NASoftware Limited

Incorporating InfoSAR

# ARM DSP report: VSIPL FFT Benchmarks. Version 1.0

N.*A. Software Ltd*

*29th February 2020*

# 1. Introduction.

NA Software (NAS) sells low level DSP libraries to board manufacturers and directly to companies within the defence and aerospace industry. The best selling NAS API is the Vector Signal Processing Library (VSIPL) on PowerPC and Intel platforms. In 2018, NAS also developed a version of our VSIPL DSP library for a range of ARM platforms. These platforms include the ARM A53, A57 and A72 chips. Our ARM VSIPL library was further optimised within 2019 and is now a stable reliable software product.

The aim of this report is to compare our ARM VSIPL library on a number of ARM platforms to ARM Performance Libraries as provided by the ARM Corporation in the following link:

ARM Performance Libraries

In carrying out this comparison NAS studied the following range of FFT operations with vector and matrix data as given in the table below:

**Table 1: VSIPL operations.**

| Routine | Benchmark Parameters. |
|---|---|
| 1D FFT with Vectors, complex-to-complex, in-place | 32, 64, 128, 256, 1K, 4K, 16K, 256K, 512K |
| 1D FFT with Vectors, complex-to-complex, out-of-place | 32, 64, 128, 256, 1K, 4K, 16K, 256K, 512K |
| Multiple FFT with Square Matrices, complex-complex, in-place. | 32x32, 64x64, 128x128, 256x256, 512x512, 1Kx1K. [rows x columns] |
| Multiple FFT with Square Matrices, complex-complex, out-of-place. | 32x32, 64x64, 128x128, 256x256, 512x512, 1Kx1K. [rows x columns] |
| Multiple FFT with Non-Square Matrices, complex-to-complex, in-place. | 1Kx100, 4Kx50, 16Kx20, 64Kx20 128Kx20 [rows x columns] |
| Multiple FFT with Non-Square Matrices, complex-to-complex, out-of-place. | 1Kx100, 4Kx50, 16Kx20, 64Kx20 128Kx20 [rows x columns] |
| 2D FFT with Square Matrices, complex-to-complex, in-place. | 32x32, 64x64, 128x128, 256x256, 512x512, 1Kx1K. [rows x columns] |
| 2D FFT with Square Matrices, complex-to-complex, out-of-place. | 32x32, 64x64, 128x128, 256x256, 512x512, 1Kx1K. [rows x columns] |
| 2D FFT with Non-Square Matrices, complex-to-complex, in-place. | 1Kx100, 4Kx50, 16Kx20, 64Kx20 128Kx20 [rows x columns] |
| 2D FFT with Non-Square Matrices, complex-to-complex, out-of-place. | 1Kx100, 4Kx50, 16Kx20, 64Kx20 128Kx20 [rows x columns] |

All the above lengths are given in units of complex cells.

In this document we are going to study the above FFT operations and report on the:

1. Performance with in-place and out-of-place data;
2. Performance with Vector and Matrix data;
3. Performance of 1D, Multiple 1D and 2D FFT operations;
4. Performance with Square and Non-Square Matrix data;

The three test platform details are as follows:

**Table 2: Test Platform Information.**

| | | |
|---|---|---|
| Platform 1. | Type: | ARM Cortex-A53 |
| | Platform: | Odroid |
| | Bit: | 64 |
| | Operating Frequency: | 1.5 GHz. |
| | SIMD Instruction set: | ARMv8 |
| | Operating System: | Linux. |
| | Physical Cores: | 4 |
| Platform 2. | Type: | ARM Cortex-A57 |
| | Platform: | Nvidia Jetson TX2 |
| | Bit: | 64 |
| | Operating Frequency: | 2.0 GHz. |
| | SIMD Instruction set: | ARMv8 |
| | Operating System: | Linux |
| | Physical Cores: | 4 |
| Platform 3. | Type: | ARM Cortex-A72 |
| | Platform: | NXP LX2160A |
| | Bit: | 64 |
| | Operating Frequency: | 2.0 GHz. |
| | SIMD Instruction set: | ARMv8 |
| | Operating System: | Linux |
| | Physical Cores: | 4 |

The project version information is as follows:

**Table 3: Version Information.**

| | |
|---|---|
| A53 NAS VSIPL Serial Library | Version 4.19.8 |
| A57 NAS VSIPL Serial Library | Version 4.19.8 |
| A72 NAS VSIPL Serial Library | Version 4.19.8 |
| ARM Performance Library | Version  20.2.0 |

This report is organised into the following sections:

**Section 1: Introduction** – this introduction.

**Section 2: VSIPL FFT Performance with Vector Data** -
Compares the 1D FFT vector performance of the NAS VSIPL library to the ARM performance library on the A53, A57 and A72 test systems.

