽管 FFT 表现不凡，但还是有很多问题需要解决。这些方面包括：计算精度的提高、高效低功耗的 FFT 处理器设计、计算复杂度在理论与实际的最小边界、FFT 数据迁移量的减少、系数访问效率的改善等。本项目将对比不同的傅里叶变换算法在不同的程序设计方案（串行、并行等）下的性能，并给出指导性报告。

二、国内外研究现状综述

快速傅里叶变换最早由 Guass 提出，后来又被 Runge 和 Konig 发现^[9]。但直到 1965 年，快速傅里叶变换被 Cooley-Tukey 重新提出后才得到了充分认识^[8]。从此以后，关于快速傅里叶变换的研究快速增加，包含了高阶基- K FFT 算法^[10]、分裂基 FFT 算法^[11-14]、混合基算法^[15, 16]、质数因子算法^[17-19]、递归 FFT 算法^[20]、Winograd 傅里叶变换算法^[21, 22]。

高斯在 18 世纪使用和 Cooley-Tukey 相同的分治法计算三角级数。值得注意的是，因其算法基于任何长度的整数的队列变换，其拥有很高的通用性。在 1903 年，Runge 提出了一个用来计算 2 的幂次方的队列长度算法，该算法也能用来计算 3 的幂次方。1965 年，Cooley 和 Tukey 提出了一种快速傅里叶变换算法^[8]，该方法减少了长度为 $N=2^m$ 的 DFT 操作数的阶数，使其从 N^2 降低到了 $N\log_2 N$ 。同时，该算法适应任何长度的 DFT。

当一个 DFT 被分解为一些不是互素的子 DFT 时，分治会导致辅助的复数乘的出现，最初被叫做旋转因子。Coolkey-Tukey 算法适用于任意长度的 DFT，按通用形式解释。Coolkey 和 Tukey 给出了一个长度为 $N=2^m$ 的例子，即提出了一个现在叫按时域分解的基-2FFT 算法。

1965 年，Coolkey-Tukey 算法可以理解为一个假的一维到多维的镜像。而 Good 算法^[23]可以在变换长度为因子互素乘的 DFT 中，实现真正的一维到多维的镜像，并且不产生旋转因子。Good 算法适应的分解长度，其因子时互素的；并不适用长度为 K 的幂次方的情况。Rader 的方法^[10]展示了如何将一个长度为 N 的 DFT 镜像为一个长度为 $N-1$ 的循环卷积，其中 N 为素数。Winograd 算法^[22]，首先使用 Good 算法镜像一个 DFT 成多维的 DFT 后，使用卷积方案来得到一个多种乘法的循环。9

Jiang 在文章^[24]中提出，以前的 FFT 方法在存储访问问题。除非处理器提供了大量寄存器，否则重复的访问寄存器去装载旋转因子时不可避免的。同时，Jiang 提出了基-2FFT 宽度优先算法。 N -点基-2FFT 算法的两个情况考虑宽度优先算法，即 W^0 和 $W^j(j \neq 0)$ 。对应的

FFT 结构被组织成两个阶段，第一阶段计算所有需要被 $W^j (j \neq 0)$ 乘的蝶。在该阶段，系数 W^j 被装入寄存器，一直使用到当前处理层不需要使用到为止。该方法系数被装载的次数为 $N/2 - 1$ ，前面提到的方法则是 $N \log N$ 次。第二个阶段计算所有需要被 W^0 乘的所有蝶。

麻省理工大学 Frigo 和 Johnson 开发的 FFTW 软件库^[1, 2]，可以计算任意长度、一维或多维的实数或复数 DFT。FFTW 通过支持多种算法并选择它估计或测量在特定情况下更可取的算法（将转换特定分解为更小的转换）来快速转换数据。它在具有小素数因子大小的数组上效果最佳，2 的幂次方和大素数情况下效果最差，但其复杂度仍然是 $N \log_2 N$ 。该软件库使用 Cooley-Tukey 算法、Rader 算法或 Bluestein 算法。

