INTEL AVX VSIPL BENCHMARKS.

N*A. Software Ltd*

2nd June 2013

1. Introduction.

NA Software (NAS) sell 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: www.vsipl.org) on PowerPC and on a number of Intel platforms including Sandy Bridge and Ivy Bridge. In 2011, NAS surveyed its customers to find out which DSP operations within this library were of most interest to them and under what data lengths and sizes. The results of this survey helped form the following list:

Table 1: VSIPL operations.

Routine	Benchmark Parameters.
1D FFT, complex-complex, in-place	256, 1K, 4K, 16K, 256K, 512K
2D FFT, complex-complex, in-place.	64*64, 128*128, 256*256, 512*512, 1K*1K, 2K*2K [columns*rows]
Multiple FFT, complex-complex, in-place.	64*64, 128*128, 256*256, 512*512, 1K*1K, 2K*2K [rows*columns]
Complex matrix transpose.	64*64, 128*128, 256*256, 512*512, 1K*1K, 2K*2K [rows*columns]
Complex vector multiply.	256, 1K, 4K, 16K, 32K, 64K, 128K
Vector sine.	256, 1K, 4K, 16K, 32K, 64K, 128K
Vector cosine.	256, 1K, 4K, 16K, 32K, 64K, 128K
Vector square root.	256, 1K, 4K, 16K, 32K, 64K, 128K
Vector scatter.	256, 1K, 4K, 16K, 32K, 64K, 128K
Vector gather.	256, 1K, 4K, 16K, 32K, 64K, 128K

All the above lengths are given as the number of complex cells.

This report benchmarks the above operations and lengths with the latest NAS Intel DSP libraries on the latest Intel platforms. The aim of this work is to benchmark the NAS Intel libraries and compare them to the best DSP libraries on the market. The work has two targets platforms of interest which are the Intel Ivy Bridge system (AVX1) and the new Intel Haswell (AVX2) platform. The following table shows the system information for both platforms:

Table 2: Test Platform Information.

Platform 1.	Type:	Ivy Bridge.
	Bit:	64
	Operating Frequency:	2.3 GHz.
	Turbo boost:	Enabled.
	Operating System:	Linux Fedora 15.
	Physical Cores:	4
Platform 2.	Type:	Haswell.
	Bit:	64
	Operating Frequency:	2.4 GHz.
	Turbo boost:	Enabled.
	Operating System:	Linux Fedora 17.

	Physical Cores:	4

There are many DSP libraries on the market that run under Ivy Bridge and Haswell platforms. However NAS has found that two of the best performing libraries are as followings:

- The Intel Integrated Performance Primitives (IPP) Library;
- The Intel Maths Kernel Library (MKL);

These two libraries represent the best of the best DSP libraries and this report considers the NAS DSP performance to them. This report benchmarks the NAS VSIPL DSP library with the IPP and MKL libraries called from a series of VSIPL wrappers. The project version information is as follows:

Table 3: Version Information.

AVX1 Timing Code Compiler	GCC Version 4.6.1
AVX1 IPP DSP library	IPP Version 7.0.205
AVX1 MKL DSP library	MKL Version 10.3.4
AVX1 NAS VSIPL DSP library	NAS Version 4.1.1
AVX1 DSP Code Compiler	ICC version 12.0.4
AVX2 Timing Code Compiler	GCC Version 4.7.1
AVX2 IPP DSP library	IPP Version 7.1.1
AVX2 MKL DSP library	MKL Version 11.0.3
AVX2 NAS VSIPL DSP library	NAS Version 4.1.1
AVX2 DSP Code Compiler	ICC version 13.1.1

This report is organised into the following sections:

Section 1: Introduction – this introduction.

Section 2: VSIPL System Performance – compares the performance of the VSIPL DSP operations of interest on Ivy Bridge and Haswell platforms.

Section 3: Conclusions – gives conclusions.

Appendix – States the GFlops rate values shown in report graphs.

2. VSIPL System Performance.

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

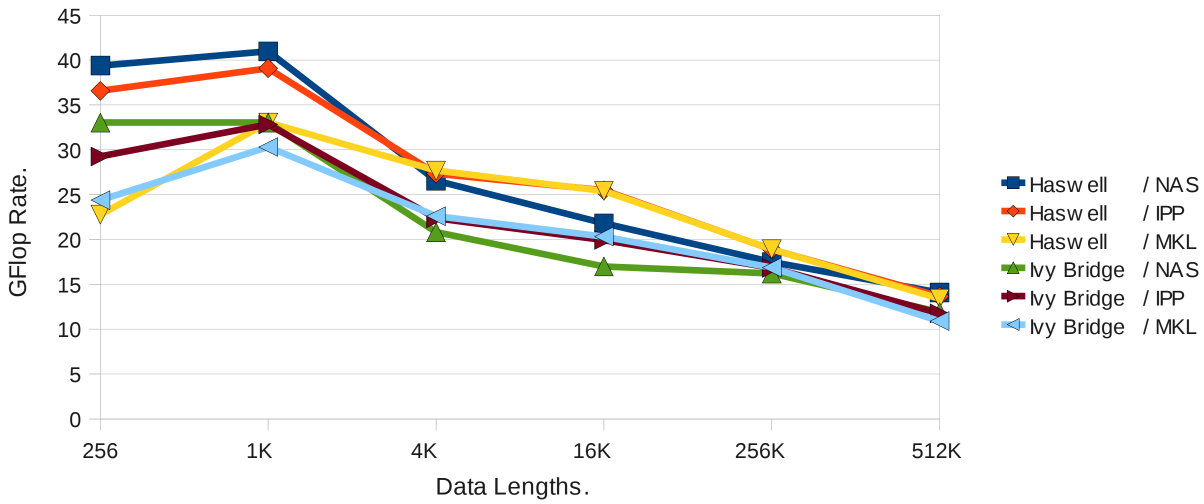
In VSIPL the in-place complex-to-complex FFT operation is carried out by a call to the library function vsip_ccfftip_f, after views and FFT plans have been created. The following table shows the obtained timings in microseconds for the 1D FFT operation with split complex data. The table contains timings for both the Haswell and Ivy Bridge systems. The benchmarks were taken with NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 4: vsip_ccfftip_f timings in microseconds.