**Section 3: VSIPL Multiple FFT Performance with Square Matrix Data** -
Compares the multiple 1D FFT performance with a set of square matrices.

**Section 4: VSIPL Multiple FFT Performance with Non-Square Matrix Data** -
Compares the multiple 1D FFT performance with a set of non-square matrices.

**Section 5: VSIPL 2D FFT Performance with Square Matrix Data** -
Compares the 2D FFT performance with a set of square matrices.

**Section 6: VSIPL 2D FFT Performance with Non-Square Matrix Data** -
Compares the 2D FFT performance with a set of non-square matrices.

**Section 7: Conclusions** – gives conclusions.

## 2. VSIPL FFT Performance with Vector Data.

In this section we investigate the performance of the 1D complex-to-complex FFT over a range of vector lengths with in-place and out-of-place operations. The range of vector lengths below are 32, 64, 128, 256, 1K, 4K, 256K and 512K. All vector lengths are given in units of complex cells.

### *2.1. 1D in-place complex-to-complex FFT.*

The tables below shows the benchmark figures in microseconds for our A53, A57 and A72 platforms. Each table shows the time for the NAS function vsip_ccfftip_f and the equivalent set of FFTW functions in the ARM performance libraries. These functions performs a 1D complex-to-complex in-place FFT operation.

#### 2.1.1. A53 System Performance with Vector Data.

**Table 4: A53 vsip_ccfftip_f with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *NAS* | 0.35 | 0.64 | 1.69 | 3.32 | 15.96 | 128.62 | 628.87 | 37477.00 | 73340.80 |
| *ARM* | 0.80 | 1.48 | 2.82 | 5.78 | 54.14 | 327.90 | 3633.57 | 135333.00 | 283998.00 |

#### 2.1.2. A57 System Performance with Vector Data.

**Table 5: A57 vsip_ccfftip_f with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *NAS* | 0.27 | 0.36 | 1.18 | 1.97 | 10.33 | 65.65 | 328.24 | 12933.00 | 36968.20 |
| *ARM* | 0.37 | 1.00 | 2.04 | 4.32 | 32.26 | 167.63 | 2085.67 | 98012.60 | 222667.00 |

#### 2.1.3. A72 System Performance with Vector Data.

**Table 6: A72 vsip_ccfftip_f with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *NAS* | 0.27 | 0.35 | 0.97 | 1.88 | 9.67 | 53.05 | 306.79 | 9168.94 | 21059.20 |
| *ARM* | 0.34 | 0.91 | 1.92 | 3.99 | 23.79 | 154.43 | 1016.93 | 55185.90 | 154189.00 |

The A72 and the A57 have a similar performance for short FFT lengths. The A72 has an advantage over the A57 for larger FFT lengths. The A53 has the worst performance. However, the A53 is only clocked at 1.5 GHz where as both the A57 and A72 are clocked at 2.0 GHz. The NAS library has a better performance over all data lengths.

## *2.2. 1D out-of-place complex-to-complex FFT.*

The tables below shows the benchmark figures for the 1D complex-to-complex out-of-place FFT operation. Each table shows the benchmarks for our A53, A57 and A72 platforms in microseconds. The NAS out-of-place FFT operation is carried out by a call to the function vsip_ccfftop_f.

