Raspberry Pi 5 64 Bit Benchmarks and Stress Tests

Roy Longbottom

Contents

Summary

As indicated below, some of the benchmarks provided higher average Pi5/Pi4 performance gains than the official claim of two to three times, where individual programs or test functions were between 10 and 18 times faster. This was due to the improved CPU caching arrangements and advanced SIMD hardware and compilation facilities. Examples of compiled SIMD vector instructions are included.

The latest 5 amps power supply and active cooler were not available initially, when tests were run with no cooling fan. Then, stress tests lead to CPU temperature increasing up to 91.7°C, but the Pi 5 continued running at a lower speed with controlled CPU MHz and voltage variations, still much faster than a fan cooled Pi 4.

On the downside, my rather extreme stress tests produced a number of system crashes and disk drive reading errors. I believe that the results show that this was not associated with high temperatures but inadequate USB power was to blame. Although stress tests ran successfully using the 5 amps power supply, USB power demands of disk and solid state drives appear to be rather excessive. In this case, the system could be easily crashed by overloading. So these drives should probably only be connected via a powered hub.

Surprisingly, execution of a new stress test, with integer calculations, generated more heat than the floating point variety. The hottest occurred when handling data from L2 cache with higher power demands. Faster L1 cache based data transfers produced somewhat lower temperatures.

Benchmarks - Besides detailed results, Pi5/Pi4 performance comparisons are provided using older gcc8 compiled versions, also the latter with new varieties from gcc12, included in the new 64 bit Operating System software.

Single Core CPU Tests - comprising varieties of Whetstone, Dhrystone, Linpack 100 and Livermore Loops Classic Benchmarks. Pi 5 gains were between 2.14 and 4.65 times from 182 measurements.

Single Core Memory Benchmarks - measuring performance using data from caches and RAM. More than 250 Pi5/Pi4 comparisons are provided from five benchmarks, indicating a Pi 5 average gain of 3.1 times maximum 13.3 times. Pi 5 new compilation average gain was 2.6 times and maximum 10 times. High gains were due to improved caching and SIMD vector processing operations.

MultiThreading Benchmarks - These 8 benchmarks execute the same calculations using 1, 2, 4 and 8 threads. From 150 plus comparisons Pi5/Pi4 average/maximum gains were 3.4/18.2 times, with 1.2/5.6 times for Pi 5 gcc12/gcc8 compilations. The reasons for the high gains were improved caching and SIMD as above.

Miscellaneous - average Pi5/Pi4 performance gains for a series of tests were Java Whetstones 2.47 times, JavaDraw 1.98 times and OpenGL 4.0 times for 6 tests at 4 screen resolutions.

Input/Output Benchmarks - These measure performance of large files, small files and random access with numerous performance measurements of Gbps LAN, WiFi, large files with 64 bit OS, main SD and USB 3 FAT and Ext disk drives and 11 main and USB boot drives. Also are booting times, main and USB volts and amps power usage. First test result indicated that Pi 5 was typically 50% faster than Pi 4 handling large files on a high speed USB 3 flash drive.

Drive Stress Test - This writes four large files with data comprising numerous binary data patterns, reads them randomly for a specified time, then repetitively reads each different data block for a time. Eleven 15 minute tests were successfully run on the Pi 5 comprising LAN, WiFi, OS SD, 3 USB 3 flash drives and 5 disk drive partitions, plus 2 network tests from a Pi 400.

Disk Drive Errors and System Crashes - (Power supply issues) - Two out of three tests using 2 disk drives caused crashes one with both on a USB 3 hub, due to exceeding 900 mA USB 3 port specification. Next crash was with one drive via hub, one direct USB and a CPU stress test leading to measured main power supply exceeding the 3 amps specification. This lead to reading the wrong file and data comparison failures. Two disks on different USB 3 ports ran successfully.

CPU Stress Tests - Initial 3 floating point and 3 integer tests were run without fan cooling, each for 15 minutes, using 1, 2 and 4 threads, whilst recording performance, CPU MHz, volts and temperatures. All suffered from MHz throttling at temperatures up to 90°C, with measured performance deterioration less than 50%, still faster than a fan cooled Pi 4. I acquired a 4 amps power supply and repeated the test that crashed at 3 amps, this time with no failures.

INTitHOT New Integer Stress - This read only test produced the hottest and fastest effects, through executing continuous SIMD AND instructions. On the Pi 5, fastest, via L1 cache sized data, obtained 240 GB/second or Terabit speed of 1.92 Tbps. Via L2 cache, maximum speed was 168 GB/second with higher power consumption and Temperature. The Pi 5 was around 4.6 times faster than a Pi 4 using 1 or 2 threads, and much greater at 4 threads where the Pi 4 was unbelievably slow.

System Stress Tests - These were run for 30 minutes using the 4 amps power supply and included INTitHOT, disk drive and OpenGL stress tests. Initial tests ran successfully at near maximum speed with the fan but reached a CPU temperature of 91.7°C with a 40% reduction in CPU and graphics performance without the fan. The next ones included floating point and network stress tests. The no fan test ran successfully with the usual high temperature and degraded performance but, with the fan, crashed with disk drive errors again. Then a low USB voltage was recorded.

Other Tests and Comparisons - Tests were carried out involving Firefox, Bluetooth sound and YouTube videos. Next is Pi-5 The Vector Processor, with examples and comparing performance with 1978 to 1991 supercomputers, then Comparisons with PCs from 1991 to 2021. Results for the latter indicate that the Raspberry Pi 5 can be assumed to be 194 times faster than the Cray 1 supercomputer.

New 5 Amps Power Supply and Active Cooler - Graphs of temperature increases with time are provided for initial CPU only stress tests, followed by others using the new items, now all much less than the the CPU MHz throttling level. Hottest was not the floating point test but the one using integer calculations with L2 cache based data. Next was a repeat of the Heavy System Stress Tests. This ran successfully twice. It was then repeated with the 4 amps power supply and failed as before but at a much lower CPU temperature, then ran without any issues at a second attempt. The strange measured power volts and amps probably indicate a marginal condition, compared to the 5 amps measurements.

Solid State Hard Drive - Following an earlier disastrous attempt, I repeated the last system stress test powered with 4 and 5 amps supplies on the Pi 5, providing similar performance. Then I ran the drive benchmarks where average large file write/reading speeds were around 360/400 MB/second, faster than the old hard drive. A surprise was tha the measured USB current was the relatively high 640 mA.

Introduction below or Go To Start

Introduction

This report provides results from a wide range of benchmarks and stress tests run on the Raspberry Pi 5 during the Alpha Testing stage. and includes comparisons with the Pi 4. It follows the format of many other reports from 2014 to 2023 available from This ResearchGate Index. The latter includes access to historic results, opening the opportunity to compare Pi 5 performance with computers from as far back as the pre-1960 iron age.

The new Raspberry Pi 5 features a 2.4GHz quad-core 64-bit Arm Cortex-A76 CPU, with near 64 KB L1 and 512 KB L2 caches per core, and a 2MB shared L3 cache, also a host of other enhanced features. Compared to the Raspberry Pi 4, it was claimed to have between two and three times the CPU and GPU performance, with roughly twice the memory and I/O bandwidth. Part of the reason for this is that the Pi 4 runs at 1.5 GHz with a 32 KB L1 cache and 1024 KB shared L2 cache.

The first benchmarks measure performance of a single CPU core, covering integer and floating point performance plus data transfer speeds at all memory cache and RAM levels. Then there are multi-core benchmarks of the same variety and more, plus others for Java and graphics. The stress testing programs measure performance, CPU MHz and temperatures with and without fan cooling, initially for each program then during systems tests, including all CPU cores, disk and network drives and graphics. Then there are other measurements as identified in the contents table, including comparisons with PCs and supercomputers.

All the programs save the results in log files, full details from some are included in the report. These include the following information of the system under test.

```
Raspberry Pi 4 Old OS

Architecture:        aarch64
Byte Order:          Little Endian
CPU(s):              4
On-line CPU(s) list: 0-3
Thread(s) per core:  1
Core(s) per socket:  4
Socket(s):           1
Vendor ID:           ARM
Model:               3
Model name:          Cortex-A72
Stepping:            r0p3
CPU max MHz:         1500.0000
CPU min MHz:         600.0000
BogoMIPS:            108.00
Flags:               fp asimd evtstrm crc32 cpuid
Linux raspberrypi 4.19.118-v8+ #1311 SMP PREEMPT
Mon Apr 27 14:32:38 BST 2020 aarch64 GNU/Linux

Raspberry Pi 5

Architecture:                    aarch64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
CPU(s):                          4
On-line CPU(s) list:             0-3
Vendor ID:                       ARM
Model name:                      Cortex-A76
Model:                           1
Thread(s) per core:              1
Core(s) per cluster:             4
Socket(s):                       -
Cluster(s):                      1
Stepping:                        r4p1
CPU(s) scaling MHz:              100%
CPU max MHz:                     2400.0000
CPU min MHz:                     1000.0000
BogoMIPS:                        108.00
Flags:                           fp asimd evtstrm aes pmull sha1
                                 sha2 crc32 atomics fphp asimdhp
                                 cpuid asimdrdm lrcpc dcpop asimddp
Linux raspberrypi 6.1.32-v8+ #1 SMP PREEMPT
Sat Aug  5 07:03:33 BST 2023 aarch64 GNU/Linux
```

The last count indicated that 31 different benchmarking and stress testing programs were run, producing hundreds of results included here. The devil is in the details.

Whetstone Benchmark below or Go To Start

Whetstone Benchmark - whetstonePi64g8 and g12
Vector Versions - Whetv64SPg8 and g12, whetvDP64g8 and g12

This has a number of simple programming loops, with the overall MWIPS rating dependent on floating point calculations. with no accessing of data in L2 cache or RAM.

Results are provided for the original scalar single precision (SP) version, along with those for single and double precision (DP) varieties of the vector version, originally written for use on the first Cray 1 supercomputer delivered to the UK. For more information see Pi 5 The Vector Processor later. Examination of the time used by the different tests shows that this can be dominated by those executing such as COS and EXP functions.