System / DSP	*256*	*1K*	*4K*	*16K*	*256K*	*512K*
Haswell / NAS	0.26	1.25	9.26	52.63	1,349.25	3,538.21
Haswell / IPP	0.28	1.31	8.99	45.03	1,248.61	3,645.37
Haswell / MKL	0.45	1.55	8.88	45.13	1,247.28	3,720.65
Ivy Bridge / NAS	0.31	1.55	11.81	67.48	1,455.55	4,185.49
Ivy Bridge / IPP	0.35	1.56	10.99	57.51	1,402.18	4,235.54
Ivy Bridge / MKL	0.42	1.69	10.88	56.44	1,397.49	4,570.81

The above table shows that the Haswell platform has a significant performance advantage for the 1D FFT operation. The following graph show the performance of the NAS, IPP and MKL DSP libraries in GFlops for both the Haswell platform and the Ivy Bridge platform.

Graph 1: NAS vsip_ccfftip_f with split complex data.
MFLOPS = 5 N Log_2(N) / (time for one FFT in microseconds)



The peak performance is 41 GFlops for the NAS library on the Haswell platform at a data length of 1K. All three DSP libraries produce similar timings with NAS being slightly quicker than IPP/MKL for three lengths and slightly slower for three lengths. The peak performance on the Ivy Bridge platform is 33 GFlops at a data length of 1K for the NAS library.

2.2. 2D in-place complex-to-complex FFT.

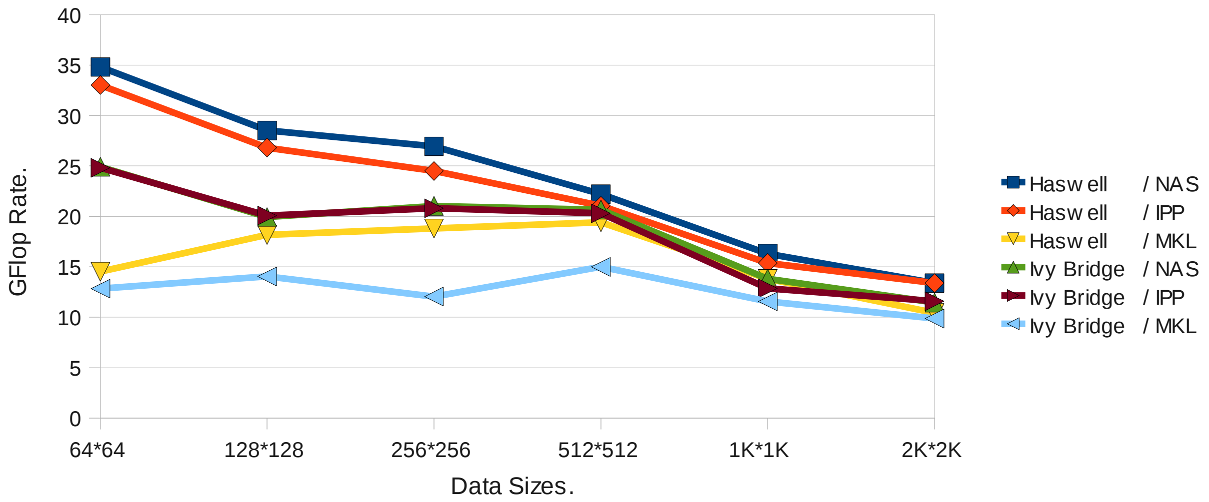
The 2D in-place complex-to-complex operation is carried out by the library call vsip_ccfft2dip_f. This function gets called after VSIPL data matrix views and FFT plans have been created. The following section show the performance of this operation over a range of square matrices with complex split data. The following table shows the timings obtained in microseconds for the 2D FFT operation. The table contains timings for both Haswell and Ivy Bridge systems. The benchmarks were taken with NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 5: vsip_ccfft2dip_f timings in microseconds.

System / DSP	*64*64*	*128*128*	*256*256*	*512*512*	*1K*1K*	*2K*2K*
Haswell / NAS	7.06	40.22	194.58	1,062.47	6,428.83	34,453.40
Haswell / IPP	7.44	42.76	213.93	1,121.13	6,817.57	34,487.10
Haswell / MKL	16.93	63.16	278.91	1,214.92	7,590.97	44,265.10
Ivy Bridge / NAS	9.86	57.52	249.32	1,141.92	7,586.55	40,288.60
Ivy Bridge / IPP	9.90	57.12	252.04	1,162.13	8,146.23	39,812.70
Ivy Bridge / MKL	19.15	81.69	435.07	1,572.92	9,062.33	46,753.60

The above table shows that the Haswell system has a significant performance advantage over Ivy Bridge over all data sizes. The following graph show the performance of the NAS, IPP and MKL DSP libraries in GFlops for both Haswell and Ivy Bridge.

Graph 2: NAS vsip_ccfft2dip_f with square complex split data.
MFLOPS = 5NLog$_2$(N)*M + 5MLog$_2$(M)*N / (time for one FFT in microseconds)



The peak performance is 35 GFlop on the Haswell platform with the NAS library at a data size of 64*64 complex cells. On Haswell the NAS library (shown in blue) is slightly quicker than IPP, which is a lot quicker than the MKL library. On Ivy Bridge the peak performance is 25 GFlops at a data size of 64*64 for both the NAS and IPP libraries. On Ivy Bridge both NAS and IPP obtain the same performance. The NAS green graph is almost covered by the IPP purple graph. Both NAS and IPP are quicker than MKL.

2.3. Multiple 1D Complex to Complex FFTs.

The multiple in-place complex-to-complex operation is carried out by the library call vsip_ccfftmip_f. This function gets called after VSIPL data matrix views and FFT plans have been created. The following section show the performance of this operation over a range of square matrix sizes with split complex data.

