IEEE 754 FFT Core Development Kit

designed by

Koppsoft

for

1 Introduction

1.1 Overview

The IEEE 754 FFT Core Development Kit (DK) is a tool to assist users of the 4DSP IEEE 754 FFT IP core (1D and 2D, forward and inverse). It provides a bit-true software model plus various other tools such as data generation, file conversion, and file comparison.

1.2 References

1. FM480 Virtex-4 PMC-X
2. IEEE 754 FFT core user manual

2 Installation

2.1 Windows 2000 / XP

The Microsoft .NET framework is required in order to run the 4DSP IEEE 754 FFT DK under Windows 2000 or XP. If not already installed, the .NET framework can be downloaded free of charge from:

http://www.microsoft.com/downloads

It is also available via Microsoft Updates (select custom install, Software, Optional).

Unzip the FFT4DSPDK.zip file. It contains only one executable. Place this in the directory where your test data exists. The DK works only on data within the same directory as the executable.

2.2 Linux

Mono is required to run the DK under Linux. If not already installed, Mono can be downloaded free of charge from:

http://www.mono-project.com/Downloads

The easiest method is to use the Installer for which detailed instructions are provided.

Alternatively, the source code can be downloaded and compiled.

Unpack the FFT4DSP.rar file. It contains only one executable. Place this in the directory where your test data exists. The DK works only on data within the same directory as the executable.

3 Overview

The DK permits data generation, bit-true 1D and 2D FFT, file comparison, and in the Windows version, file conversion.

3.1 Data Formats

All data files are ASCII and store the data one complex sample per two lines. Output data is always complex. For example the number -1.079066 + i0.8934436 would be represented in the *.float.txt files by:

-1.079066
0.8934436

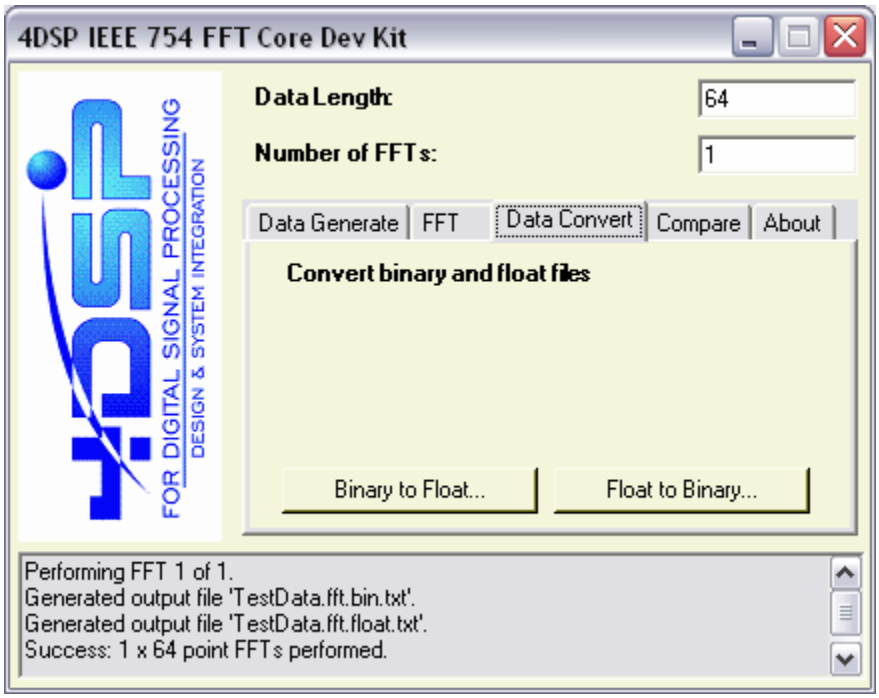
And in the *.bin.txt files by:

10111111100010100001111011010100
00111111011001001011100010111001

The *.bin.txt files are used by the test bench and the DK's FFT model. The *.float.txt files allow easy export of the data to applications such as Mathworks MATLAB.

3.1.1 Data Format Conversion

(Windows DK only) It is possible to covert between the two data file formats by using the 'Data Convert' tab of the DK. Clicking either button, the first file dialog asks for the source file, the second allows you to specify the name to save the file as.



3.2 Generating Data

For basic testing it is possible to generate three sets of test data. This data can then be used with the 4DSP IEEE 754 FFT test bench and the DK's own bit-true model. Data types are:

- Sin(x) / x
- Random
- Dirac

The test bench also requires a parameter file. This is generated with the test data and can be set for forward or inverse FFT direction.