英伟达公司提供了一种基于 GPU 的 FFT 算法：CUFFT。CUFFT 是 CUDA 的函数库，其相对于 CPU 在运算速度上有着明显优势，但仍然未能充分发挥 GPU 在并行计算方面的优势。Govindaraju、Lloyd 和 Dotsenko 提出了一种在 GPU 上运行的分层混合基 FFT 算法，该算法通过 Stockham 公式利用 GPU 上的共享内存来减少分层算法中的存储器转置开销。相比于 CUFFT 算法，该算法的性能提高了 2-4 倍。Gu、Li 和 Siegel 提出的一种基于多维 Cooley-Tukey 算法的 GPU 实现方案，通过减少计算内核的数量，并优化最小全局存储器的访问数量来提高算法效率，该算法相比于 CUFFT 算法有着精度上的优势。

三、研究目标与研究内容

(1) 研究目标:

本项目内容主要为基于对 CPU 与 GPU 上 FFT 的并行算法的研究。同时介绍几种典型的实现多线程、多进程编程方法。在此基础上分析 FFT 算法的可行性，并给出多线程优化。

(2) 具体研究内容包括:

- (1) 介绍多种传统串行 FFT 算法，包括 Cooley-Tukey 算法、Rader 算法等。
- (2) 介绍多种并行技术下的 FFT 算法，包括 CUFFT 算法、Govindaraju-Lloyd-Dotsenko 算法等。
- (3) 比较不同算法之间的性能，评估方法有：蝶分析、算法复杂度分析、旋转因子分析等。
- (4) 通过不同算法直接比较，给出并行 FFT 算法的选择、优化建议。

四、拟采用的研究思路（方法、技术路线、可行性论证等）

- (1) 查询文献。利用图书馆、档案馆及互联网等途径，广泛查找相关的文献资料，加以分析与研究。
- (2) 构建模型。使用 C++编写串行的 FFT 算法；利用 MPI 或 OpenMP，同时使用 C++实现基于 CPU 的多进程的 FFT 算法；利用 C++的 STL 库实现基于 CPU 的多线程 FFT 算法；利用 CUDA 与 C++实现基于 GPU 的并行 FFT 算法。
- (3) 算法分析。通过不同的指标参数，分析各 FFT 算法的性能。

五、研究工作进度安排

- (1) 2022 年 1 月 20 日—2022 年 2 月 10 日 查阅有关资料，明确课题研究的目的及意义。
- (2) 2022 年 2 月 11 日—2022 年 2 月 21 日 编写串行 FFT 算法。
- (3) 2022 年 2 月 22 日—2022 年 4 月 1 日 编写并行 FFT 算法并进行需求分析。
- (4) 2022 年 4 月 1 日—2022 年 4 月 19 日 分析各 FFT 算法的性能，并给出意见与建议。
- (5) 2022 年 3 月 19 日—2022 年 5 月 15 日 根据论文撰写规范，完成初稿。
- (6) 2022 年 5 月 16 日—2022 年 6 月 1 日 完善论文，准备答辩。

六、参考文献

- [1] FRIGO M, JOHNSON S G. FFTW: An adaptive software architecture for the FFT; proceedings of the Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat No 98CH36181), F, 1998 [C]. IEEE, 1998.
- [2] FRIGO M, JOHNSON S G. The design and implementation of FFTW3 [J]. Proceedings of the IEEE, 2005, 93(2): 216-231.
- [3] KATOH K, KUMA K-I, TOH H, et al. MAFFT version 5: improvement in accuracy of multiple sequence alignment [J]. Nucleic acids research, 2005, 33(2): 511-518.
- [4] 任山. 基于查找表的 FFT CUDA 并行算法研究 [D]. 长沙:湖南大学, 2017.
- [5] 徐金棒. 基于多核多线程的 FFT 算法和堆排序算法的并行优化和实现 [D]. 郑州:郑州大学, 2011.
- [6] 郑伟华. 快速傅立叶变换-算法及应用 [D]. 长沙:湖南大学, 2015.
- [7] 周益民. 图像处理并行算法的研究 [D]. 长沙:电子科技大学, 2006.
- [8] COOLEY J W, TUKEY J W. An algorithm for the machine calculation of complex Fourier series [J]. Mathematics of computation, 1965, 19(90): 297-301.