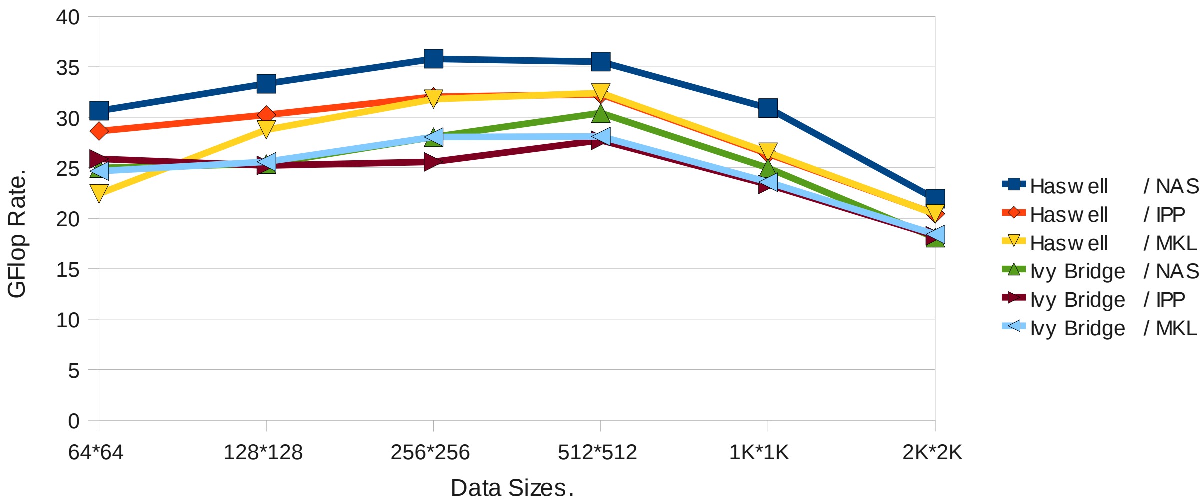
The following table shows the timings obtained in microseconds for the multiple FFT operation. The table contains timings for the both Haswell, and Ivy Bridge systems. The benchmarks were taken with NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 6: vsip_ccfftmip_f timings in microseconds.

System / DSP	*64*64*	*128*128*	*256*256*	*512*512*	*1K*1K*	*2K*2K*
Haswell / NAS	4.01	17.21	73.22	332.21	1,695.70	10,517.90
Haswell / IPP	4.29	18.96	81.84	365.57	1,988.26	11,281.60
Haswell / MKL	5.48	19.94	82.43	363.97	1,976.90	11,290.05
Ivy Bridge / NAS	4.92	22.60	93.50	387.75	2,098.57	12,763.90
Ivy Bridge / IPP	4.75	22.73	102.45	425.41	2,244.99	12,626.20
Ivy Bridge / MKL	4.98	22.41	93.45	419.87	2,223.06	12,540.50

The above table shows that the Haswell platform has a significant performance advantage over Ivy Bridge for every data size. The following graph show the performance of the NAS, IPP and MKL libraries in GFlops for Haswell and Ivy Bridge.

Graph 3: NAS vsip_ccfftmip_f with square complex split data.
MFLOPS = 5NLog$_2$(N)*M / (time for one FFT in microseconds)



The peak performance is 36 GFlops on the Haswell platform for the NAS library (shown in blue) at a data size of 256*256. The NAS library has a clear performance advantage over all data lengths on the Haswell platform. The peak performance on the Ivy Bridge platform is 30 GFlops with the NAS library at a data size of 512*512 complex cells.

2.4. Complex Transpose.

A complex matrix transpose operation is performed in VSIPL by calling the function vsip_cmtrans_f. This function is given a VSIPL view representing an input data matrix and the function transposes the views data into an output view. The operation is therefore performed out-of-place. The following table shows the timings obtained for the complex matrix transpose operation with square complex split data. The timings are given in microseconds.

Table 7: vsip_cmtrans_f timings in microseconds.

System / DSP	64*64	128*128	256*256	512*512	1K*1K	2K*2K
Haswell / NAS	2.31	8.64	39.37	176.18	2,015.80	10,315.50
Haswell / IPP	2.40	10.43	58.29	247.21	3,198.97	14,542.90
Haswell / MKL	23.67	67.87	209.01	856.63	4,411.78	18,639.10
Ivy Bridge / NAS	2.38	10.50	50.38	217.87	1,994.26	9,893.69
Ivy Bridge / IPP	2.70	11.79	74.96	348.05	3,928.70	18,409.20
Ivy Bridge / MKL	15.05	40.34	167.14	572.86	3,427.42	21,758.90

Looking at the NAS library the above table shows that the Haswell platform has a clear performance advantage for data sizes 128*128, 256*256, 512*512 and 1K*1K. Ivy Bridge and Haswell produce similar timings for 64*64 and 2K*2K.

Looking at the IPP library the Haswell platform is faster with lengths 256*256, 512*512 and 1K*1K.

However, looking at the MKL timings the Ivy Bridge system is faster than the Haswell system for every data length apart from 2K*2K. However, it looks like the MKL library is not well optimised for this operation given that the MKL benchmarks are a lot slower than both NAS and IPP.

The above timings show that the NAS library is substantially quicker than both IPP and MKL for all data sizes on the Ivy Bridge and Haswell platforms.

2.5. Complex Vector Multiply.

The complex vector multiply is performed in the VSIPL library by a call to the function vsip_cvmul_f. In this test one complex split vector is multiplied with a second and the result is placed in a third output vector.