Pi 5/Pi 4 comparisons are provided for the gcc 8 scalar versions, indicting performance gains between 2.44 to 2.59 times for the three (MFLOPS) floating point tests and 2.79 on overall MWIPS. Performance of the Pi 5 gcc 12 compilations were essentially identical to those from gcc 8.

Pi 5/Pi 4 vector SP and DP gcc 8 performance gains were similar between 2.34 to 3.10 times for MFLOPS and around 2.3 for MWIPS. Pi 5 SP Vector/Scalar gains are also provided, giving 5.40 to 7.86 times for MFLOPS but only 1.88 times for overall MWIPS, deflated by the COS/EXP tests. Maximum SP scalar speed was 1.36 GFLOPS with vectors at 8.08 SP and 4.0 DP.

```
 Pi 4 GCC 8
 Whetstone Single Precision C Benchmark  64 Bit gcc 8R, Fri May 22 10:48:53 2020

 Loop content                  Result              MFLOPS      MOPS   Seconds

 N1 floating point     -1.12475013732910156       524.251              0.076
 N2 floating point     -1.12274742126464844       534.904              0.524
 N3 if then else        1.00000000000000000                2978.570    0.073
 N4 fixed point        12.00000000000000000                2493.078    0.264
 N5 sin,cos etc.        0.49911010265350342                  57.643    3.012
 N6 floating point      0.99999982118606567       397.676              2.831
 N7 assignments         3.00000000000000000                 996.647    0.387
 N8 exp,sqrt etc.       0.75110864639282227                  27.327    2.841

 MWIPS                                           2085.311             10.008

 Pi 5 GCC 8
 Whetstone Single Precision C Benchmark  64 Bit gcc 8R, Thu Aug 10 15:44:50 2023

 Loop content                  Result              MFLOPS      MOPS   Seconds  G8 Pi5/4

 N1 floating point     -1.12475013732910156      1279.196              0.087    2.44
 N2 floating point     -1.12274742126464844      1364.748              0.573    2.55
 N3 if then else        1.00000000000000000                7190.834    0.084    2.41
 N4 fixed point        12.00000000000000000                5995.954    0.306    2.41
 N5 sin,cos etc.        0.49911010265350342                 154.725    3.131    2.68
 N6 floating point      0.99999982118606567      1027.998              3.055    3.59
 N7 assignments         3.00000000000000000                2398.668    0.449    2.41
 N8 exp,sqrt etc.       0.75110864639282227                  93.596    2.314    3.43

 MWIPS                                           5822.922              9.998    2.79

 Pi 5 GCC 12
  Whetstone Single Precision C Benchmark  64 Bit gcc 12, Thu Sep 28 11:46:43 2023

 Loop content                  Result              MFLOPS      MOPS   Seconds

 N1 floating point     -1.12475013732910156      1279.140              0.088
 N2 floating point     -1.12274742126464844      1364.558              0.575
 N3 if then else        1.00000000000000000                3594.939    0.168
 N4 fixed point        12.00000000000000000                5994.963    0.307
 N5 sin,cos etc.        0.49911010265350342                 157.996    3.075
 N6 floating point      0.99999982118606567      1027.940              3.064
 N7 assignments         3.00000000000000000                2398.054    0.450
 N8 exp,sqrt etc.       0.75110864639282227                  95.590    2.273

 MWIPS                                          5839.767             10.000

 #################### Vector Whetstone Vecton Length 258 ####################

 Pi 4 GCC 8 SP
 Whetstone Vector Benchmark 64 Bit Single Precision, Wed Aug 30 10:41:57 2023

 Loop content                  Result              MFLOPS      MOPS   Seconds

 N1 floating point     -1.13316142559051514      2338.496              0.391
 N2 floating point     -1.13312149047851562      1651.957              3.877
 N3 if then else        1.00000000000000000                4427.445    1.114
 N4 fixed point        12.00000000000000000                1733.458    8.659
 N5 sin,cos etc.        0.49998238682746887                  74.913   52.923
 N6 floating point      0.99999982118606567      2573.346              9.988
 N7 assignments         3.00000000000000000               18596.381    0.474
 N8 exp,sqrt etc.       0.75002217292785645                  78.503   22.581

 MWIPS                                           4764.843            100.007
```

Continued below

Continued from above - Note different single and double precision numeric results.

Pi 5 GCC 8 SP
Whetstone Vector Benchmark 64 Bit Single Precision, Sat Oct 7 10:15:16 2023

Loop content	Result	MFLOPS	MOPS	Seconds	G8 Pi5/4
N1 floating point	-1.13316142559051514	7111.676		0.290	3.04
N2 floating point	-1.13312149047851562	3857.446		3.746	2.34
N3 if then else	1.00000000000000000		10141.446	1.097	2.29
N4 fixed point	12.00000000000000000		2396.242	14.135	1.38
N5 sin,cos etc.	0.49998238682746887		177.032	50.534	2.36
N6 floating point	0.99999982118606567	7986.011		7.263	3.10
N7 assignments	3.00000000000000000		42584.598	0.467	2.29
N8 exp,sqrt etc.	0.75002217292785645		178.102	22.459	2.27
MWIPS		10753.538		99.990	2.26

Pi 5 GCC 12 SP
Whetstone Vector Benchmark gcc 12 64 Bit Single Precision, Sat Oct 7 10:46:30 2023

Loop content	Result	MFLOPS	MOPS	Seconds	Pi 5 GCC12/8	Vector/ Scalar G12 Pi5
N1 floating point	-1.13316142559051514	7393.282		0.286	1.04	5.78
N2 floating point	-1.13312149047851562	7364.751		2.009	1.91	5.40
N3 if then else	1.00000000000000000		14169.053	0.804	1.40	3.94
N4 fixed point	12.00000000000000000		2398.742	14.457	1.00	0.40
N5 sin,cos etc.	0.49998238682746887		177.260	51.673	1.00	1.12
N6 floating point	0.99999982118606567	8078.622		7.351	1.91	7.86
N7 assignments	3.00000000000000000		26419.105	0.770	0.62	11.02
N8 exp,sqrt etc.	0.75002217292785645		178.359	22.961	1.00	1.87
MWIPS		10974.928		100.311	1.02	1.88

Pi 4 GCC 8 DP
Whetstone Vector Benchmark 64 Bit Double Precision, Wed Aug 30 10:48:05 2023

Loop content	Result	MFLOPS	MOPS	Seconds
N1 floating point	-1.13314558088707962	1146.624		0.709
N2 floating point	-1.13310306766606850	1094.230		5.203
N3 if then else	1.00000000000000000		4405.221	0.995
N4 fixed point	12.00000000000000000		1730.427	7.711
N5 sin,cos etc.	0.49998080312723675		73.193	48.149
N6 floating point	0.99999988868927014	1294.129		17.655
N7 assignments	3.00000000000000000		9967.123	0.785
N8 exp,sqrt etc.	0.75002006515491115		83.614	18.845
MWIPS		4233.571		100.052

Pi 5 GCC 8 DP
Whetstone Vector Benchmark 64 Bit Double Precision, Sat Oct 7 10:18:59 2023

Loop content	Result	MFLOPS	MOPS	Seconds	G8 Pi5/4
N1 floating point	-1.13314558088707962	3499.307		0.535	3.05
N2 floating point	-1.13310306766606850	2793.370		4.688	2.55
N3 if then else	1.00000000000000000		10158.471	0.993	2.31
N4 fixed point	12.00000000000000000		2396.163	12.809	1.38
N5 sin,cos etc.	0.49998080312723675		171.834	47.176	2.35
N6 floating point	0.99999988868927014	3994.760		13.156	3.09
N7 assignments	3.00000000000000000		21713.754	0.829	2.18
N8 exp,sqrt etc.	0.75002006515491115		184.857	19.607	2.21
MWIPS		9763.593		99.793	2.31

Pi 5 GCC 12 DP
Whetstone Vector Benchmark gcc 12 64 Bit Double Precision, Sat Oct 7 10:50:40 2023

Loop content	Result	MFLOPS	MOPS	Seconds
N1 floating point	-1.13314558088707962	3602.841		0.523
N2 floating point	-1.13310306766606739	3619.564		3.647
N3 if then else	1.00000000000000000		14167.623	0.718
N4 fixed point	12.00000000000000000		2398.696	12.898
N5 sin,cos etc.	0.49998080312723675		172.068	47.491
N6 floating point	0.99999988868927014	3997.801		13.252
N7 assignments	3.00000000000000000		13172.392	1.378
N8 exp,sqrt etc.	0.75002006515491115		182.557	20.014
MWIPS		9829.517		99.920

Dhrystone Benchmark below or Go To Start

Dhrystone Benchmark - dhrystonePi64g8 and g12

This is the most popular ARM integer benchmark, often subject to over optimisation, rated in VAX MIPS aka DMIPS.

Pi 5 GCC 8 gain over Pi 4 was 2.37 times. There was a slight gain using GCC 12, where DMIPS/MHz ratio reached 8.57.

```
Pi 4 GCC 8
Dhrystone Benchmark 2.1 64 Bit gcc8, Mon May 25 22:16:05 2020

Nanoseconds one Dhrystone run:        72.83
Dhrystones per Second:             13729822
VAX MIPS rating =                   7814.36

Numeric results were correct


Pi 5 GCC 8
Dhrystone Benchmark 2.1 64 Bit gcc8, Thu Aug 10 15:49:13 2023

Nanoseconds one Dhrystone run:        30.69
Dhrystones per Second:             32578833
VAX MIPS rating =                  18542.31   Pi 5/Pi 4 Gain 2.37

Numeric results were correct


Pi 5 GCC 12
Dhrystone Benchmark 2.1 64 Bit gcc12, Thu Sep 28 11:44:33 2023

Nanoseconds one Dhrystone run:        27.68
Dhrystones per Second:             36120831
VAX MIPS rating =                  20558.24   GCC 12/8 Gain 1.11

Numeric results were correct
```

Linpack 100 Benchmark below or Go To Start

Linpack 100 Benchmark MFLOPS - linpackPi64g8 and g12, linpackPi64gSP, linpackPi64NEONig8

This original Linpack benchmark executes double precision arithmetic. I introduced two single precision versions, one using NEON functions to include vector processing. Performance of this benchmark can vary, with its dependence on data placement in L2 cache.

