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

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 verfication tools, automatted 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 iproper input datas to complete a

FFT or IFFT calcualtion. 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.
[probablely some static check —SS]

4.4 Verification Tools

1. Unittest as unit testing framework
2. Make as script to call test cases
3. Coverage.py 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

[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

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

5.1.2 Area of Testing2

...

5.2 Tests for Nonfunctional Requirements

5.2.1 Area of Testing1

Title for Test

1. test-id1

   Type:

Initial State:

Input/Condition:

Output/Result:

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.