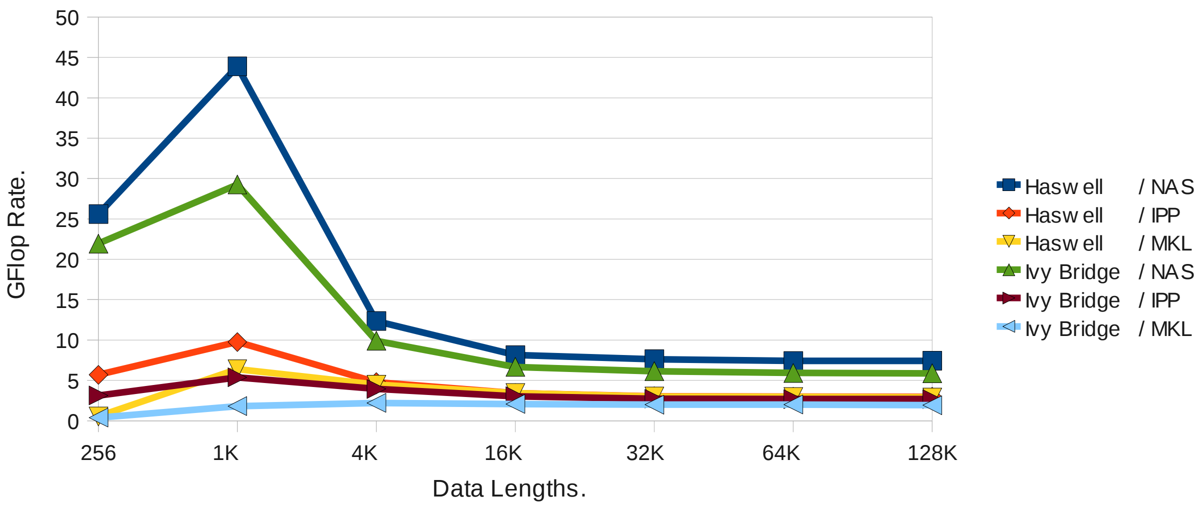
The following table shows the timings obtained in microseconds for the complex vector multiply operation. The table contains timings for both the Haswell and Ivy Bridge systems. The benchmarks were taken with both NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 8: vsip_cvmul_f timings in microseconds.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	0.06	0.14	1.99	12.09	25.77	52.97	105.86
Haswell / IPP	0.27	0.63	5.13	28.63	64.76	131.64	264.79
Haswell / MKL	2.78	0.96	5.48	28.54	65.62	133.28	272.38
Ivy Bridge / NAS	0.07	0.21	2.48	14.74	32.02	66.39	133.83
Ivy Bridge / IPP	0.49	1.14	6.23	32.30	72.10	146.49	292.61
Ivy Bridge / MKL	3.97	3.37	11.18	47.19	98.81	195.81	407.19

The table shows that the Haswell platform has a performance advantage for all three DSP libraries. The following graph show the performance of the NAS, IPP and MKL libraries for the complex vector multiply operation in GFlops.

Graph 4: NAS vsip_cvmul_f with complex split data.
MFLOPS = 6 * N / (time for one vector multiply in microseconds)



The above graph shows:

- Both the NAS AVX1 (Ivy Bridge in green) and AVX2 (Haswell in blue) libraries out perform IPP/MKL with this operations across all data lengths.
- The peak performance is 44 GFlops for the NAS library on Haswell at a data size of 1K complex cells.
- The peak performance for Ivy Bridge is 29 GFlops for the NAS library at a data size of 1K complex cells.

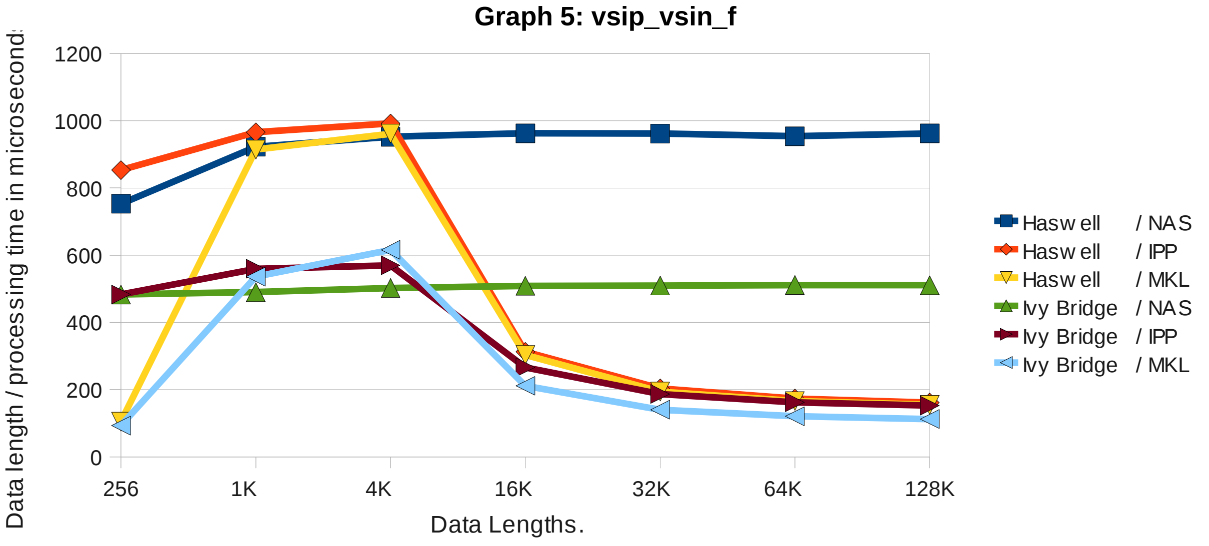
2.6. Vector Sine.

The vector sin operation is performed in the VSIPL library by a call to the function vsip_vsin_f. In this test the sine is taken of each element within a floating point input vector and the result is placed into an output vector. The following table shows the timings obtained in microseconds for the vector sin operation with float data. The table contains timings for the Haswell and Ivy Bridge systems. The benchmarks were taken with both NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 9: vsip_vsin_f timings in microseconds.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	0.34	1.11	4.30	17.02	34.07	68.69	136.22
Haswell / IPP	0.30	1.06	4.13	52.41	160.81	377.79	811.86
Haswell / MKL	2.44	1.12	4.26	54.06	168.93	394.37	846.17
Ivy Bridge / NAS	0.53	2.09	8.15	32.20	64.27	128.23	256.51
Ivy Bridge / IPP	0.53	1.83	7.19	61.53	175.44	403.89	859.56
Ivy Bridge / MKL	2.75	1.91	6.65	77.48	233.33	543.89	1,166.00

The above table shows that there is a large performance advantage for the Haswell platform over Ivy Bridge with all three DSP libraries. The following graph show the performance of the NAS, IPP and MKL libraries for the sin operation. The performance in the y axis is shown in units of data length divided by processing time in microseconds.