Unlike when the Pi 5 was introduced. later compilers produced code as fast as the NEON version. Now with GCC 12, The NEON variety was slower and the others produced a small gain over GCC 8 compiations. Comparisons for the latter indicated Pi 5 gains were between 3.16 and 3.54 times over the three versions. Maximum Pi 5 speeds were 6.60 GFLOPS SP and 3.93 GFLOPS DP.

```
 Pi 4 GCC 8
 Linpack Double Precision Unrolled Benchmark n @ 100
 Optimisation 64 Bit gcc 8, Mon May 25 22:05:47 2020

 Speed    1111.51 MFLOPS

 Numeric results were as expected

 Linpack Single Precision Unrolled Benchmark n @ 100
 Optimisation 64 Bit gcc 8, Mon May 25 22:09:12 2020

 Speed    1930.27 MFLOPS

 Numeric results were as expected

 Linpack Single Precision Benchmark n @ 100
  NEON Intrinsics 64 bit gcc 8, Mon May 25 22:11:15 2020

 Speed    2030.95 MFLOPS

 Numeric results were as expected

------------------------------------------------------
 Pi 5 GCC 8                                                Pi5/Pi4
 Linpack Double Precision Unrolled Benchmark n @ 100
 Optimisation 64 Bit gcc 8, Thu Aug 10 16:12:47 2023

 Speed    3933.38 MFLOPS                                     3.54

 Numeric results were as expected

 Linpack Single Precision Unrolled Benchmark n @ 100
 Optimisation 64 Bit gcc 8, Thu Aug 10 16:04:18 2023

 Speed    6106.68 MFLOPS                                     3.16

 Numeric results were as expected

 Linpack Single Precision Benchmark n @ 100
  NEON Intrinsics 64 bit gcc 8, Thu Aug 10 16:13:52 2023

 Speed    6603.58 MFLOPS                                     3.25

 Numeric results were as expected

------------------------------------------------------
 Pi 5 GCC 12                                                GCC 12/5
 Linpack Double Precision Unrolled Benchmark n @ 100
 Optimisation 64 Bit gcc 12, Thu Sep 28 15:58:07 2023

 Speed    4136.39 MFLOPS                                     1.05

 Numeric results were as expected

 Linpack Single Precision Unrolled Benchmark n @ 100
 Optimisation 64 Bit gcc 12, Thu Sep 28 16:04:19 2023

 Speed    6472.77 MFLOPS                                     1.06

 Numeric results were as expected

 Linpack Single Precision Benchmark n @ 100
  NEON Intrinsics 64 bit gcc 12, Thu Sep 28 15:49:56 2023

 Speed    5665.39 MFLOPS                                     0.86

 Numeric results were as expected
 But 4 needed changing in program, via #define GCC12ARM64N,
 to avoid unnecessary error reports.
```

Livermore Loops Benchmark below or Go To Start

Livermore Loops Benchmark MFLOPS - liverloopsPi64g8 and g12

This benchmark measures performance of 24 double precision kernels, initially used to select the latest supercomputer. The official average is geometric mean, where Cray 1 supercomputer was rated as 11.9 MFLOPS. Following are MFLOPS for the individual kernels, followed by overall scores. Although each kernel is executed for a relatively long time, performance of some can be inconsistent.

Pi 5 GCC 8 maximum speed was 9.87 DP GFLOPS, with gains over the Pi 4 between 2.14 and 4.65 over the 24 loops.

Maximum performance via GCC 12 was 10.57 DP GFLOPS, with those for all of the loops similar to GCC 8 scores.

```
 Pi 4 GCC 8
 Livermore Loops Benchmark 64 Bit gcc 8 via C/C++ Mon May 25 10:39:10 2020

 MFLOPS for 24 loops
 2108.4  936.3  959.9  965.1  382.5  808.6 2312.9 2488.4 2065.7  668.7  500.3  980.7
  180.7  404.8  815.0  643.8  726.8 1189.6  449.8  397.2 1716.0  366.9  817.7  312.7

 Overall Ratings
 Maximum Average Geomean Harmean Minimum
  2616.7   959.8   766.7   613.0   169.7

 Numeric results were as expected


 Pi 5 GCC 8
 Livermore Loops Benchmark 64 Bit gcc 8 via C/C++ Thu Aug 10 16:14:33 2023

 MFLOPS for 24 loops
 7423.6 2147.9 2356.6 2472.9  911.5 1871.0 9872.3 5317.7 5162.9 2125.8 1173.2 2672.0
  709.1 1108.7 2966.6 1598.5 1761.3 5526.8 1190.0  956.0 5425.1 1489.5 2147.9  858.2

 Overall Ratings
 Maximum Average Geomean Harmean Minimum
  9872.3  2873.9  2208.3  1763.4   646.6

 Numeric results were as expected

-----------------------------------------------------------------------------------
 GCC 8 Pi5/Pi4 Performance Ratios
 For 24 loops
   3.52   2.29   2.46   2.56   2.38   2.31   4.27   2.14   2.50   3.18   2.34   2.72
   3.92   2.74   3.64   2.48   2.42   4.65   2.65   2.41   3.16   4.06   2.63   2.74
   Min    2.14   Max    4.65

 Overall Ratings
 Maximum Average Geomean Harmean Minimum
    3.77    2.99    2.88    2.88    3.81

-----------------------------------------------------------------------------------
 Pi 5 GCC 12
 Livermore Loops Benchmark 64 Bit gcc 12 via C/C++ Thu Sep 28 16:38:37 2023

 MFLOPS for 24 loops
 7833.8 2404.6 2377.2 2346.8  913.0 1857.1  10577 5350.6 5109.2 2117.4 1186.0 2351.4
  760.0 1121.2 3103.4 1597.7 1776.1 5455.9 1197.2 2490.5 5657.5 1855.7 2139.8  780.4

 Overall Ratings
 Maximum Average Geomean Harmean Minimum
 10576.9  2964.4  2308.1  1870.7   733.9

 Numeric results were as expected via #define GCC12ARMPI
```

Fast Fourier Transforms Benchmarks below or Go To Start

Fast Fourier Transforms Benchmarks - fft1Pi64g, fft3cPi64g8 and g12

This is a real application provided by my collaborator at Compuserve Forum. There are two benchmarks. The first one is the original C program. The second is an optimised version, originally using my x86 assembly code, but translated back into C code, making use of the partitioning and (my) arrangement to optimise for burst reading from RAM. Three measurements use both single and double precision data, calculating FFT sizes between 1K and 1024K, with data from caches and RAM. Note that steps in performance levels occur at data size changes between caches, then to RAM.

Comparisons of averages of the three runs are provided. Those for FFT1 demonstrate the clear and different advantage of the Pi 5 over the Pi 4, depending on the source of the data, with that from L3 cache providing gains of up to 13.34 times and up to 4.71 times involving the larger L2 cache. Most other gains are in the two to four times range. With the faster CPU speed limited FFT3c, gains were mainly mbetween 2 and 3 times. GCC 12 over GCC 8 comparisons indicate a slight advantage of the former using data from caches, but the role reversed, dealing with RAM data transfers.

Pi 4 GCC 8

Pi 4 RPi FFT gcc 8 64 Bit Benchmark 1 Mon May 25 10:54:42 2020

Size	milliseconds					
K	Single Precision			Double Precision		
1	0.05	0.04	0.04	0.04	0.04	0.05
2	0.08	0.08	0.08	0.15	0.14	0.14
4	0.23	0.23	0.23	0.39	0.38	0.44
8	0.73	0.80	0.70	0.97	1.04	0.97
16	1.98	1.87	1.79	2.66	2.52	2.83
32	4.92	4.92	5.29	5.67	4.92	4.89
64	8.80	8.69	8.67	32.21	32.23	33.31
128	49.82	49.79	50.17	161.36	159.61	159.39
256	295.55	280.43	303.20	411.97	415.90	340.34
512	506.01	601.29	572.36	781.10	779.05	782.21
1024	1375.42	1377.64	1375.77	1898.28	1876.88	1896.22

1024	Square Check	Maximum Noise	Average Noise
SP	9.999520e-01	3.346482e-06	4.565234e-11
DP	1.000000e+00	1.133294e-23	1.428110e-28

End at Mon May 25 10:55:00 2020

Pi 4 RPi FFT gcc 8 64 Bit Benchmark 3c.0 Mon May 25 10:56:49 2020

Size	milliseconds					
K	Single Precision			Double Precision		
1	0.06	0.04	0.04	0.04	0.04	0.03
2	0.09	0.07	0.07	0.10	0.10	0.10
4	0.23	0.20	0.20	0.23	0.26	0.23
8	0.50	0.44	0.46	0.52	0.50	0.50
16	1.21	1.19	1.05	1.23	1.17	1.19
32	2.36	2.23	2.18	3.33	3.32	3.29
64	6.16	5.70	5.31	10.20	10.20	10.18
128	16.39	15.69	15.69	24.35	24.45	24.48
256	38.70	37.46	37.40	54.57	54.65	54.59
512	83.83	80.96	81.40	119.71	118.70	119.27
1024	182.08	176.05	176.97	268.43	259.16	259.30

1024	Square Check	Maximum Noise	Average Noise
SP	9.999520e-01	3.346482e-06	4.565234e-11
DP	1.000000e+00	1.133294e-23	1.428110e-28