- [9] PRESS W H, TEUKOLSKY S A, FLANNERY B P, et al. Numerical recipes in Fortran 77: volume 1, volume 1 of Fortran numerical recipes: the art of scientific computing [M]. Cambridge Eng: Cambridge university press, 1992.
- [10] RADER C M. Discrete Fourier transforms when the number of data samples is prime [J]. Proceedings of the IEEE, 1968, 56(6): 1107-1108.
- [11] BOUGUEZEL S, AHMAD M O, SWAMY M. Improved radix-4 and radix-8 FFT algorithms; proceedings of the 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat No 04CH37512), F, 2004 [C]. IEEE, 2004.
- [12] BOUGUEZEL S, AHMAD M O, SWAMY M. A general class of split-radix FFT algorithms for the computation of the DFT of length- 2^m [J]. IEEE Transactions on signal processing, 2007, 55(8): 4127-4138.
- [13] BOUGUEZEL S, AHMAD M O, SWAMY M S. A new radix-2/8 FFT algorithm for length-q/spl times/2/sup m/DFTs [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2004, 51(9): 1723-1732.
- [14] DUHAMEL P, HOLLMANN H. Split radix'FFT algorithm [J]. Electronics letters, 1984, 20(1): 14-16.
- [15] HSIAO C-F, CHEN Y, LEE C-Y. A generalized mixed-radix algorithm for memory-based FFT processors [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2010, 57(1): 26-30.
- [16] JO B G, SUNWOO M H. New continuous-flow mixed-radix (CFMR) FFT processor using novel in-place strategy [J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2005, 52(5): 911-919.
- [17] BLUESTEIN L. A linear filtering approach to the computation of discrete Fourier transform [J]. IEEE Transactions on Audio and Electroacoustics, 1970, 18(4): 451-455.
- [18] KOLBA D, PARKS T. A prime factor FFT algorithm using high-speed convolution [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1977, 25(4): 281-294.
- [19] BURRUS C, ESCHENBACHER P. An in-place, in-order prime factor FFT algorithm [J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1981, 29(4): 806-817.
- [20] MARTENS J-B. Recursive cyclotomic factorization--A new algorithm for calculating the discrete Fourier transform [J]. IEEE transactions on acoustics, 1984, 32(4): 750-761.
- [21] JIANG Y, ZHOU T, TANG Y, et al. Twiddle-factor-based FFT algorithm with reduced memory access; proceedings of the Proceedings 16th International Parallel and Distributed Processing Symposium, F, 2002 [C]. IEEE, 2002.

<p>指导教师 意见</p>	<p>该生针对快速傅里叶变换算法研究从选题背景、目的和意义以及国内外研究现状进行了综述。指出了 FFT 算法在不同行业使用的重要程度，并且指出了部分 FFT 算法在特定环境下存在的不足。归纳总结了前人在该领域中的工作成果与得失。开题报告体现出研究内容丰富和目标明确，给出的研究方案，技术路线合理，具有较强的可实施性。开题报告给出的毕业设计各个环节的进度安排时间节点合理，同意开题。</p> <p style="text-align: right;">指导教师（签名）_____</p> <p style="text-align: right;">2022 年 3 月 11 日</p>
<p>答辩小组 意见</p>	<p style="text-align: center;"><input type="checkbox"/>通过 <input type="checkbox"/>不通过</p> <p>答辩组成员（签名）_____</p> <p>答辩组组长（签名）_____</p> <p style="text-align: right;">2022 年 3 月 14 日</p>
<p>学院审核 意见</p>	<p style="text-align: right;">分管教学院领导签字（公章）_____</p> <p style="text-align: right;">年 月 日</p>