### 2.2.1. A53 System Performance with Vector Data.

**Table 7: A53 vsip_ccfftop_f with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *NAS* | 0.36 | 0.63 | 1.62 | 3.88 | 17.27 | 123.69 | 668.21 | 39012.80 | 76573.90 |
| *ARM* | 1.22 | 1.52 | 2.91 | 6.15 | 58.88 | 375.84 | 4411.66 | 132130.00 | 280565.00 |

### 2.2.2. A57 System Performance with Vector Data.

**Table 8: A57 vsip_ccfftop_f with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *NAS* | 0.29 | 0.37 | 1.34 | 1.99 | 10.68 | 60.91 | 340.17 | 15246.40 | 35880.50 |
| *ARM* | 0.36 | 0.98 | 2.00 | 4.53 | 30.31 | 182.17 | 1035.71 | 90029.90 | 216526.00 |

### 2.2.3. A72 System Performance with Vector Data.

**Table 9: A72 vsip_ccfftop_f with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *NAS* | 0.27 | 0.36 | 0.90 | 1.97 | 10.21 | 57.13 | 311.11 | 9649.18 | 21041.90 |
| *ARM* | 0.34 | 0.91 | 1.92 | 4.48 | 25.02 | 152.93 | 1163.19 | 57157.20 | 150243.00 |

These performance figures are similar to the in-place complex-to-complex results. Again, the NAS library has a performance advantage over all data lengths for the A53, A57 and A72 systems.

# 3. VSIPL Multiple FFT Performance with Square Matrix Data.

In this section we investigate the performance of the complex-to-complex multiple FFT with square matrices over a range of data sizes. The data sizes studied below are 32 by 32, 64 by 64, 128 by 128, 256 by 256, 512 by 512 and 1K by 1K with each matrix data size given in units of complex cells.

## *3.1. Multiple 1D in-place complex-to-complex FFT.*

The tables below shows the benchmark figures in microseconds for our A53, A57 and A72 platforms. Each table shows the time for the NAS function vsip_ccfftmip_f and the equivalent set of FFTW functions in the ARM performance libraries. These functions performs a multiple 1D complex-to- complex in-place FFT operation along each row in the matrix.

### 3.1.1. A53 System Performance with Square Matrix Data.

**Table 10: A53 vsip_ccfftmip_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 8.96 | 39.94 | 204.65 | 1153.88 | 6494.74 | 31887.40 |
| *ARM* | 27.77 | 117.48 | 589.48 | 4544.35 | 22330.20 | 101000.00 |

### 3.1.2. A57 System Performance with Square Matrix Data.

**Table 11: A57 vsip_ccfftmip_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 6.07 | 22.71 | 125.30 | 554.32 | 2867.55 | 16070.30 |
| *ARM* | 12.95 | 73.96 | 318.72 | 1997.42 | 12381.18 | 776181.00 |

### 3.1.3. A72 System Performance with Square Matrix Data.

**Table 12: A72 vsip_ccfftmip_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 5.82 | 22.01 | 116.55 | 518.60 | 2403.71 | 10808.20 |
| *ARM* | 12.94 | 67.81 | 297.16 | 1477.62 | 7247.31 | 45233.40 |

The A72 system and the NAS library have the best performance.

### *3.2. Multiple 1D out-of-place complex-to-complex FFT.*

The following tables show the results with the out-of-place operation with the same set of square matrices:

## 3.2.1. A53 System Performance with Square Matrix Data.

**Table 13: A53 vsip_ccfftmop_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 9.67 | 43.07 | 218.04 | 1775.30 | 7259.34 | 33782.80 |
| *ARM* | 22.29 | 95.86 | 526.99 | 3765.06 | 19176.20 | 89973.50 |

## 3.2.2. A57 System Performance with Square Matrix Data.

**Table 14: A57 vsip_ccfftmop_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 5.62 | 24.10 | 125.34 | 600.69 | 4202.86 | 20259.60 |
| *ARM* | 10.33 | 63.61 | 278.70 | 1556.09 | 10485.40 | 48512.30 |

## 3.2.3. A72 System Performance with Square Matrix Data.

**Table 15: A72 vsip_ccfftmop_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 5.43 | 23.11 | 119.13 | 569.85 | 2727.81 | 14343.80 |
| *ARM* | 10.01 | 59.05 | 258.50 | 1323.77 | 6493.43 | 41483.30 |

The NAS library is quicker with in-place operations than with out-of-place operations. The ARM performance library is quicker with out-of-place operations than in-place. However, the NAS library is substantially quicker than the ARM performance library for both in-place and out-of-place operations in the above examples.

