

FFT Library

Yuzhi Zhao

October 19, 2017

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	General Information	1
3.1	Purpose	1
3.2	Scope	1
3.3	Overview of Document	1
4	Plan	1
4.1	Software Description	1
4.2	Test Team	2
4.3	Automated Testing Approach	2
4.4	Verification Tools	2
4.5	Non-Testing Based Verification	2
5	System Test Description	2
5.1	Tests for Functional Requirements	2
5.1.1	Calculation Test	2
5.1.2	Loading Library Test	5
5.2	Tests for Nonfunctional Requirements	6
5.2.1	Speed Comparation Test	6
5.3	Traceability Between Test Cases and Requirements	6
6	Unit Testing Plan	6
7	Appendix	7
7.1	Symbolic Parameters	7
7.2	Usability Survey Questions?	7

List of Tables

List of Figures

This document ...

3 General Information

The following section provides an overview of the Verification and Validation (V & V) Plan for a FFT library.

3.1 Purpose

The main purpose of this document is to describe the verification and validation process that will be used to test a FFT Library. This document is intended to be used as a reference for all future testing and will be used to increase confidence in the software implementation.

This document will be used as a starting point for the verification and validation report. The test cases presented within this document will be executed and the output will be analyzed to determine if the library is implemented correctly.

3.2 Scope

The whole library includes four FFT or IFFT calculation functions. All tests should be applied based on this scope.

3.3 Overview of Document

The following sections provides more details about the V&V of a FFT Library. Information about verification tools, automated testing approaches will be stated. And test cases for all system testing and part of unit testing will be provided.

4 Plan

4.1 Software Description

The software being tested is a library for FFT algorithm. Users choose different FFT or IFFT functions and give proper input datas to complete a

FFT or IFFT calculation. The library includes radix-2 and radix-3 FFT (and IFFT) calculation functions.

4.2 Test Team

Yuzhi Zhao

4.3 Automated Testing Approach

A unit testing framework will be implemented in both unit testing and system testing.

Script will be used to call all the test cases in test suite.

Test coverage analysis will be applied to measure code coverage.

Compiler can do syntax check automatically.

4.4 Verification Tools

1. Cutest as unit testing framework
2. Make as script to call test cases and execute test
3. Xcover as coverage analysis tool

4.5 Non-Testing Based Verification

Symbolic Execution

Because FFT library is based on a mathematical expression. Using Symbolic Execution can trace the path and the result can be compared with mathematical expression directly.

5 System Test Description

5.1 Tests for Functional Requirements

5.1.1 Calculation Test

Radix-2 Complex Number Calculation Function

1. Radix-2 Complex Number FFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Automated. For validation purpose, datas should also be compared with results from normal DFT calculations manually.

2. Radix-2 Complex Number IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Same as above.

Radix-2 Real Number Calculation Function

1. Radix-2 Real Number FFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Same as above.

2. Radix-2 Real Number IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Same as above.

Radix-3 Complex Number Calculation Function

1. Radix-3 Complex Number FFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Same as above.

2. Radix-3 Complex Number IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Same as above.

Radix-3 Real Number Calculation Function

1. Radix-3 Real Number FFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

2. Radix-3 Real Number IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: input.txt, expectedOutput.txt

Output: output.txt, TestResult: pass or not pass

How test will be performed:

Same as above.

5.1.2 Loading Library Test

Under Win X86 platform

Type: Functional, Dynamic, Manual

Initial State: None

Input: input.txt

Output: output.txt

How test will be performed: Manual, automated

Under Mac OS platform

Type: Functional, Dynamic, Manual

Initial State: None

Input: input.txt

Output: output.txt

How test will be performed: Manual, automated

5.2 Tests for Nonfunctional Requirements

5.2.1 Speed Comparison Test

Compare Calculation Speed with DFT calculation

Type: Dynamic, automated, Manual

Initial State: None

Input: input.txt

Output: Time

How test will be performed:

Manually compare the time with the time using DFT Library.

5.3 Traceability Between Test Cases and Requirements

6 Unit Testing Plan

[Unit testing plans for internal functions and, if appropriate, output files
—SS]

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

This is a section that would be appropriate for some teams.