NAS and IPP DSP libraries have a similar performance up to a vector length of 4K. After 4K the NAS library has a steady performance while the performance of IPP drops off. IPP would have a better performance if the input data is range reduced. However, this test does not perform any range reduction. It can be observed from the NAS blue and green graphs that the performance almost doubles across all vector lengths for the Haswell platform.

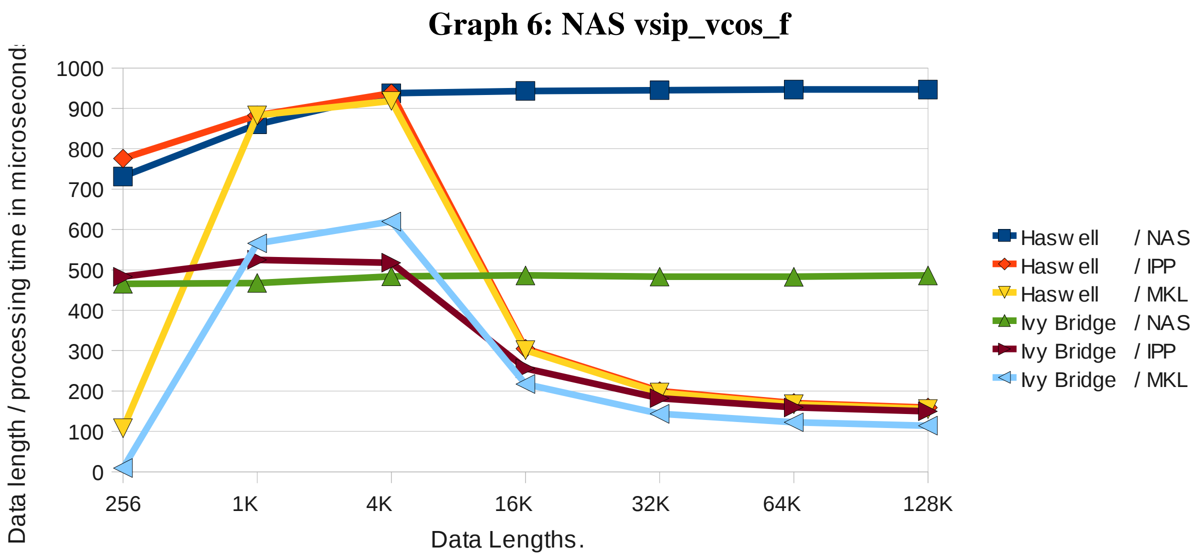
2.7. Vector Cosine.

The vector cosine operation is performed in the VSIPL library by a call to the function vsip_vcos_f. In this test the cosine is taken of each element in a floating point input vector and the result is placed into an output vector. The following table shows the timings obtained in microseconds for the vector cosine operation with float data. The table contains timings for the Haswell and Ivy Bridge systems. The benchmarks were taken with both NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 10: vsip_vcos_f timings in microseconds.

System / DSP	256	1K	4K	16K	32K	64K	128K
Haswell / NAS	0.35	1.19	4.37	17.37	34.68	69.23	138.44
Haswell / IPP	0.33	1.16	4.37	53.79	164.04	384.65	824.34
Haswell / MKL	2.38	1.16	4.46	54.39	167.19	392.25	843.45
Ivy Bridge / NAS	0.55	2.19	8.46	33.70	67.81	135.69	249.19
Ivy Bridge / IPP	0.53	1.95	7.91	64.00	179.85	411.29	875.36
Ivy Bridge / MKL	28.33	1.81	6.61	75.44	228.35	534.60	1,148.03

The following three graphs show the performance of the NAS, IPP and MKL libraries respectively for the cosine operation. The performance in the y axis is shown in units of data length divided by processing time in microseconds.



The NAS and IPP DSP libraries have a similar performance up to a vector length of 4K. After 4K the NAS library has a steady performance while the performance of IPP drops off. IPP would have a better performance if the input data is range reduced. However, this test does not perform any range reduction. It can be observed from the NAS blue and green graphs that the performance almost doubles across all vector lengths for the Haswell platform.

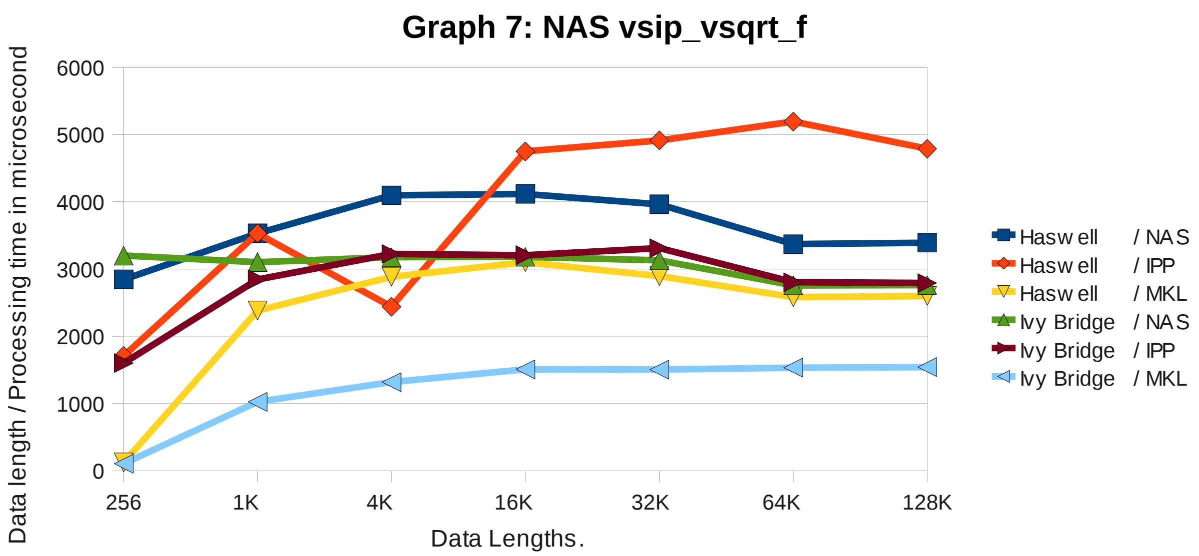
2.8. Vector Square Root.