End at Mon May 25 10:56:52 2020

Pi 5 GCC 8

Pi 5 RPi FFT gcc 8 64 Bit Benchmark 1 Fri Aug 11 16:47:11 2023

Size	milliseconds						Average Pi5/Pi4	
K	Single Precision			Double Precision			SP	DP
1	0.02	0.02	0.02	0.02	0.02	0.02	2.20	2.51
2	0.04	0.04	0.04	0.04	0.04	0.04	1.98	3.81
4	0.09	0.09	0.09	0.09	0.09	0.09	2.64	4.71
8	0.19	0.20	0.19	0.29	0.29	0.29	3.88	3.48
16	0.56	0.56	0.56	0.65	0.67	0.78	3.35	3.82
32	1.30	1.27	1.29	1.55	1.50	1.80	3.92	3.18
64	3.18	3.00	2.99	4.16	3.90	3.91	2.85	8.17
128	7.76	7.30	7.28	14.27	14.44	13.71	6.70	11.33
256	23.23	21.27	21.40	99.92	94.38	94.97	13.34	4.04
512	157.82	152.33	173.93	329.15	321.16	323.41	3.47	2.41
1024	608.66	606.77	600.94	1069.84	1048.00	1049.41	2.27	1.79

1024	Square Check	Maximum Noise	Average Noise
SP	9.999520e-01	3.346482e-06	4.565234e-11
DP	1.000000e+00	1.133294e-23	1.428110e-28

End at Fri Aug 11 16:47:19 2023

Continued below

Pi 5 RPi FFT gcc 8 64 Bit Benchmark 3c.0 Fri Aug 11 16:48:27 2023

Size	milliseconds						Average Pi5/Pi4	
K	Single Precision			Double Precision			SP	DP
1	0.03	0.02	0.02	0.02	0.02	0.02	1.88	1.96
2	0.05	0.04	0.04	0.04	0.04	0.04	1.93	2.61
4	0.10	0.08	0.08	0.09	0.09	0.09	2.37	2.74
8	0.21	0.18	0.18	0.23	0.21	0.21	2.43	2.37
16	0.45	0.41	0.41	0.53	0.48	0.49	2.70	2.40
32	1.16	0.90	0.93	1.22	1.07	1.06	2.27	2.97
64	2.39	2.04	2.39	2.98	2.76	2.69	2.52	3.63
128	5.26	4.82	4.86	9.92	9.90	9.86	3.20	2.47
256	14.58	13.92	13.89	29.15	27.71	26.90	2.68	1.96
512	42.03	39.73	39.84	72.71	72.32	71.70	2.02	1.65
1024	101.56	99.35	98.31	176.62	171.45	175.48	1.79	1.50

1024	Square Check	Maximum Noise	Average Noise
SP	9.999520e-01	3.346482e-06	4.565234e-11
DP	1.000000e+00	1.133294e-23	1.428110e-28

End at Fri Aug 11 16:48:29 2023

Pi 5 GCC 12

RPi FFT gcc 12 64 Bit Benchmark 1 Thu Sep 28 19:10:33 2023

Size	milliseconds						Average GCC 12/8	
K	Single Precision			Double Precision			SP	DP
1	0.02	0.02	0.02	0.02	0.02	0.02	1.15	1.02
2	0.06	0.04	0.04	0.04	0.04	0.04	0.92	1.05
4	0.08	0.08	0.08	0.08	0.08	0.08	1.09	1.05
8	0.18	0.18	0.18	0.80	0.26	0.25	1.09	0.65
16	0.55	0.62	0.61	0.78	0.62	0.68	0.95	1.01
32	1.19	1.19	1.18	3.14	1.66	2.23	1.08	0.69
64	2.90	2.87	3.12	4.14	3.83	4.62	1.03	0.95
128	8.01	7.72	8.41	19.04	16.31	19.17	0.93	0.78
256	28.65	29.22	30.38	142.81	143.44	144.91	0.75	0.67
512	256.41	209.11	215.07	400.84	410.99	448.06	0.71	0.77
1024	798.30	749.85	753.61	1073.95	1075.09	1051.38	0.79	0.99

1024	Square Check	Maximum Noise	Average Noise
SP	9.999520e-01	3.346482e-06	4.565234e-11
DP	1.000000e+00	1.133294e-23	1.428110e-28

End at Thu Sep 28 19:10:41 2023

RPi FFT gcc 12 64 Bit Benchmark 3c.0 Thu Sep 28 19:13:51 2023

Size	milliseconds						Average GCC 12/8	
K	Single Precision			Double Precision			SP	DP
1	0.02	0.02	0.02	0.02	0.02	0.02	1.20	1.06
2	0.04	0.04	0.04	0.04	0.04	0.04	1.04	1.06
4	0.09	0.08	0.08	0.08	0.08	0.08	1.06	1.06
8	0.19	0.18	0.18	0.20	0.19	0.19	1.06	1.10
16	0.41	0.39	0.39	0.46	0.43	0.43	1.07	1.12
32	0.88	0.85	0.86	1.01	0.96	0.96	1.15	1.14
64	1.98	1.91	1.91	2.57	2.48	2.47	1.17	1.12
128	5.65	4.68	4.63	10.10	10.04	10.06	1.00	0.98
256	14.59	14.50	14.59	36.02	35.29	34.84	0.97	0.79
512	55.50	54.91	55.79	100.99	102.62	99.96	0.73	0.71
1024	143.39	142.49	143.22	231.27	228.44	229.17	0.70	0.76

1024	Square Check	Maximum Noise	Average Noise
SP	9.999520e-01	3.346482e-06	4.565234e-11
DP	1.000000e+00	1.133294e-23	1.428110e-28

End at Thu Sep 28 19:13:53 2023

BusSpeed Benchmark below or Go To Start

BusSpeed Benchmark - busspeedPi64g8 and g12

This is a read only benchmark with data from caches and RAM. The program reads one word with 32 word increments for the next one, skipping following data word by decreasing increments. finally reading all data. This shows where data is read in bursts, enabling estimates being made of bus speeds, as 16 times the speed of appropriate measurements at Inc16.

The most important ratios are from Read All, others demonstrating when all data is not being read sequentially and the Pi 5 appears to be significantly faster than the Pi 4. The main results indicate Pi 5 gains of just over twice reading data from L1 and L2 caches, but can be more than four times from L3 and more than three times from RAM. Maximum bus speed, using one CPU core, is estimated as around 14 GB/second from Inc16 also shown under Read All. See MP results for higher estimates.

Pi 5 performance produced from GCC 8 and GCC 12 compilations was essentially the same.

Pi 4 GCC 8

BusSpeed 64 Bit gcc 8 Mon May 25 22:13:11 2020

Reading Speed 4 Byte Words in MBytes/Second

Memory KBytes	Inc32 Words	Inc16 Words	Inc8 Words	Inc4 Words	Inc2 Words	Read All	Cache	Pi 5
16	4898	5109	5626	5860	5879	9238	L1	L1
32	1109	1389	2485	3804	5026	8435		
64	804	1030	2025	3285	4871	8312	L2 Shared	
128	737	951	1877	3130	4908	8556		L2
256	732	953	1897	3147	4941	8617		
512	701	939	1766	2902	4601	8150		
1024	323	494	986	1807	3060	5553	RAM	L3 Shared
4096	242	259	486	964	1932	3856		RAM
16384	236	268	493	971	1939	3878		
65536	242	271	494	973	1942	3884		

End of test Mon May 25 22:13:21 2020

Pi 5 GCC 8 — P5/P4 Comparison

BusSpeed 64 Bit gcc 8 Fri Aug 11 16:46:13 2023

Reading Speed 4 Byte Words in MBytes/Second

Memory KBytes MP-bus	Inc32 Words	Inc16 Words	Inc8 Words	Inc4 Words	Inc2 Words	Read All	Inc32 Words	Inc16 Words	Inc8 Words	Inc4 Words	Inc2 Words	Read All
16	8300	8413	15451	17849	18151	18721	1.69	1.65	2.75	3.05	3.09	2.03
32	9159	9235	15509	17911	18132	18721	8.26	6.65	6.24	4.71	3.61	2.22
64	7460	7644	13739	17008	17665	18593	9.28	7.42	6.78	5.18	3.63	2.24
128	2375	4452	7168	11555	13968	18203	3.22	4.68	3.82	3.69	2.85	2.13
256	2375	4425	7225	11540	13964	18243	3.24	4.64	3.81	3.67	2.83	2.12
512	1784	2980	5758	10362	13685	18203	2.54	3.17	3.26	3.57	2.97	2.23
1024	1225	2325	4639	9336	13467	18281	3.79	4.71	4.70	5.17	4.40	3.29
4096	656	1375	2700	5120	9599	15984	2.71	5.31	5.56	5.31	4.97	4.15
16384	579	864	1741	3502	7020	14015	2.45	3.22	3.53	3.61	3.62	3.61
65536	604	796	1595	3195	6351	12699	2.50	2.94	3.23	3.28	3.27	3.27

End of test Fri Aug 11 16:46:22 2023

Pi 5 GCC 12 — Pi 5 GCC 12/8 Comparison

BusSpeed 64 Bit gcc 12 Thu Sep 28 19:02:33 2023

Reading Speed 4 Byte Words in MBytes/Second

Memory KBytes	Inc32 Words	Inc16 Words	Inc8 Words	Inc4 Words	Inc2 Words	Read All	Inc32 Words	Inc16 Words	Inc8 Words	Inc4 Words	Inc2 Words	Read All
16	8493	8509	16377	17918	18170	18733	1.02	1.01	1.06	1.00	1.00	1.00
32	9127	9295	16478	18023	18212	18740	1.00	1.01	1.06	1.01	1.00	1.00
64	7530	7604	14030	17241	17877	18603	1.01	0.99	1.02	1.01	1.01	1.00
128	2375	4189	7212	11566	13961	18230	1.00	0.94	1.01	1.00	1.00	1.00
256	2358	4275	7265	11595	13985	18274	0.99	0.97	1.01	1.00	1.00	1.00
512	1557	2879	5524	10229	13877	18231	0.87	0.97	0.96	0.99	1.01	1.00
1024	1225	2339	4606	9318	13902	18271	1.00	1.01	0.99	1.00	1.03	1.00
4096	780	1387	2672	5115	9407	16053	1.19	1.01	0.99	1.00	0.98	1.00
16384	652	880	1763	3479	7034	13979	1.13	1.02	1.01	0.99	1.00	1.00
65536	624	801	1605	3178	6416	12800	1.03	1.01	1.01	0.99	1.01	1.01

MemSpeed Benchmark below or Go To Start

MemSpeed Benchmark MB/Second - memspeedPi64g8 and g12

The benchmark includes CPU speed dependent calculations using data from caches and RAM, via single and double precision floating point and integer functions. The instruction sequences used are shown in the results column titles.