# 4. VSIPL Multiple FFT Performance with Non-Square Matrix Data.

In this section we investigate the performance of the complex-to-complex multiple FFT with non-square matrices over a range of data sizes. The non-square matrix sizes studied below are 1K by 100, 4K by 50, 16K by 20, 64K by 20 and 128K by 20 in units of complex cells.

## *4.1. Multiple 1D in-place complex-to-complex FFT.*

The tables below show the benchmark figures in microseconds for our A53, A57 and A72 platforms. Each table shows the time for the NAS function vsip_ccfftmip_f and the equivalent set of FFTW functions in the ARM performance libraries. These functions performs a multiple 1D complex-to-complex in-place FFT operation along each row in the non-square matrix.

### 4.1.1. A53 System Performance with Non-Square Matrix Data.

**Table 16: A53 vsip_ccfftmip_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 2852.26 | 9574.60 | 18773.90 | 125083.00 | 423862.00 |
| *ARM* | 10628.00 | 32480.90 | 95919.40 | 624606.00 | 1311390.00 |

### 4.1.2. A57 System Performance with Non-Square Matrix Data.

**Table 17: A57 vsip_ccfftmip_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 1090.23 | 3145.24 | 9152.27 | 53663.90 | 127232.00 |
| *ARM* | 5057.94 | 14735.40 | 37608.70 | 237040.00 | 776181.00 |

### 4.1.3. A72 System Performance with Non-Square Matrix Data.

**Table 18: A72 vsip_ccfftmip_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 1047.63 | 3021.41 | 6763.12 | 37928.80 | 97674.30 |
| *ARM* | 2852.20 | 9574.60 | 18773.90 | 125083.00 | 423862.00 |

The A72 system and the NAS library have the best performance.

### *4.2. Multiple 1D out-of-place complex-to-complex FFT.*

The following tables show the results with the out-of-place operation with the same set of non-square matrices:

## 4.2.1. A53 System Performance with Non-Square Matrix Data.

**Table 19: A53 vsip_ccfftmop_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 3306.15 | 10488.20 | 25511.30 | 131081.00 | 435479.00 |
| *ARM* | 8842.06 | 29768.80 | 83311.60 | 595969.00 | 1226380.00 |

## 4.2.2. A57 System Performance with Non-Square Matrix Data.

**Table 20: A57 vsip_ccfftmop_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 1338.27 | 5082.70 | 11058.90 | 54435.00 | 130160.00 |
| *ARM* | 4208.99 | 13371.80 | 31989.80 | 219867.00 | 544264.00 |

## 4.2.3. A72 System Performance with Non-Square Matrix Data.

**Table 21: A72 vsip_ccfftmop_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 1135.20 | 3224.17 | 8109.61 | 39559.50 | 97889.20 |
| *ARM* | 3101.26 | 9686.25 | 27128.10 | 174879.00 | 463647.00 |

Again, the A72 system and the NAS library have the best performance.

# 5. VSIPL 2D FFT Performance with Square Matrix Data.

In this section we investigate the performance of the complex-to-complex 2D FFT with square matrices over a range of data sizes. The square matrix sizes studied below are 32 by 32, 64 by 64, 128 by 128, 256 by 256, 512 by 512 and 1K by 1K with each matrix data size given in complex cells.

## *5.1. 2D in-place complex-to-complex FFT.*

The tables below shows the benchmark figures in microseconds for our A53, A57 and A72 platforms. Each table shows the time for the NAS function vsip_ccfft2dip_f and the equivalent set of FFTW functions in the ARM performance libraries. These functions performs a single 2D complex-to- complex in-place FFT operation.

### 5.1.1. A53 System Performance with Square Matrix Data.

**Table 22: A53 vsip_ccfft2dip_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 15.44 | 72.51 | 531.20 | 3984.65 | 28310.90 | 140164.00 |
| *ARM* | 45.58 | 225.94 | 1905.39 | 17161.70 | 80004.70 | 373372.00 |

### 5.1.2. A57 System Performance with Square Matrix Data.

**Table 23: A57 vsip_ccfft2dip_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 11.66 | 48.16 | 288.06 | 1424.36 | 10739.90 | 70208.90 |
| *ARM* | 20.61 | 160.87 | 690.99 | 5138.52 | 41436.50 | 201888.00 |

### 5.1.3. A72 System Performance with Square Matrix Data.

**Table 24: A72 vsip_ccfft2dip_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 9.83 | 42.48 | 263.93 | 1350.16 | 7251.54 | 37230.50 |
| *ARM* | 20.83 | 136.93 | 626.61 | 3929.89 | 22812.90 | 224987.00 |

Again, the A72 system and the NAS library have the best performance.