The vector square root operation is performed in the VSIPL library by a call to the function vsip_vsqrt_f. In this test the square root is taken of each element in a floating point input vector and the result is placed into an output vector. The following table shows the timings obtained in microseconds for the vector square root operation with float data. The table contains timings for the Haswell and Ivy Bridge systems. The benchmarks were taken with both NAS based timings, IPP based timings and timings based on Intel's MKL library.

Table 7: vsip_vsqrt_f timings in microseconds.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	0.09	0.29	1.00	3.98	8.27	19.45	38.64
Haswell / IPP	0.15	0.29	1.68	3.45	6.67	12.62	27.37
Haswell / MKL	2.09	0.43	1.42	5.28	11.32	25.38	50.41
Ivy Bridge / NAS	0.08	0.33	1.29	5.15	10.47	23.77	47.47
Ivy Bridge / IPP	0.16	0.36	1.27	5.11	9.90	23.38	46.94
Ivy Bridge / MKL	2.45	1.00	3.11	10.87	21.79	42.73	84.98

The above table shows that the Haswell platform has a major performance advantage over Ivy Bridge for every data length. The graph below highlights this performance. The performance in the y axis is shown in units of data length divided by processing time in microseconds.



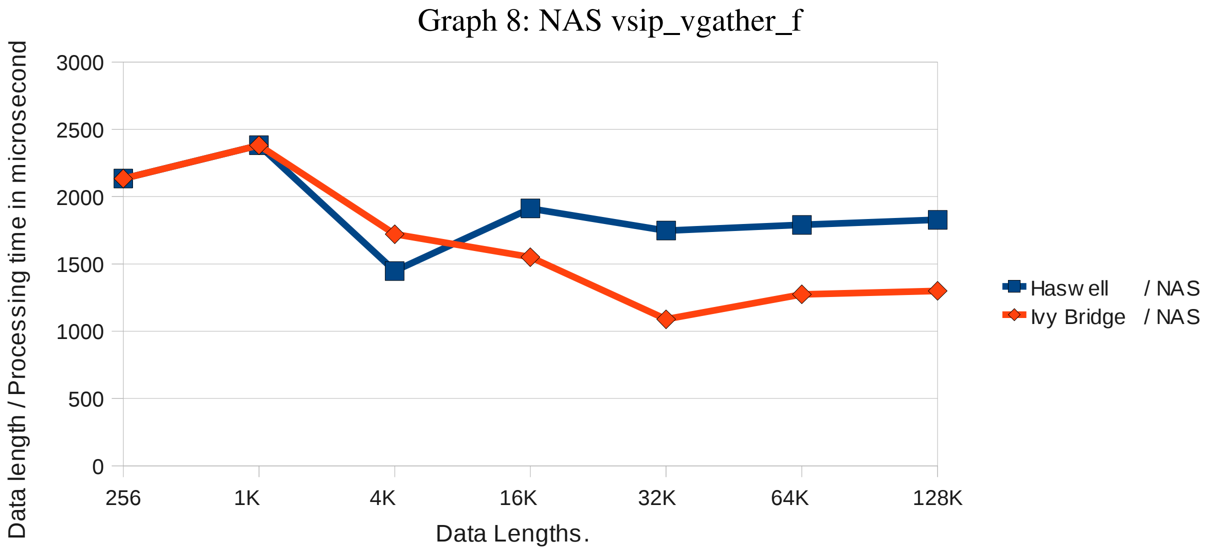
The graphs show that NAS (blue/green) and IPP (red/purple) have a better performance than the MKL yellow and light blue graphs. Under Ivy Bridge IPP and NAS have a similar performance apart from length 256 where NAS has a significant performance advantage. Under Haswell NAS has the performance advantage up to 16K where IPP then has a significant performance advantage.

2.9. Vector Gather.

The vector gather operation usually gathers elements from a large input vector into a smaller output vector. The smaller output vector comes with a vector of indexes stating the position of the elements in the larger vector. The output vector and vector of indexes are the same size and are given in the table below. The IPP and MKL libraries do not have an equivalent function to the vector gather operation. Therefore, the table below shows the processing time of the NAS library only.

Table 12: vsip_vgather_f timings in microseconds.

System / DSP	256	1K	4K	16K	32K	64K	128K
Haswell / NAS	0.12	0.43	2.83	8.57	18.75	36.61	71.72
Ivy Bridge / NAS	0.12	0.43	2.38	10.57	30.10	51.46	100.82