When compiled with GCC 6, earlier results identified unusual slow operation dealing with 32 bit floating point and integer calculations. This looks as though the effect is to read data from RAM instead of caches, and why Pi 5 performance gains were mainly less than two times. With double precision floating point, average Pi 5 gains were around four times for the first two sets of calculations, including more that 10 times with L3 cache involvement.

The GCC 12 compilation appears to have corrected the above misoperations, providing gains of more than eight times over GCC 8. These calculations also show slight improvements in double precision calculations. Maximum calculated speeds are provided, indicating 15.3 single core GFLOPS SP and 6.86 DP, the relationship expected using SIMD calculations. The tests also confirmed this with the near 6.4 GFLOPS/GHz SP and near half that DP. This performance was obtained using data from L1 and L2 caches with almost that from L3 cache.

Pi 4 GCC 8

Memory Reading Speed Test 64 Bit gcc 8 by Roy Longbottom

Start of test Mon May 25 22:23:53 2020

Memory	x[m]=x[m]+s*y[m] Int+			x[m]=x[m]+y[m]			x[m]=y[m]		
KBytes	Dble	Sngl	Int32	Dble	Sngl	Int32	Dble	Sngl	Int32
Used	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S
8	15531	3999	3957	15576	4387	4358	11629	9313	9314
16	15717	3992	3922	15770	4355	4377	11799	9444	9446
32	12020	3818	3814	12043	4179	4198	11549	9496	9497
64	12228	3816	3887	12220	4166	4195	8935	8506	8506
128	12265	3869	3941	12157	4182	4206	8080	8193	8196
256	12230	3873	3932	12073	4199	4216	8129	8224	8223
512	9731	3832	3902	9709	4150	4171	8029	7845	7865
1024	3772	3682	3769	3467	3887	3920	5478	5543	5378
2048	1896	3463	3496	1886	3616	3612	2937	2945	2923
4096	1924	3520	3528	1933	3651	3394	2752	2796	2785
8192	1996	3523	3555	1988	3643	3630	2668	2661	2663

End of test Mon May 25 22:24:10 2020

Pi 5 GCC 8

Memory Reading Speed Test 64 Bit gcc 8 by Roy Longbottom

Start of test Fri Aug 11 16:34:06 2023

Memory	x[m]=x[m]+s*y[m] Int+			x[m]=x[m]+y[m]			x[m]=y[m]		
KBytes	Dble	Sngl	Int32	Dble	Sngl	Int32	Dble	Sngl	Int32
Used	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S
8	50862	6851	6746	50686	7193	7490	37629	18595	25168
16	51032	6820	6717	51024	7164	7468	38002	18888	24946
32	49985	6814	6676	50568	7150	7446	37609	18972	25259
64	50868	6857	6656	50864	7168	7411	37799	19114	25426
128	32618	6797	6670	32666	7142	7278	35466	19143	25439
256	32540	6788	6640	32744	7183	7278	34821	19144	25360
512	26949	6786	6668	30112	7155	7246	33493	14598	16816
1024	25094	6719	6645	19272	6821	7206	21805	17292	22671
2048	20586	6365	6586	19261	6887	7172	4740	4662	13673
4096	5004	6680	6710	4963	6776	6249	7938	8990	8797
8192	3229	5589	4662	3205	6496	6573	6654	6719	4613

End of test Fri Aug 11 16:34:22 2023

P5/P4 Comparison

Memory	x[m]=x[m]+s*y[m] Int+			x[m]=x[m]+y[m]			x[m]=y[m]		
KBytes	Dble	Sngl	Int32	Dble	Sngl	Int32	Dble	Sngl	Int32
Used	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S
8	3.27	1.71	1.70	3.25	1.64	1.72	3.24	2.00	2.70
16	3.25	1.71	1.71	3.24	1.65	1.71	3.22	2.00	2.64
32	4.16	1.78	1.75	4.20	1.71	1.77	3.26	2.00	2.66
64	4.16	1.80	1.71	4.16	1.72	1.77	4.23	2.25	2.99
128	2.66	1.76	1.69	2.69	1.71	1.73	4.39	2.34	3.10
256	2.66	1.75	1.69	2.71	1.71	1.73	4.28	2.33	3.08
512	2.77	1.77	1.71	3.10	1.72	1.74	4.17	1.86	2.14
1024	6.65	1.82	1.76	5.56	1.75	1.84	3.98	3.12	4.22
2048	10.86	1.84	1.88	10.21	1.90	1.99	1.61	1.58	4.68
4096	2.60	1.90	1.90	2.57	1.86	1.84	2.88	3.22	3.16
8192	1.62	1.59	1.31	1.61	1.78	1.81	2.49	2.52	1.73

Continued below

Pi 5 GCC 12

Memory Reading Speed Test 64 Bit gcc 12 by Roy Longbottom

Start of test Thu Sep 28 18:54:28 2023

Memory KBytes Used	x[m]=x[m]+s*y[m] Int+ Dble MB/S	Sngl MB/S	Int32 MB/S	x[m]=x[m]+y[m] Dble MB/S	Sngl MB/S	Int32 MB/S	x[m]=y[m] Dble MB/S	Sngl MB/S	Int32 MB/S
8	54902	61264	65610	55241	65554	63848	37768	25475	25486
16	54803	60539	64671	55169	64700	64750	38078	24891	24891
32	51859	60967	64278	52558	65247	65275	37520	25234	25234
64	52597	61169	65523	52485	65514	65523	37945	25408	25402
128	33580	60278	63742	33647	63692	62897	37218	25370	25457
256	33724	60317	63873	33711	63840	63865	35555	25371	25375
512	33522	59194	63298	33502	63259	63175	35909	25459	25451
1024	32078	57946	60718	31576	60680	59199	26110	22319	23059
2048	29249	55376	57648	29028	57558	57290	16245	18242	19514
4096	4508	11981	11906	4864	11894	9313	10254	10529	10668
8192	3175	6507	6150	3178	6441	6499	6678	6904	6364
Max MFLOPS	6862	15316							

End of test Thu Sep 28 18:54:43 2023

Pi 5 GCC 12/8

Memory KBytes Used	x[m]=x[m]+s*y[m] Int+ Dble MB/S	Sngl MB/S	Int32 MB/S	x[m]=x[m]+y[m] Dble MB/S	Sngl MB/S	Int32 MB/S	x[m]=y[m] Dble MB/S	Sngl MB/S	Int32 MB/S
8	1.08	8.94	9.73	1.09	9.11	8.52	1.00	1.37	1.01
16	1.07	8.88	9.63	1.08	9.03	8.67	1.00	1.32	1.00
32	1.04	8.95	9.63	1.04	9.13	8.77	1.00	1.33	1.00
64	1.03	8.92	9.84	1.03	9.14	8.84	1.00	1.33	1.00
128	1.03	8.87	9.56	1.03	8.92	8.64	1.05	1.33	1.00
256	1.04	8.89	9.62	1.03	8.89	8.78	1.02	1.33	1.00
512	1.24	8.72	9.49	1.11	8.84	8.72	1.07	1.74	1.51
1024	1.28	8.62	9.14	1.64	8.90	8.22	1.20	1.29	1.02
2048	1.42	8.70	8.75	1.51	8.36	7.99	3.43	3.91	1.43
4096	0.90	1.79	1.77	0.98	1.76	1.49	1.29	1.17	1.21
8192	0.98	1.16	1.32	0.99	0.99	0.99	1.00	1.03	1.38

NeonSpeed Benchmark below or Go To Start

NeonSpeed Benchmark MB/Second - NeonSpeedPi64g8 and g12

This carries out some of the same calculations as MemSpeed. All results are for 32 bit floating point and integer calculations. Norm functions were as generated by the compiler and NEON through using intrinsic functions.

The initial GCC 8 test functions produced the same irregular results as MemSpeed first “Normal Float and Int” calculations that appear to only read RAM based data. Performance from NEON code indicated that the Pi 5 was typically 2.5 times faster than the Pi 4, using cache based data, and 1.5 times from RAM. Exceptions were gains of up to 7.9 times using L3 cache and nearly 4.8 from lower level caches.

The GCC 12 compiler produced acceptable “Normal” performance on the Pi 5, reflected by gains of up to more than ten times over GCC 8 results. This compiler is also shown to provide faster operation than that from NEON functions. Many of the latter show 20% improvements but some were slower. Maximum floating point speed demonstrated was nearly 17 GFLOPS.