西南科技大学本科毕业设计（论文）中期检查表

学 院	计算机科学与技术学院	专 业	软件工程	班 级	软件 1804
姓 名	肖劲涛	学 号	5120184509	指导教师	苏波
设计（论文）题目	快速傅里叶变换的并行算法研究及实现				
开题以来的进展情况和已取得的阶段性成果	<p>串行部分：</p> <ol style="list-style-type: none">1. Naive FFT/iFFT,2. Cooley-Tukey/Cooley-Tukey-r,3. r2c/c2r,4. convolve/correlate,5. 2d FFT6. radix-2/3/5,7. srFFT,8. PFA,9. Hartley Transform. <p>并行部分：</p> <ol style="list-style-type: none">1. Cooley-Tukey using CUDA on GPU2. Cooley-Tukey using MPI on CPU				
存在的问题及解决思路	<ol style="list-style-type: none">1. 基-2/3/5 分裂基运行时要求数组长度必须为 2/3/5 的 n 次方，因此在对比时无法精确对比其运行时间（runtime）。2. r2c 在运行时即现通过实数构造复数数组，之后再调用其他 FFT 实现，因此再复杂度上需要额外的时间与空间。3. 2d FFT 的时间难以与 1d FFT 比较，但可以同 PFA（也是一种二维变换）比较。4. MPI 实现内存共享需要通过拷贝时间，因此具有额外的时间开销。				
下一阶段的工作计划和研究内容	<ol style="list-style-type: none">1. 添加并行部分内容。2. 查找其他高效的串行 FFT 算法，并实现。3. 对比不同的算法，再不同平台下的性能，例如 x86-64, linux-amd64。并给出选择与优化建议。4. 完成论文。				

指导教师 意见	<div><input type="checkbox"/>通过<input type="checkbox"/>不通过</div> <div>指导教师(签名)_____</div> <div>2022 年 5 月 7 日</div>
学院审核	<div><input type="checkbox"/>通过<input type="checkbox"/>不通过</div> <div>审 核 人 （ 签 名 ）</div> <div>_____</div> <div>2022 年 5 月 11 日</div>

西南科技大学本科毕业设计（论文）指导教师指导记录表

指导老师姓名	苏波	职称	讲师(高校)	工作单位	计算机科学与技术学院
学生姓名	肖劲涛	学号	5120184509	专业	软件工程
设计（论文）题目:快速傅里叶变换的并行算法研究及实现					
指导时间	指导内容				
2022-01-03	对题目进行探讨，讲解内容和要求。				
2022-01-27	开题报告与项目调查研究指导。				
2022-02-11	检查开题报告完成情况。				
2022-02-25	检查项目设计的可行性。				
2022-03-04	指导开题答辩，并给出答辩建议。				
2022-03-25	就开题答辩情况与未来的项目设计提出指导意见。				
2022-04-22	检查项目设计情况，并给出修改建议。				
2022-05-13	检查项目设计，指导论文写作。				
2022-05-20	督促论文收尾，提出致谢部分内容添加与修改建议。				
2022-05-23	指导论文目录与引用排版。				
2022-05-27	指导论文查重与格式修改。				
2022-06-01	检查论文最终完成情况，并给出答辩的建议。				

备注：1.本表由学生填写，指导教师确认。2.指导教师应对学生毕业设计（论文）工作的进展情况一般每周至少进行一次检查、指导,并要求学生在表中记录检查、指导的实际内容。3.本表格不够可自行扩页。