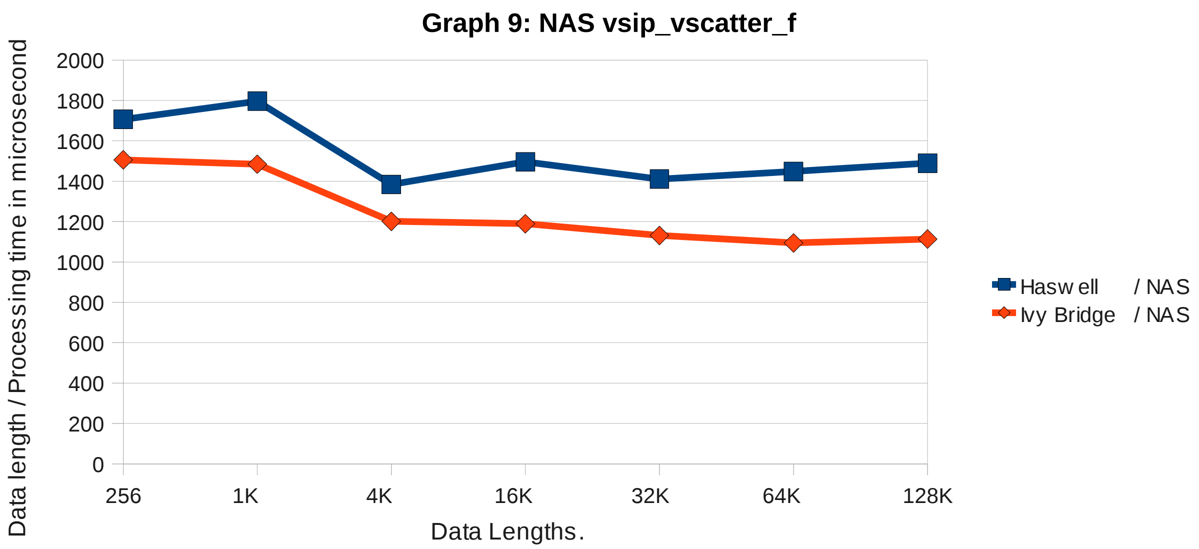
The graphs show the Haswell platform having a significant performance advantage after a vector length of 4K. The peak performance occurs at a vector length of 1K where all platforms have a similar performance.

2.10. Vector Scatter.

The vector scatter operation usually uses a small input vector and a large output vector. The smaller input vector also comes with an input vector of indexes stating how its elements should be scattered into the larger vector. The VSIPL library performs the vector scatter operation with a call to vsip_vscatter_f. In the tests below the larger vector is twice the size of the smaller vector and the elements are scattered into every other element of the larger vector. The data lengths below give the input vector size. The IPP and MKL libraries do not contain the vector scatter operation so the table below only considers the NAS library.

Table 13: vsip_vscatter_f timings in microseconds.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	0.15	0.57	2.96	10.95	23.23	45.25	88.02
Ivy Bridge / NAS	0.17	0.69	3.41	13.78	28.96	59.90	117.73



The graphs show the Haswell platform having a significant performance advantage across all vector lengths for the vector scatter operation. The peak performance occurs at a vector length of 1K with the Haswell platform.

3. Conclusions.

This report has studied the performance of the following VSIPL DSP operations on the Ivy Bridge and Haswell platforms. The NAS, IPP and MKL DSP libraries have been used under VSIPL wrappers to benchmark the DSP operations. The report has shown that the Haswell platform has a significant performance advantage for all the DSP operations over most of the data sizes studied.

Section 2 of this report has shown that all three libraries are well optimised for the DSP operations and data sizes of interest with the following exceptions:

- NAS is better optimised for the complex vector multiply operations;
- NAS is better optimised for sin/cosine when the data is not range reduced;
- IPP and MKL do not cover vector scatter operations.
- IPP and MKL do not cover vector gather operations.

The following table shows which DSP library and platform produced the optimum performance given in section 2:

Table 14: Optimum Library.

Routine	Platform	GFLOP Peak Performance	Optimum DSP library.
1D FFT	Haswell*	41 GFlops with NAS	NAS or IPP or MKL
	Ivy Bridge	33 GFlops with NAS	NAS or IPP or MKL
2D FFT	Haswell*	35 GFlops with NAS	NAS
	Ivy Bridge	25 GFlops with NAS or IPP	NAS or IPP
Multiple FFT	Haswell*	36 GFlops with NAS	NAS
	Ivy Bridge	30 GFlops with NAS	NAS or IPP or MKL
Transpose	Haswell*	N/A	NAS
	Ivy Bridge	N/A	NAS
Complex vector multiply	Haswell*	44 GFlops with NAS	NAS
	Ivy Bridge	29 GFlops with NAS	NAS
Vector sine.	Haswell*	N/A	NAS
	Ivy Bridge	N/A	NAS
Vector cosine.	Haswell*	N/A	NAS
	Ivy Bridge	N/A	NAS
Vector square root.	Haswell*	N/A	NAS or IPP
	Ivy Bridge	N/A	NAS or IPP
Vector scatter.	Haswell*	N/A	NAS
	Ivy Bridge	N/A	NAS
Vector gather.	Haswell*	N/A	NAS
	Ivy Bridge	N/A	NAS

(units* = optimum platform for operation).

The following table shows the speed ups obtained with the Haswell platform compared to Ivy Bridge with all three DSP libraries. The table shows both the largest speed up and the average speed up over the data lengths of interest.

Table 15: Haswell Speed Ups.

Routine.	NAS Speed ups		IPP Speed ups		MKL Speed ups	
	Largest	Average	Largest	Average	Largest	Average
1D FFT	1.28	1.20	1.27	1.20	1.25	1.14
2D FFT	1.43	1.25	1.34	1.20	1.55	1.25
Multiple FFT	1.31	1.24	1.25	1.16	1.15	1.09
Transpose	1.42	1.15	1.40	1.23	1.16	0.77
Complex vector multiply	1.57	1.28	1.81	1.42	3.5	1.87
Vector sine	1.89	1.83	1.77	1.38	1.70	1.42
Vector cosine	1.95	1.88	1.81	1.35	1.56	1.38
Vector square root	1.29	1.19	1.85	1.36	2.32	1.86
Vector gather	1.56	1.21	N/A	N/A	N/A	N/A
Vector scatter	1.34	1.24	N/A	N/A	N/A	N/A

The above table shows that the Haswell platform obtains large performance increases across all DSP operations and most data lengths of interest. The only exception to this are the MKL transpose operations. The trigonometric functions obtain speed ups close than 2 and the FFT's often achieve speed-ups up to 1.5.

4. Appendix.

4.1. Values in Graphs.

The following tables show the values used to form the graphs throughout this report.

4.1.1. Graphs for 1D FFT split complex data.

Graph 1: vsip_ccfftip_f with split complex data in GFlops.