Pi 4 GCC 8
NEON Speed 64 Bit gcc 8 Mon May 25 22:21:51 2020
Vector Reading Speed in MBytes/Second

Memory	Float	v=v+s*v	Int	v=v+v+s	Neon	v=v+v
KBytes	Norm	Neon	Norm	Neon	Float	Int
16	3629	14987	3925	13643	14457	16642
32	3475	10933	3821	9970	11029	11055
64	3447	11749	3845	11098	11802	12079
128	3332	11392	3912	10813	11430	11513
256	3325	11565	3926	10981	11598	11699
512	3313	10553	3917	10269	10755	10740
1024	3239	3331	3737	3291	3302	3321
4096	2987	1888	3331	1777	1881	1878
16384	3150	1821	3347	1814	1812	1834
65536	2747	1954	3132	2017	1904	2021
Max MFLOPS		3747				

End of test Mon May 25 22:22:11 2020

Pi 5 GCC 8 P5/P4 Comparison
NEON Speed 64 Bit gcc 8 Fri Aug 11 16:44:52 2023

Vector Reading Speed in MBytes/Second

Memory	Float	v=v+s*v	Int	v=v+v+s	Neon	v=v+v	Float	v=v+s*v	Int	v=v+v+s	Neon	v=v+v
KBytes	Norm	Neon	Norm	Neon	Float	Int	Norm	Neon	Norm	Neon	Float	Int
16	6745	46851	6968	44490	46849	46847	1.86	3.13	1.78	3.26	3.24	2.81
32	6727	47104	6947	44618	47061	47056	1.94	4.31	1.82	4.48	4.27	4.26
64	6703	46642	6962	44166	47040	46955	1.94	3.97	1.81	3.98	3.99	3.89
128	6587	27383	6840	27199	27404	27398	1.98	2.40	1.75	2.52	2.40	2.38
256	6579	27491	6857	27299	27509	27509	1.98	2.38	1.75	2.49	2.37	2.35
512	6571	27433	6862	26599	24237	26163	1.98	2.60	1.75	2.59	2.25	2.44
1024	6531	26340	6756	25226	24597	24527	2.02	7.91	1.81	7.67	7.45	7.39
4096	6414	9410	6505	9986	9474	8835	2.15	4.98	1.95	5.62	5.04	4.70
16384	5690	2850	5501	2830	2865	2488	1.81	1.57	1.64	1.56	1.58	1.36
65536	4837	2534	4736	2458	2401	2450	1.76	1.30	1.51	1.22	1.26	1.21
Max MFLOPS		11776										

End of test Fri Aug 11 16:45:12 2023

Pi 5 GCC 12 Pi 5 GCC 12/8
NEON Speed 64 Bit gcc 12 Thu Sep 28 18:57:35

Vector Reading Speed in MBytes/Second

Memory	Float	v=v+s*v	Int	v=v+v+s	Neon	v=v+v	Float	v=v+s*v	Int	v=v+v+s	Neon	v=v+v
KBytes	Norm	Neon	Norm	Neon	Float	Int	Norm	Neon	Norm	Neon	Float	Int
16	67042	45164	67037	45358	54228	54166	9.94	0.96	9.62	1.02	1.16	1.16
32	67631	45190	67621	45415	53833	53675	10.05	0.96	9.73	1.02	1.14	1.14
64	67812	44856	67491	45171	52338	51321	10.12	0.96	9.69	1.02	1.11	1.09
128	62779	33147	64360	33074	33619	33458	9.53	1.21	9.41	1.22	1.23	1.22
256	64352	33405	64803	33187	33699	33719	9.78	1.22	9.45	1.22	1.23	1.23
512	61159	33171	61798	32263	33178	28319	9.31	1.21	9.01	1.21	1.37	1.08
1024	58937	32149	57732	31639	32219	32108	9.02	1.22	8.55	1.25	1.31	1.31
4096	9215	2639	7168	3800	3823	3776	1.44	0.28	1.10	0.38	0.40	0.43
16384	5546	2830	5592	2772	2753	2503	0.97	0.99	1.02	0.98	0.96	1.01
65536	4633	2445	4196	1922	2196	2294	0.96	0.96	0.89	0.78	0.91	0.94
Max MFLOPS	16953											

MultiThreading Benchmark next or Go To Start

MultiThreading Benchmarks

Most of the multithreading benchmarks execute the same calculations using 1, 2, 4 and 8 threads. One of them, MP-MFLOPS, is available in two different versions, using standard compiled “C” code for single and double precision arithmetic. A further version uses NEON intrinsic functions. Another variety uses OpenMP procedures for automatic parallelism.

MP-Whetstone Benchmark - MP-WHETSPi64g8 and g12

Multiple threads each run the eight test functions at the same time, but with some dedicated variables. Measured speed is based on the last thread to finish. Performance was generally proportional to the number of cores used. Overall seconds indicates MP efficiency, with around 5 seconds for 1, 2 and 4 threads, doubling with 8.

The Pi 5 CPU temperature reached 80.7°C within the 26 second testing time. Pi5/Pi4 4 thread performance ratios were between 2.22 and 3.43.

Performance of all GCC 8 compilations were essentially the same as those from GCC 12.

```
Pi 4 GCC 8
MP-Whetstone Benchmark 64 Bit gcc 8 Mon May 25 10:18:21 2020

                    Using 1, 2, 4 and 8 Threads

           MWIPS  MFLOPS  MFLOPS  MFLOPS     Cos     Exp   Fixpt      If   Equal
                       1       2       3    MOPS    MOPS    MOPS    MOPS    MOPS

1T        2146.7   530.1   530.1   397.2    60.5    27.3  7451.7  2240.2   998.1
2T        4290.4  1056.0  1055.3   794.0   120.9    54.7 14859.4  4488.5  1995.2
4T        8583.9  2115.8  2113.4  1590.5   241.8   109.3 29265.9  8940.7  3984.5
8T        8806.6  2676.0  2140.1  1627.3   244.8   113.0 37995.0 11565.4  4097.5

   Overall Seconds   5.00 1T,   5.01 2T,   5.02 4T,  10.10 8T

   All calculations produced consistent numeric results

Pi 5 GCC 8
MP-Whetstone Benchmark 64 Bit gcc 8 Mon Aug 14 10:09:58 2023

                    Using 1, 2, 4 and 8 Threads

           MWIPS  MFLOPS  MFLOPS  MFLOPS     Cos     Exp   Fixpt      If   Equal
                       1       2       3    MOPS    MOPS    MOPS    MOPS    MOPS

1T        6138.4  1278.2  1278.2  1020.4   174.1    94.8 17273.2  7033.6  2394.9
2T       12198.6  2542.8  2549.5  2029.7   344.4   188.4 35246.9 14307.3  4794.1
4T       24008.3  5013.1  4683.8  4045.3   674.5   374.4 69938.6 28558.3  9381.9
8T       24768.0  5170.6  5867.3  4080.9   693.9   385.9 74272.7 30002.8  9478.1

   Overall Seconds   5.00 1T,   5.04 2T,   5.22 4T,  10.37 8T

   All calculations produced consistent numeric results

P5/P4 Comparison
1T          2.86    2.41    2.41    2.57    2.88    3.47    2.32    3.14    2.40
2T          2.84    2.41    2.42    2.56    2.85    3.44    2.37    3.19    2.40
4T          2.80    2.37    2.22    2.54    2.79    3.43    2.39    3.19    2.35
8T          2.81    1.93    2.74    2.51    2.83    3.42    1.95    2.59    2.31

Pi 5 GCC 12
MP-Whetstone Benchmark 64 Bit gcc 12 Thu Sep 28 21:58:24 2023

                    Using 1, 2, 4 and 8 Threads

           MWIPS  MFLOPS  MFLOPS  MFLOPS     Cos    Exp   Fixpt      If    Equal
                       1       2       3    MOPS   MOPS    MOPS    MOPS     MOPS

1T        6180.4  1279.0  1273.5  1028.0   173.8    96.7 17586.5  7187.4  2396.5
2T       12353.4  2550.4  2556.9  2049.9   347.7   193.3 35875.6 14220.6  4796.8
4T       24647.0  5100.9  5078.2  4106.7   695.5   385.9 63256.4 28609.7  9549.0
8T       25053.6  5121.0  5293.6  4174.6   706.8   386.4 78259.8 31001.5  9658.4

   Overall Seconds   5.00 1T,   5.01 2T,   5.06 4T,  10.10 8T

Pi 5 GCC 12/8
1T          1.01    1.00    1.00    1.01    1.00    1.02    1.02    1.02    1.00
2T          1.01    1.00    1.00    1.01    1.01    1.03    1.02    0.99    1.00
4T          1.03    1.02    1.08    1.02    1.03    1.03    0.90    1.00    1.02
8T          1.01    0.99    0.90    1.02    1.02    1.00    1.05    1.03    1.02
```

MP-Dhrystone Benchmark next or Go To Start

MP-Dhrystone Benchmark - MP-DHRYPi64g8 and g12

This executes multiple copies of the same program, but with some shared data, leading to unacceptable multithreaded performance. Results are in VAX MIPS aka DMIPS.

Using the GCC 8 version, the Pi 5 performance was 2.27 times faster than the Pi 4, achieving 7.67 DMIPS/MHz. The GCC 12 compilation was slightly faster than the former, running on the Pi 5.

```
Pi 4 GCC 8
MP-Dhrystone Benchmark 64 Bit gcc 8 Tue May 26 11:41:49 2020

                    Using 1, 2, 4 and 8 Threads

Threads                        1       2       4       8
Seconds                     0.55    1.08    2.15     4.3
Dhrystones per Second    1.5E+07 1.5E+07 1.5E+07 1.5E+07
VAX MIPS rating             8271    8419    8478    8465

         Internal pass count correct all threads

         End of test Tue May 26 11:41:57 2020

Pi 5 GCC 8
MP-Dhrystone Benchmark 64 Bit gcc 8 Mon Aug 14 10:16:15 2023

                    Using 1, 2, 4 and 8 Threads

Threads                        1       2       4       8
Seconds                     0.62    1.88    4.18    8.45    Pi5/Pi4
Dhrystones per Second    3.2E+07 2.1E+07 1.9E+07 1.9E+07
VAX MIPS rating            18415   12137   10899   10771      2.27

         Internal pass count correct all threads

         End of test Mon Aug 14 10:16:31 2023

Pi 5 GCC 12
  MP-Dhrystone Benchmark 64 Bit gcc 12 Thu Sep 28 22:03:10 2023

                    Using 1, 2, 4 and 8 Threads

 Threads                        1        2        4        8
 Seconds                     0.57     1.95     4.31     8.70     Pi 5 GCC 12/8
 Dhrystones per Second   35046385 20477300 18570390 18398880
 VAX MIPS rating            19947    11655    10569    10472        1.08

         Internal pass count correct all threads

         End of test Thu Sep 28 22:03:26 2023
```

MP SP NEON Linpack Benchmark next or Go To Start

MP SP NEON Linpack Benchmark - linpackMPNeonPi64g8 and g12

This was produced to show that the original Linpack benchmark was completely unsuitable for benchmarking multiple CPUs or cores, and this is reflected in the results. The program uses NEON intrinsic functions, with increasing data sizes. Single core performance ratios are provided below for the three different memory array sizes that use N x N x 4 bytes or 40 KB, 1 MB and 4 MB. The three Pi 5/Pi 4 performance ratios were 2.94, 5.24, and 4.13 times. Maximum single core speed was 6.85 GFLOPS.