### *5.2. 2D out-of-place complex-to-complex FFT.*

The following tables show the results with the out-of-place performance with the NAS function vsip_ccfft2dop_f:

#### 5.2.1. A53 System Performance with Square Matrix Data.

**Table 25: A53 vsip_ccfft2dop_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 17.83 | 91.50 | 544.84 | 4941.48 | 29788.30 | 150362.00 |
| *ARM* | 47.99 | 251.94 | 2351.33 | 23513.30 | 80116.80 | 310184.00 |

#### 5.2.2. A57 System Performance with Square Matrix Data.

**Table 26: A57 vsip_ccfft2dop_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 9.99 | 47.71 | 280.21 | 1486.41 | 12904.10 | 76717.00 |
| *ARM* | 22.26 | 159.29 | 704.11 | 6144.18 | 42255.50 | 202102.00 |

#### 5.2.3. A72 System Performance with Square Matrix Data.

**Table 27: A72 vsip_ccfft2dop_f with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *NAS* | 10.34 | 45.49 | 260.10 | 1388.98 | 7610.27 | 44960.90 |
| *ARM* | 20.63 | 139.24 | 653.40 | 3989.17 | 26790.00 | 229909.00 |

The NAS library is slightly quicker with the in-place operation than it is with the out-of-place operation for the 2D FFT. The ARM library produces similar results when comparing in-place and out-of-place. However, even the out-of-place results with the NAS library are substantially quicker than the ARM library.

# 6. VSIPL 2D FFT Performance with Non-Square Matrix Data.

In this section we investigate the performance of the complex-to-complex 2D FFT with non-square matrices over a range of data sizes. The non-square matrix sizes studied below are 1K by 100, 4K by 50, 16K by 20, 64K by 20 and 128K by 20 given in units of complex cells.

## *6.1. 2D in-place complex-to-complex FFT.*

The tables below show the benchmark figures in microseconds for our A53, A57 and A72 platforms. Each table shows the time for the NAS function vsip_ccfft2dip_f and the equivalent set of FFTW functions in the ARM performance libraries. These functions performs a 2D complex-to-complex in-place FFT operation.

### 6.1.1. A53 System Performance with Non-Square Matrix Data.

**Table 28: A53 vsip_ccfft2dip_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 11025.60 | 18736.70 | 30433.33 | 177866.00 | 538690.00 |
| *ARM* | 17775.40 | 68499.20 | 140340.00 | 693176.00 | 1456390.00 |

### 6.1.2. A57 System Performance with Non-Square Matrix Data.

**Table 29: A57 vsip_ccfft2dip_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 2493.87 | 6362.64 | 12906.50 | 76953.11 | 205618.00 |
| *ARM* | 8579.80 | 25418.40 | 56200.90 | 296675.00 | 726713.00 |

### 6.1.3. A72 System Performance with Non-Square Matrix Data.

**Table 30: A72 vsip_ccfft2dip_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 2438.91 | 5504.75 | 9924.71 | 55034.20 | 138218.00 |
| *ARM* | 5682.02 | 17281.70 | 36680.40 | 227244.00 | 734595.00 |

The A72 system and the NAS library have the best performance.