The data generator requires the FFT direction (inverse or forward), a data (FFT) length (i.e. between 64 and 1M points and be a power of 2), and the number of FFTs. The number of FFTs * Data Length must not exceed 1M (= 1K x 1K).

Five files will be generated. The generated data files are:

- TestData.float.txt

- TestData.bin.txt
- Twiddles.float.txt
- Twiddles.bin.txt

The parameters file is:

- TestData.params.txt

3.3 FFT

The FFT is always performed on the data in file 'TestData.bin.txt'. FFT direction can be forward or inverse and 1D and 2D FFTs are supported.

Two files are generated:

- TestData.fft.bin.txt
- TestData.fft.float.txt

3.3.1 1D FFT

Input parameters include direction (inverse or forward), FFT length (between 64 and 1M point and power of 2), and number of FFTs (loops). For a 1D FFT, number of FFTs * FFT length must not exceed 1M (= 1K x 1K).

3.3.2 2D FFT

Input parameters include direction (inverse or forward), FFT side length (between 64 and 1024 point and power of 2), and number of FFTs (loops). For a 2D FFT, number of FFTs * side length * side length must not exceed 1M (= 1K x 1K).

3.4 Compare

Compares the files TestData.fft.bin.txt' and 'TestBenchData.fft.bin.txt' and places the results in 'ComparisonReport.txt' and 'ComparisonReport.bin.txt'. Both reports are identical except in terms of presentation of results.

3.4.1 ComparisonReport.txt - Example

0: (-1.079066, 0.8934436i) != (-1.079065, 0.8934436i) (Difference = (1, 2.170702E-07i))
FFT 0: SNR = 36.1236

The first line lists the sample number in which a difference was found, followed by the source (x) and destination (y) values and the ratio between them (x/y).

The next line shows the FFT number in which the difference occurred and the signal to noise ratio. Important: This is based on the 'Data Length' that is currently displayed at the top of the DK.

3.4.2 ComparisonReport.bin.txt - Example

```
0
10111111100010100001111011010100 00111111011001001011100010111001
10111111100010100001111011010000 00111111011001001011100010111001
```

The first line is the sample number, the second line is the number from the source file, and the third value is from the destination file.

4 Running – Windows

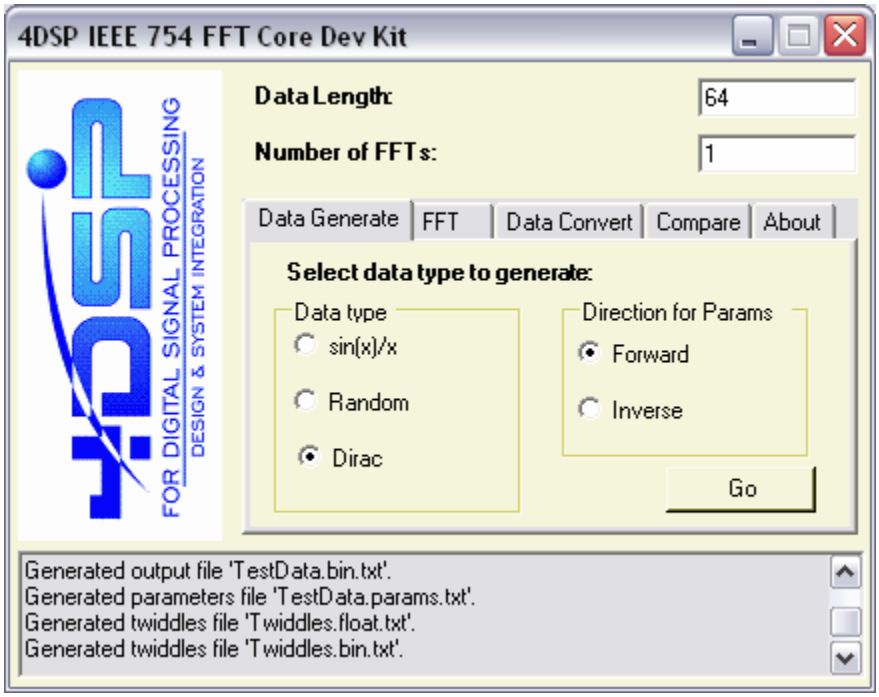
See chapter 3 for details of the described functionality.

4.1 Start

Double click the FFT4DSP.exe.