Two out of three of the new GCC 12 compilations produced slower performance on the Pi 5 and completely different numeric sumchecks.

```
Pi 4 GCC 8
Linpack Single Precision MultiThreaded Benchmark
 NEON Intrinsics 64 Bit gcc 8, Tue May 26 11:43:46 2020

   MFLOPS 0 to 4 Threads, N 100, 500, 1000

Threads    None        1       2       4

N  100   2167.70   91.82   89.65   89.96
N  500   1438.27  644.85  635.89  635.33
N 1000    394.99  376.97  383.92  384.19

 NR=norm resid RE=resid MA=machep X0=x[0]-1 XN=x[n-1]-1

 N              100             500            1000

 NR            1.97            5.40           13.51
 RE  4.69621336e-05  6.44138840e-04  3.22485110e-03
 MA  1.19209290e-07  1.19209290e-07  1.19209290e-07
 X0 -1.31130219e-05  5.79357147e-05 -3.08930874e-04
 XN -1.30534172e-05  3.51667404e-05  1.90019608e-04

Thread
 0 - 4 Same Results    Same Results    Same Results

Pi 5 GCC 8
Linpack Single Precision MultiThreaded Benchmark
 NEON Intrinsics 64 Bit gcc 8, Mon Aug 14 10:22:53 2023

   MFLOPS 0 to 4 Threads, N 100, 500, 1000

Threads    None        1       2       4    Pi5/Pi4

N  100   6375.62  154.59  151.48  150.82      2.94
N  500   7536.07 2250.75 2263.15 2222.61      5.24
N 1000   1631.94 1452.80 1401.29 1298.10      4.13

 NR=norm resid RE=resid MA=machep X0=x[0]-1 XN=x[n-1]-1

 N              100             500            1000

 NR            1.97            5.40           13.51
 RE  4.69621336e-05  6.44138840e-04  3.22485110e-03
 MA  1.19209290e-07  1.19209290e-07  1.19209290e-07
 X0 -1.31130219e-05  5.79357147e-05 -3.08930874e-04
 XN -1.30534172e-05  3.51667404e-05  1.90019608e-04

Thread
 0 - 4 Same Results    Same Results    Same Results

Pi 5 GCC 12
 Linpack Single Precision MultiThreaded Benchmark
 NEON Intrinsics 64 Bit gcc 12, Thu Sep 28 22:05:37 2023

   MFLOPS 0 to 4 Threads, N 100, 500, 1000

Threads    None        1       2       4  Pi 5 GCC 12/8

N  100   5461.61  169.27  176.25  174.14      0.86
N  500   6853.70 2538.16 2554.26 2562.31      0.91
N 1000   1741.83 1486.68 1493.84 1501.34      1.07

 NR=norm resid RE=resid MA=machep X0=x[0]-1 XN=x[n-1]-1

 N              100             500            1000

 NR            2.17            5.42            9.50
 RE  5.16722466e-05  6.46698638e-04  2.26586126e-03
 MA  1.19209290e-07  1.19209290e-07  1.19209290e-07
 X0 -2.38418579e-07 -5.54323196e-05 -1.26898289e-04
 XN -5.06639481e-06 -4.70876694e-06  1.41978264e-04

Thread
 0 - 4 Same Results    Same Results    Same Results
```

MP BusSpeed Benchmark below or Go To Start

MP BusSpeed (read only) Benchmark - MP-BusSpd2Pi64g8 and g12

For further details see the single core BusSpeed Benchmark that obtains the same order of GCC 8 results as the single thread of this MP version. For the latter, each thread exercises a dedicated segment of the data, circulated on a round robin basis, reading all data every pass.

Considering the most important GCC 8 Rdall tests, Pi5/Pi4 performance gains mainly approached three times for cache based data but multithreaded application showed gains up to 9.47 times. Highest gains of up to 18.17 times were in other areas. The high gains are due to improved caching on a read only basis.

The early Pi 4 GCC 12/8 comparisons indicated similar performance but increased progressively as more data was being read, reaching up to more than five times on RdAll. Here, single thread data transfer speeds reached nearly 68 GB/second and 4 thread up to 150 GB/second. This lead to me writing a new program New INTitHOT Integer Stress Test, where it is shown that GCC 12 produced highly efficient SIMD vector instructions.

Pi 4 GCC 8
MP-BusSpd 64 Bit gcc 8 Tue May 26 11:51:30 2020

MB/Second Reading Data, 1, 2, 4 and 8 Threads

KB	Inc32	Inc16	Inc8	Inc4	Inc2	RdAll
12.3	5168	5542	5641	4205	4095	4230
2T	8968	10728	10161	8110	8058	8368
4T	7874	13255	15586	13641	15485	16533
8T	8186	13386	15239	13469	14431	16372
122.9	598	927	1876	2792	3746	4059
2T	514	719	1538	4846	7596	8083
4T	486	933	2060	4126	8175	13690
8T	483	937	2059	4160	8166	13817
12288	224	257	488	964	1933	3579
2T	219	427	889	1832	3493	5371
4T	280	353	562	859	2168	3286
8T	229	230	527	1075	1880	4480

No Errors Found

End of test Tue May 26 11:51:43 2020

Pi 5 GCC 8 Pi 5/4 GCC 8
MP-BusSpd 64 Bit gcc 8 Mon Aug 14 10:37:37 2023

MB/Second Reading Data, 1, 2, 4 and 8 Threads

	Pi 5 GCC 8						Pi 5/4 GCC 8					
KB	Inc32	Inc16	Inc8	Inc4	Inc2	RdAll	Inc32	Inc16	Inc8	Inc4	Inc2	RdAll
12.3	9289	9450	15464	12578	12443	12073	1.80	1.71	2.74	2.99	3.04	2.85
2T	11465	15018	23403	20058	22357	22997	1.28	1.40	2.30	2.47	2.77	2.75
4T	8757	11343	21200	26582	32854	42575	1.11	0.86	1.36	1.95	2.12	2.58
8T	9036	8602	11448	17821	26795	30949	1.10	0.64	0.75	1.32	1.86	1.89
122.9	2358	4293	7257	11306	11657	11609	3.94	4.63	3.87	4.05	3.11	2.86
2T	4466	7819	13844	21220	23109	23119	8.69	10.87	9.00	4.38	3.04	2.86
4T	8831	14835	20781	42375	45809	44669	18.17	15.90	10.09	10.27	5.60	3.26
8T	7011	11818	19792	34990	39720	43742	14.52	12.61	9.61	8.41	4.86	3.17
12288	654	884	1585	3502	7243	10088	2.92	3.44	3.25	3.63	3.75	2.82
2T	726	743	1303	3454	7723	18286	3.32	1.74	1.47	1.89	2.21	3.40
4T	735	1551	1405	5166	10906	31106	2.63	4.39	2.50	6.01	5.03	9.47
8T	771	933	1486	3197	9182	18377	3.37	4.06	2.82	2.97	4.88	4.10

No Errors Found

End of test Mon Aug 14 10:37:49 2023

Pi 5 GCC 12 Pi 5 GCC 12/8
MP-BusSpd 64 Bit gcc 12 Thu Sep 28 22:11:28 2023

MB/Second Reading Data, 1, 2, 4 and 8 Threads

	Pi 5 GCC 12						Pi 5 GCC 12/8					
KB	Inc32	Inc16	Inc8	Inc4	Inc2	RdAll	Inc32	Inc16	Inc8	Inc4	Inc2	RdAll
12.3	9444	9504	16195	17543	27434	67773	1.02	1.01	1.05	1.39	2.20	5.61
2T	10884	14542	23738	28964	38304	92983	0.95	0.97	1.01	1.44	1.71	4.04
4T	10566	11790	21233	28439	44074	91129	1.21	1.04	1.00	1.07	1.34	2.14
8T	8657	10289	12122	19920	30038	45788	0.96	1.20	1.06	1.12	1.12	1.48
122.9	2380	4359	7261	11627	20970	44300	1.01	1.02	1.00	1.03	1.80	3.82
2T	4586	7699	13845	22597	40901	73723	1.03	0.98	1.00	1.06	1.77	3.19
4T	5469	10629	24698	38945	69318	150304	0.62	0.72	1.19	0.92	1.51	3.36
8T	6902	11176	19387	36720	64760	144651	0.98	0.95	0.98	1.05	1.63	3.31
12288	632	806	1838	3628	7366	13161	0.97	0.91	1.16	1.04	1.02	1.30
2T	961	711	1520	3527	5546	13012	1.32	0.96	1.17	1.02	0.72	0.71
4T	670	1566	3062	5403	13675	19563	0.91	1.01	2.18	1.05	1.25	0.63
8T	726	1117	2322	4747	9371	17111	0.94	1.20	1.56	1.48	1.02	0.93

MP RandMem Benchmark below or Go To Start

MP RandMem Benchmark - MP-RandMemPi64g8 and g12

The benchmark uses the same complex indexing for serial and random access, with separate read only and read/write tests. The performance patterns were as expected. Random access is dependent on the impact of burst reading and writing, producing those slow speeds. Read only performance increased, as expected, relative to the thread count, with that for read/write remaining constant at particular data size, probably due to write back to shared data space.

Again the new PI 5 caching arrangement produced high performance gains over the Pi 4, via GCC 8 compilations. In this case they were between 4 and 18 times. Others were between 2 and 3 times for cached based data and half that from RAM.

