

FFT Library

Yuzhi Zhao

October 18, 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	3
5.1	Tests for Functional Requirements	3
5.1.1	Area of Testing1	3
5.1.2	Area of Testing2	4
5.2	Tests for Nonfunctional Requirements	4
5.2.1	Area of Testing1	4
5.2.2	Area of Testing2	5
5.3	Traceability Between Test Cases and Requirements	5
6	Unit Testing Plan	5
7	Appendix	6
7.1	Symbolic Parameters	6
7.2	Usability Survey Questions?	6

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 provide 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 data 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 (functions called by calculation functions) and system testing (four main calculation functions). Because a unit testing framework is usually being used to test the individual function or procedure. As for FFT Library, the four calculation functions can be considered as individual functions each.

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

Test coverage analysis will be carried out to measure code coverage.

[\[probablyly some static check —SS\]](#)

4.4 Verification Tools

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

[\[Thoughts on what tools to use, such as the following: unit testing framework, valgrind, static analyzer, make, continuous integration, test coverage tool, etc. —SS\]](#)

4.5 Non-Testing Based Verification

Not applicable. [\[List any approaches like code inspection, code walkthrough, symbolic execution etc. Enter not applicable if that is the case. —SS\]](#)

5 System Test Description

5.1 Tests for Functional Requirements

5.1.1 Area of Testing1

Calculation Test

1. Radix-2 Complex Number FFT And IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: Test case file

Output: File with output data

How test will be performed:

This test would be mostly done automatedly by using "Make". But after getting the output datas, the datas should be compared with results from other FFT libraries manually. And for validation purpose, datas should also be compared with results from normal DFT calculations manually.

2. Radix-3 Complex Number FFT And IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: Test case file

Output: File with output data

How test will be performed:

Same as above.

3. Radix-2 Real Number FFT And IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: Test case file

Output: File with output data

How test will be performed:
Same as above.

4. Radix-3 Real Number FFT And IFFT Calculation Function

Type: Functional, Dynamic, Automated, Manual

Initial State: None

Input: Test case file

Output: File with output data

How test will be performed:
Same as above.

5.1.2 Area of Testing2

Library Function Calling Test

1. Loading library Under Different Versions Of Compilers of Different Language

Type: Functional, Dynamic, Manual

Initial State: None

Input: Test case file

Output: File with output data

How test will be performed:

5.2 Tests for Nonfunctional Requirements

5.2.1 Area of Testing1

Title for Test

1. Loading library Under Different Versions Of Compilers of Different Language

Type: Functional, Dynamic, Manual

Initial State: None

Input: Test case file

Output: File with output data

How test will be performed:

2. test-id2

Type: Functional, Dynamic, Manual, Static etc.

Initial State:

Input:

Output:

How test will be performed:

5.2.2 Area of Testing2

...

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.