学生(签名)_____

指导教师(签名)_____

西南科技大学本科毕业设计（论文）指导教师审阅意见表

（理工科用表）

学生姓名	肖劲涛	学 号	5120184509	专 业	软件工程		
设计（论文）题目	快速傅里叶变换的并行算法研究及实现						
评价项目	具体要求（A 级标准）	最高分	评 分				
			A	B	C	D	E
选题质量	选题符合专业培养目标，体现综合训练基本要求；题目有一定难度；有一定的理论意义或实际价值。	20	19-20	17-18	15-16	13-14	≤12
			19				
文献资料应用能力	能独立查阅文献；能正确翻译外文资料；具有收集、加工各种信息及获取新知识的能力。	10	10	9	8	7	≤6
			9				
研究能力	能较好地理解课题任务；研究方案设计合理；实验方法科学；理论分析与计算正确，实验数据准确可靠；有较强的动手能力、分析能力和实验数据处理能力；能综合运用所学知识发现与解决实际问题，得出有价值的结论。	20	19-20	17-18	15-16	13-14	≤12
			18				
创新能力	有创新意识，或对前人的工作有改进或突破，或设计（论文）有独到见解。	10	10	9	8	7	≤6
			9				
设计（论文）格式	设计（论文）格式、图表（或图纸）规范，符合要求。	10	10	9	8	7	≤6
			9				
设计（论文）质量	设计（论文）结构严谨，逻辑性强；语言文字表达准确流畅；有一定的学术价值或实用价值。	20	19-20	17-18	15-16	13-14	≤12
			18				
工作量及工作态度	工作量饱满；能圆满完成任务书规定的各项工作；工作认真、努力，遵守纪律，工作作风严谨务实；团队协作能力强。	10	10	9	8	7	≤6
			9				
总分	91	设计（论文）能否提交答辩：能（ ）否（ ）					
对设计（论文）的综合评语： 论文选题符合专业培养目标，达到了综合训练要求。通过学习快速傅里叶变换的基本理论以及并行算法的相关知识，本文提出了相应的快速傅里叶变换并行化方法，并应用多线程技术、MPI 技术和 GPGPU 技术分别实现了傅里叶变换的快速计算，对相关数据和实现技术进行了详实的分析和对比。论文成果有一定的理论意义和实际价值。该生在毕业设计期间，能按期完成规定的任务，积极与老师沟通，反馈问题及时。能独立查阅文献资料和进行相关的研究工作，能独立的翻译外文资料，工作认真努力，遵守纪律，工作量较饱满，达到本科毕业论文的要求，同意该同学参加论文答辩。 <div style="text-align: right;">年 月 日</div>							
指导教师（签名）		职 称	讲师(高校)	工作单位	计算机科学与技术学院		

西南科技大学本科毕业设计（论文）评阅教师审阅意见表

（理工科用表）

学生姓名	肖劲涛	学 院	计算机科学与技术学院	专 业	软件工程				
设计（论文）题目	快速傅里叶变换的并行算法研究及实现								
评价项目	具体要求（A 级标准）	最高分	评 分						
			A	B	C	D	E		
选题质量	选题符合专业培养目标，体现综合训练基本要求；题目有一定难度；有一定的理论意义或实际价值。	20	19-20	17-18	15-16	13-14	≤12		
			18						
文献资料应用能力	能独立查阅文献；能正确翻译外文资料；具有收集、加工各种信息及获取新知识的能力。	10	10	9	8	7	≤6		
			9						
研究能力	能较好地理解课题任务；研究方案设计合理；实验方法科学；理论分析与计算正确，实验数据准确可靠；有较强的动手能力、分析能力和实验数据处理能力；能综合运用所学知识发现与解决实际问题，得出有价值的结论。	20	19-20	17-18	15-16	13-14	≤12		
			18						
创新能力	有创新意识，或对前人的工作有改进或突破，或设计（论文）有独到见解。	10	10	9	8	7	≤6		
			9						
设计（论文）格式	设计（论文）格式、图表（或图纸）规范，符合要求。	10	10	9	8	7	≤6		
			9						
设计（论文）质量	设计（论文）结构严谨，逻辑性强；语言文字表达准确流畅；有一定的学术价值或实用价值。	20	19-20	17-18	15-16	13-14	≤12		
			18						
工作量	工作量饱满；能圆满完成任务书规定的各项工作。	10	10	9	8	7	≤6		
			9						
总分	90	设计（论文）能否提交答辩：能（ ）否（ ）							
<p>对设计（论文）的综合评语：</p> <p>FFT 是工程中重要的工具，是将数据从时域向频域转换的重要手段，具有重要的应用价值。该论文立足于使用 GPU 来加速 FFT 转换过程，即满足了专业毕业培养要求，又具有社会应用价值，具有一定的应用价值。</p> <p>论文格式规范，逻辑清晰，文字表达较为流畅，工作量饱满。</p> <p>同意参加答辩。</p> <p>年 月 日</p>									
评阅教师（签名）		职 称	副教授	工作单位	计算机科学与技术学院				