### *6.2. 2D out-of-place complex-to-complex FFT.*

The following tables show the results with the out-of-place operation with the same set of non-square matrices:

#### 6.2.1. A53 System Performance with Non-Square Matrix Data.

**Table 31: A53 vsip_ccfft2dop_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 11978.60 | 28205.80 | 37219.40 | 191738.00 | 556749.00 |
| *ARM* | 18240.60 | 68205.80 | 144200.00 | 661663.00 | 1338270.00 |

#### 6.2.2. A57 System Performance with Non-Square Matrix Data.

**Table 32: A57 vsip_ccfft2dop_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 2770.62 | 8384.62 | 16359.10 | 85104.90 | 223293.00 |
| *ARM* | 9134.85 | 25904.20 | 55161.20 | 290890.00 | 703817.00 |

#### 6.2.3. A72 System Performance with Non-Square Matrix Data.

**Table 33: A72 vsip_ccfft2dop_f with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *NAS* | 2502.98 | 5992.34 | 11015.90 | 57813.20 | 139726.00 |
| *ARM* | 5632.88 | 16669.00 | 43099.90 | 794510.00 | 641369.00 |

Again, the A72 system and the NAS library have the best performance.

# 7. Conclusions.

In this report we have seen that the NAS VSIPL library has a performance advantage for FFT operations, over the ARM Performance Libraries (provided by the ARM Corporation) for the three test platforms this report has studied. The three test platforms were the NXP LX2160A (ARM Cortex-A72), Nvidia Jetson TX2 (ARM Cortex-A57) and Odroid (ARM Cortex-A53) systems. The benchmark figures presented in this report have been taken using the NAS serial VSIPL library running on one core in each machine. The aim was to compare core performance between systems to report on how well the SIMD algorithms had been implemented. NAS has developed threaded versions of our VSIPL libraries that speed the multiple and 2D FFT algorithms up by using multiple cores in each system. This report did not consider the threaded VSIPL library performance.

The A72 (NXP LX2160A) and the A57 (Nvidia Jetson TX2) had a similar performance for short data sizes. The A72 had an advantage over the A57 for larger data sizes. The A53 (Odroid) had the worst performance. However, the A53 is only clocked at 1.5 GHz where as both the A57 and A72 are clocked at 2.0 GHz.

The tables below report the best complex-to-complex FFT performance with the NXP LX2160A (Cortex-A72) system and the NAS Serial VSIPL library in microseconds for a range of data sizes:

**1D FFT with Vector Data Results:**

**Table 34: A72 1D FFT with vector data in microseconds.**

| | *32* | *64* | *128* | *256* | *1K* | *4K* | *16K* | *256K* | *512K* |
|---|---|---|---|---|---|---|---|---|---|
| *In-place* | 0.27 | 0.35 | 0.97 | 1.88 | 9.67 | 53.05 | 306.79 | 9168.94 | 22035.10 |
| *Out-of-place* | 0.27 | 0.36 | 0.89 | 1.97 | 10.21 | 57.13 | 311.11 | 9649.18 | 21229.10 |

**Multiple FFT with Matrix Data Results:**

**Table 35: A72 Multiple FFT with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *In-place* | 5.82 | 22.01 | 116.55 | 518.60 | 2403.71 | 10808.20 |
| *Out-of-place* | 5.43 | 23.11 | 119.13 | 569.85 | 2727.81 | 14343.80 |

**Table 36: A72 Multiple FFT with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *In-place* | 1047.63 | 3021.41 | 6763.12 | 37928.80 | 97674.30 |
| *Out-of-place* | 1135.20 | 3224.17 | 8109.61 | 39559.50 | 97889.20 |

**2D FFT with Matrix Data Results:**

**Table 37: A72 2D FFT with square matrix data in microseconds.**

| | *32x32* | *64x64* | *128x128* | *256x256* | *512x512* | *1Kx1K* |
|---|---|---|---|---|---|---|
| *In-place* | 9.83 | 42.48 | 257.84 | 1350.16 | 7251.54 | 37230.50 |
| *Out-of-place* | 10.34 | 45.49 | 260.10 | 1388.98 | 7610.27 | 44960.90 |

**Table 38: A72 2D FFT with non-square matrix data in microseconds.**

| | *1Kx100* | *4Kx50* | *16Kx20* | *64Kx20* | *128Kx20* |
|---|---|---|---|---|---|
| *In-place* | 2438.91 | 5504.75 | 9924.71 | 55034.20 | 138218.00 |
| *Out-of-place* | 2502.98 | 5992.34 | 11015.90 | 57813.20 | 139726.00 |

It can be noted from the above results that the in-place and out-of-place benchmarks are similar for small data sizes. The in-place algorithm is quicker than the out-of-place algorithm for larger data sizes.