System / DSP	*256*	*1K*	*4K*	*16K*	*256K*	*512K*
Haswell / NAS	39.38	40.96	26.54	21.79	17.49	14.08
Haswell / IPP	36.57	39.08	27.34	25.47	18.90	13.66
Haswell / MKL	22.76	33.03	27.68	25.41	18.92	13.39
Ivy Bridge / NAS	33.03	33.03	20.81	17.00	16.21	11.90
Ivy Bridge / IPP	29.26	32.82	22.36	19.94	16.83	11.76
Ivy Bridge / MKL	24.38	30.30	22.59	20.32	16.88	10.90

4.1.2. Graphs for 2D FFT split complex data.

Graph 2: vsip_ccfft2dip_f with square split complex data in GFlops.

System / DSP	*64*64*	*128*128*	*256*256*	*512*512*	*1K*1K*	*2K*2K*
Haswell / NAS	34.81	28.52	26.94	22.21	16.31	13.39
Haswell / IPP	33.03	26.82	24.51	21.04	15.38	13.38
Haswell / MKL	14.52	18.16	18.80	19.42	13.81	10.42
Ivy Bridge / NAS	24.92	19.94	21.03	20.66	13.82	11.45
Ivy Bridge / IPP	24.82	20.08	20.80	20.30	12.87	11.59
Ivy Bridge / MKL	12.83	14.04	12.05	15.00	11.57	9.87

4.1.3. Graphs for multiple FFT with split complex data.

Graph 3: vsip_ccfftmip with split complex data in GFlops.

System / DSP	*64*64*	*128*128*	*256*256*	*512*512*	*1K*1K*	*2K*2K*
Haswell / NAS	30.64	33.32	35.80	35.51	30.92	21.93
Haswell / IPP	28.64	30.24	32.03	32.27	26.37	20.45
Haswell / MKL	22.42	28.76	31.80	32.41	26.52	20.43
Ivy Bridge / NAS	24.98	25.37	28.04	30.42	24.98	18.07
Ivy Bridge / IPP	25.87	25.23	25.59	27.73	23.35	18.27
Ivy Bridge / MKL	24.67	25.59	28.05	28.10	23.58	18.40

4.1.4. Graphs for complex vector multiply with split complex data.

Graph 4: vsip_cvmul_f with split complex data in GFlops.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	25.60	43.89	12.35	8.13	7.63	7.42	7.43
Haswell / IPP	5.69	9.75	4.79	3.43	3.04	2.99	2.97
Haswell / MKL	0.55	6.40	4.48	3.44	3.00	2.95	2.89
Ivy Bridge / NAS	21.94	29.26	9.91	6.67	6.14	5.92	5.88
Ivy Bridge / IPP	3.13	5.39	3.94	3.04	2.73	2.68	2.69
Ivy Bridge / MKL	0.39	1.82	2.20	2.08	1.99	2.01	1.93

4.1.5. Graphs for vector sin.

Graph 5: vsip_vsin_f with float data in units of data length divided by processing time.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	752.94	922.52	952.56	962.63	961.78	954.08	962.21
Haswell / IPP	853.33	966.04	991.77	312.61	203.77	173.47	161.45
Haswell / MKL	104.92	914.29	961.50	303.07	193.97	166.18	154.90
Ivy Bridge / NAS	483.02	489.95	502.58	508.82	509.85	511.08	510.98
Ivy Bridge / IPP	483.02	559.56	569.68	266.28	186.78	162.26	152.49
Ivy Bridge / MKL	93.09	536.13	615.94	211.46	140.44	120.49	112.41

4.1.6. Graphs for vector cosine.

Graph 6: vsip_vcos_f with float data in units of data length divided by processing time.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	731.43	860.50	937.30	943.24	944.87	946.64	946.78
Haswell / IPP	775.76	882.76	937.30	304.59	199.76	170.38	159.00
Haswell / MKL	107.56	882.76	918.39	301.23	195.99	167.08	155.40
Ivy Bridge / NAS	465.45	467.58	484.16	486.17	483.23	482.98	486.32
Ivy Bridge / IPP	483.02	525.13	517.83	256.00	182.20	159.34	149.73
Ivy Bridge / MKL	9.04	565.75	619.67	217.18	143.50	122.59	114.17

4.1.7. Graphs for vector square root.

Graph 7: vsip_vsqrt_f with float data in units of data length divided by processing time.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	2,844.44	3,531.03	4,096.00	4,116.58	3,962.27	3,369.46	3,392.13
Haswell / IPP	1,706.67	3,531.03	2,438.10	4,748.99	4,912.74	5,193.03	4,788.89
Haswell / MKL	122.49	2,381.40	2,884.51	3,103.03	2,894.70	2,582.19	2,600.12
Ivy Bridge / NAS	3,200.00	3,103.03	3,175.19	3,181.36	3,129.70	2,757.09	2,761.15
Ivy Bridge / IPP	1,600.00	2,844.44	3,225.20	3,206.26	3,309.90	2,803.08	2,792.33
Ivy Bridge / MKL	104.49	1,024.00	1,317.04	1,507.27	1,503.81	1,533.72	1,542.39

4.1.8. Graphs for vector gather.

Graph 8: vsip_vgather_f with float data in units of data length divided by processing time.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	2,133.33	2,381.40	1,447.35	1,911.79	1,747.63	1,790.11	1,827.55
Ivy Bridge / NAS	2,133.33	2,381.40	1,721.01	1,550.05	1,088.64	1,273.53	1,300.06

4.1.9. Graphs for vector scatter.

Graph 9: vsip_vscatter_f with float data in units of data length divided by processing time.

System / DSP	*256*	*1K*	*4K*	*16K*	*32K*	*64K*	*128K*
Haswell / NAS	1,706.67	1,796.49	1,383.78	1,496.26	1,410.59	1,448.31	1,489.12
Ivy Bridge / NAS	1,505.88	1,484.06	1,201.17	1,188.97	1,131.49	1,094.09	1,113.33