西南科技大学本科毕业设计（论文）答辩记录及评价表

答辩人姓名	肖劲涛	学院	计算机科学与技术学院	学号	5120184509	专业	软件工程
设计（论文）题目		快速傅里叶变换的并行算法研究及实现					
答辩记录	<p>教师主要提问记录 1、快速傅里叶变换在具体的工程中有哪些运用？ 2、使用的多种技术之间有何差异，各有什么优势？ 3、论文实现的并行算法与现有的并行算法有何差异？ 学生回答问题情况 1、以图像处理为例，傅里叶变换将图像转变为频域信号，物体的轮廓由低频信号表示，物体的细节由高频信号表示。通过去除高频信号可以实现对物体的轮廓提取。在音频处理中，可以通过变换后对频域信号的处理实现调音等功能。 2、多线程编程可以实现并发的 FFT，理论上由于是一个进程之间的通信，可以减少不同线程之间信息传递的开销，但虚拟线程往往不能被绑定到物理线程上，因而无法实现真正的并行；MPI 则是从进程的层面上实现并行算法，与多线程相比其优点在于实现了真正的并行，而缺点则是并行之间的通信开销比较大；CUDA 或者说 GPGPU 技术则是从 GPU 的层面实现并行算法，由于流处理器节点数量多，因而适合于大规模的任务，但是由于缺少控制器，因而对于复杂的计算较为无力。 3、以论文中的 CUDA_FFT 和 CUDA 库提供的 CUFFT 函数为例，CUDA_FFT 在面对小规模数据集的时候有着明显优势，CUFFT 则有一定的系统开销；然而随着数量规模的增大，CUFFT 的运算速度上升的趋势明显小于 CUDA_FFT。在面对大规模数据集时，CUFFT 有着更好的表现，而 CUDA_FFT 在小规模数据集上效果更佳。</p>						
评价项目	具体要求（A 级标准）	最高分	评 分				
			A	B	C	D	E
设计（论文）质量、水平	设计（论文）结构严谨，逻辑性强；有一定的学术价值或实用价值；文字表达准确流畅；论文格式规范；图表（或图纸）规范、符合要求。	50	46-50	41-45	36-40	31-35	≤ 30
			45.5				
设计（论文）报告、讲解	思路清晰；概念清楚，重点（创新点）突出；语言表达准确；报告时间、节奏掌握好。	20	19-20	17-18	15-16	13-14	≤ 12
			18.25				
答辩情况	回答问题有理有据，基本概念清楚；主要问题回答准确、有深度。	30	28-30	25-27	22-24	19-21	≤ 18
			27				
总分			90.8				

答辩组评语:

围绕选题背景、算法设计思路、算法具体实施与结论四个方面进行了详细说明。毕业设计使用了多个模型、多种技术完成了在不同平台下的傅里叶变换算法，主要实现了从频域信号到时域信号的转变。完成了多种不同的傅里叶算法的对比，运用了多种技术实现了不同的并行傅里叶算法，同时利用 CUDA 实现了 GPU 上的快速傅里叶变换。讲述了算法设计的思路、可行性以及具体实现方式，讲解了技术的难点与关键点。并正确回答了老师的提问。

答辩组成员（签名）_____ 答辩组组长（签名）_____

年 月 日

西南科技大学本科毕业设计（论文）成绩评定表

题 目	快速傅里叶变换的并行算法研究及实现				
学生姓名	肖劲涛	学号	5120184509	专业班级	软件 1804
成绩汇总	评分项目	评分	比例（%）	分数	总成绩分数
	指导教师评分	91	40	36.4	91
	评阅教师评分	90	30	27.0	
	答辩小组评分	90.8	30	27.2	
成绩等级 结论	优				
是否同意 毕业设计 （论文） 通过	<div><input type="checkbox"/>同意</div> <div><input type="checkbox"/>不同意 (<input type="checkbox"/>论文重新修改 <input type="checkbox"/>论文重新答辩)</div>				
毕业设计 （论文） 领导小组 推优评语	创新层面，目前对于快速傅里叶变换的并行算法研究大多数注重于对大规模数据的加速，以及并行理论的研究。该生在对不同的模型进行比较的基础上，开展了针对包括 Arm 嵌入式和高性能 GPU 在内的不同平台的对比试验。理论层面，该文章通过对并行模型的特点分析，论证了快速傅里叶变换的可行性，同时给出了以何种方式、在什么时候实施什么规模的并行计算。文章撰写方面，行文流畅、用词准确，文章整体逻辑性强，工作量饱满。综上所述，特此推荐为本届优秀毕业论文。				
学院答辩委员会主任签字： <div>年 月 日</div>					