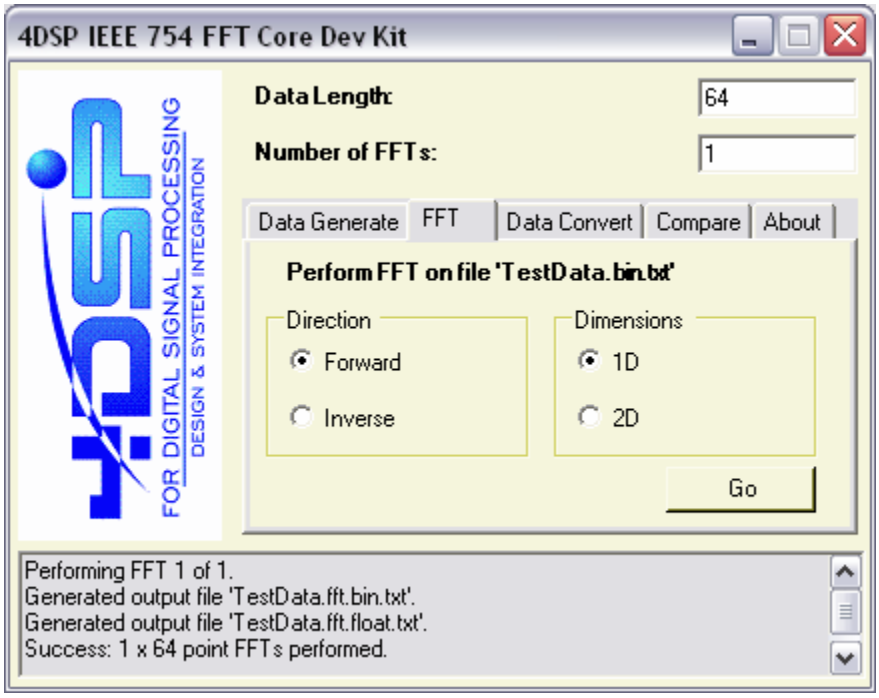
4.2 Generating Data

1. Click the 'Data Generate' tab of the DK.
2. Select the 'Data Type'
3. Select the 'Direction"
4. Enter a valid 'Data Length'.
5. Enter 'Number of FFTs'.
6. Click 'Go'



4.3 FFT

Via the FFT tab of the DK you can access the bit-true software model.



The FFT is always performed on the data in file 'TestData.bin.txt'. FFT direction can be forward or inverse and 1D and 2D FFTs are supported.

1. Select the 'Direction'
2. Select 'Dimensions'
3. Enter a valid 'Data Length' or 'Side Length'.
4. Enter 'Number of FFTs'.
5. Click 'Go'

4.4 Compare

The 'Compare' tab of the DK compares the files TestData.fft.bin.txt' and 'TestBenchData.fft.bin.txt' and places the results in 'ComparisonReport.txt' and 'ComparisonReport.bin.txt'.

5 Running - Linux

See chapter 3 for details of the described functionality.

5.1 Start

Linux uses a command line version of the DK is FFT4DSP.exe. Start with:

mono FFT4DSP.exe *params*

This DK has four top level modes:

- Data generate
- FFT1D
- FFT2D
- Compare

5.2 Generating Data

The flag –datagen invokes the data generator.

Usage:
mono FFT4DSP.exe –datagen –sin|-random|-dirac –N=<length> [-loops=<noLoops>] [-inv] [-v]

Obligatory parameters are:

Type	–sin, -random, or –dirac.
-N	FFT Length

Optional parameters are:

-loops=<noLoops>	Number of FFTs
-inv	Inverse FFT (default = forward)
-v	Verbose (Prints loops numbers)

5.3 FFT 1D

The flag –fft1d invokes the FFT 1D mode.

Usage:
mono FFT4DSP.exe –fft1d –N=<length> [-loops=<noLoops>] [-inv] [-v]

Obligatory parameters are:

-N	FFT Length

Optional parameters are:

-loops=<noLoops>	Number of FFTs
-inv	Inverse FFT (default = forward)
-v	Verbose (Prints loops numbers)

5.4 FFT 2D

The flag –fft2d invokes the FFT 2D mode.

Usage:
mono FFT4DSP.exe –fft2d –N=<sideLength> [-loops=<noLoops>] [-inv] [-v]

Obligatory parameters are:

-N	Side Length

Optional parameters are:

-loops=<noLoops>	Number of FFTs
-inv	Inverse FFT (default = forward)
-v	Verbose (Prints loops numbers)

5.5 Compare

The flag –cmp invoke the comparison mode.

5.6 Help

The flag –h provides usage information.