Performance from the GCC 12 version was little different to that from GCC 8.

```
 Pi 4 GCC 8
 MP-RandMem 64 Bit gcc 8 Tue May 26 11:53:37 2020

    MB/Second Using 1, 2, 4 and 8 Threads
     KB         SerRD  SerRW  RndRD  RndRW

   12.3    1T    5945   7898   5948   7895
           2T   11913   7937  11905   7929
           4T   23601   7875  23385   7867
           8T   23139   7777  23016   7770
  122.9    1T    5785   7090   2026   1977
           2T   10941   7074   1654   1968
           4T   10364   7052   1854   1970
           8T   10256   7031   1844   1973
  12288    1T    3861   1244    180    169
           2T    3793   1242    220    171
           4T    3941   1100    343    170
           8T    4065   1247    351    171
    No Errors Found
    End of test Tue May 26 11:54:20 2020
```

```
 Pi 4 GCC 8                                        Pi 5/4 GCC 8
 MP-RandMem 64 Bit gcc 8 Mon Aug 14 10:45:21 2023

    MB/Second Using 1, 2, 4 and 8 Threads
     KB         SerRD  SerRW  RndRD  RndRW         SerRD  SerRW  RndRD  RndRW

   12.3    1T   18593  18938  17858  17066          3.13   2.40   3.00   2.16
           2T   32655  18759  32998  16990          2.74   2.36   2.77   2.14
           4T   47087  18905  45181  17027          2.00   2.40   1.93   2.16
           8T   34725  18602  33955  17087          1.50   2.39   1.48   2.20
  122.9    1T   15501  16259  10950   9853          2.68   2.29   5.40   4.98
           2T   29970  16392  21177   9921          2.74   2.32  12.80   5.04
           4T   51762  16408  33068   9781          4.99   2.33  17.84   4.96
           8T   46575  15741  27979   9235          4.54   2.24  15.17   4.68
  12288    1T   12227   1729    538    328          3.17   1.39   2.99   1.94
           2T   16713   1724    617    311          4.41   1.39   2.80   1.82
           4T   16771   1825    722    312          4.26   1.66   2.10   1.84
           8T   13124   1739    622    319          3.23   1.39   1.77   1.87
    No Errors Found
    End of test Mon Aug 14 10:46:01 2023
```

```
Pi 5 gcc 12                                        Pi 5 GCC 12/8
 MP-RandMem 64 Bit gcc 12 Thu Sep 28 22:15:02 2023

    MB/Second Using 1, 2, 4 and 8 Threads
     KB         SerRD  SerRW  RndRD  RndRW         SerRD  SerRW  RndRD  RndRW

   12.31T       18667  19102  18108  17246           1.0    1.0    1.0    1.0
       2T       34841  19037  33292  16912           1.1    1.0    1.0    1.0
       4T       47204  18694  46771  17137           1.0    1.0    1.0    1.0
       8T       35115  18676  34015  17230           1.0    1.0    1.0    1.0
  122.91T       15826  16395  10993   9928           1.0    1.0    1.0    1.0
       2T       30566  16400  21397   9940           1.0    1.0    1.0    1.0
       4T       56413  16361  38355   9921           1.1    1.0    1.2    1.0
       8T       54596  16372  37617   9889           1.2    1.0    1.3    1.1
  122881T       13622   1902    539    343           1.1    1.1    1.0    1.0
       2T       20937   1830    603    345           1.3    1.1    1.0    1.1
       4T       26993   1892    682    343           1.6    1.0    0.9    1.1
       8T       18621   1797    650    347           1.4    1.0    1.0    1.1
    No Errors Found
    End of test Thu Sep 28 22:15:42 2023
```

MP-MFLOPS Benchmarks below or Go To Start

MP-MFLOPSPi64g8 and g12, MP-MFLOPSPi64DPg8 and g12

MP-MFLOPS measures floating point speed on data from caches and RAM. The first calculations are as used in Memory Speed Benchmark, with a multiply and an add per data word read. The second uses 32 operations per input data word of the form x[i] = (x[i]+a)*b-(x[i]+c)*d+(x[i]+e)*f -- more. Tests cover 1, 2, 4 and 8 threads, each carrying out the same calculations but accessing different segments of the data. Here are two varieties, single precision and double precision, both attempting to show near maximum MP floating point processing speeds.

At a given precision, result sumchecks should be identical when using the same run time parameters. Here, gcc 12 compiled programs were run using parameters that produce longer running times, with different sumchecks to those from earlier versions.

These are all short tests running at full MHz with low increases in temperatures. All at 12.8 and 128 KB demonstrate some near doubling performance with twice as many threads. Maximum GCC 12 Pi 5 SP 4 thread performance was 84.9 GFLOPS with DP at 42.5 GFLOPS and slightly less via GCC 8. See next page for comments on comparisons.

Pi 4 GCC 8 MP-MFLOPS 64 Bit gcc 8 Tue May 26 12:01:44 2020

	2 Ops/Word			32 Ops/Word				
KB	12.8	128	12800	12.8	128	12800	Maximum	MFLOPS
	MFLOPS						GFLOPS	per MHz
1T	3212	3162	416	6741	6720	6393	6.7	4.5
2T	6343	5109	565	13381	13376	9914	13.4	8.9
4T	11644	5077	584	25506	26028	9883	26.0	17.4
8T	7804	7953	579	20537	24446	8651		
Results x 100000, 0 indicates ERRORS								
1T	76406	97075	99969	66015	95363	99951		
2T	76406	97075	99969	66015	95363	99951		
4T	76406	97075	99969	66015	95363	99951		
8T	76406	97075	99969	66015	95363	99951		

End of test Tue May 26 12:01:46 2020

Pi 5 GCC 8 MP-MFLOPS 64 Bit gcc 8 Mon Aug 14 11:16:36 2023

	2 Ops/Word			32 Ops/Word				
KB	12.8	128	12800	12.8	128	12800	Maximum	MFLOPS
	MFLOPS						GFLOPS	per MHz
1T	9309	8856	540	20396	19543	11710	19.5	8.1
2T	17114	18565	683	35842	40506	11937	40.5	16.9
4T	29453	34610	826	75120	77896	12646	77.9	32.5
8T	28688	31506	959	59804	57700	15374		
Results x 100000, 0 indicates ERRORS								
1T	76406	97075	99969	66015	95363	99951		
2T	76406	97075	99969	66015	95363	99951		
4T	76406	97075	99969	66015	95363	99951		
8T	76406	97075	99969	66015	95363	99951		

End of test Mon Aug 14 11:16:37 2023

Pi 5/4 GCC8

	2 Ops/Word			32 Ops/Word		
KB	12.8	128	12800	12.8	128	12800
1T	2.90	2.80	1.30	3.03	2.91	1.83
2T	2.70	3.63	1.21	2.68	3.03	1.20
4T	2.53	6.82	1.41	2.95	2.99	1.28
8T	3.68	3.96	1.66	2.91	2.36	1.78

Pi 5 GCC 12 MP-MFLOPS2 64 Bit gcc 12 Tue Oct 3 09:52:45 2023

	2 Ops/Word			32 Ops/Word				
KB	12.8	128	12800	12.8	128	12800	Maximum	MFLOPS
MFLOPS							GFLOPS	per MHz
1T	10549	10320	1116	21519	21452	16879	21.5	9.0
2T	19881	20929	982	42488	43002	14280	43.0	17.9
4T	33400	40206	929	80947	84933	14772	84.9	35.4
8T	33448	37854	1093	77117	85086	17371		
Results x 100000, 0 indicates ERRORS								
1T	40015	44934	98519	35186	36769	97639		
2T	40015	44934	98519	35186	36769	97639		
4T	40015	44934	98519	35186	36769	97639		
8T	40015	44934	98519	35186	36769	97639		

End of test Tue Oct 3 09:53:21 2023

Pi 5 GCC 12/8

	2 Ops/Word			32 Ops/Word		
KB	12.8	128	12800	12.8	128	12800
1T	1.09	1.05	1.11	1.03	1.09	1.00
2T	1.12	0.98	0.98	1.15	0.94	0.89
4T	1.09	1.13	0.99	0.88	0.89	1.01
8T	0.85	0.85	1.02	0.97	1.07	0.98

Double Precision Results and More Comments below

With the running times being relatively short, individual comparison ratios might not be accurate so averages have been calculated. Pi5/Pi4 GCC 8 ratios were between 2.36 and 6.82 times, average 3.18 with cached data then 1.10 to 1.83, 1.42 from RAM. The Pi 5 improved cache sizes lead to the higher ratios. Longer running stress tests provide more reliable performance indications

GCC 8/12 averages indicated similar single precision performance, with a slight gain for the newer compiler with double precision calculations.

```
Pi 4 GCC 8 MP-MFLOPS 64 Bit gcc 8 Double Precision Tue May 26 12:11:50 2020

        2 Ops/Word              32 Ops/Word
 KB     12.8     128   12800    12.8     128   12800     Maximum  MFLOPS
 MFLOPS                                                   GFLOPS per MHz
 1T     1591    1587     269    3386    3379    3240        3.4     2.3
 2T     3228    2803     267    6728    6711    4556        6.7     4.5
 4T     5870    3284     283   12812   12866    4940       12.9     8.6
 8T     5506    4063     277   12077   11538    4695
 Results x 100000, 0 indicates ERRORS
 1T    76384   97072   99969   66065   95370   99951
 2T    76384   97072   99969   66065   95370   99951
 4T    76384   97072   99969   66065   95370   99951
 8T    76384   97072   99969   66065   95370   99951

         End of test Tue May 26 12:11:52 2020

Pi 5 GCC 8 MP-MFLOPS 64 Bit gcc 8 Double Precision Mon Aug 14 11:18:26 2023

        2 Ops/Word              32 Ops/Word
    KB  12.8     128   12800    12.8     128   12800     Maximum  MFLOPS
        MFLOPS                                            GFLOPS per MHz
 1T     4661    4127     296   10498   10217    4938       10.2     4.3
 2T     8408    9292     333   20699   19275    5579       19.3     8.0
 4T    14723   17372     399   39480   42352    6572       42.4    17.6
 8T    14387   15799     461   38706   28821    7667
 Results x 100000, 0 indicates ERRORS
 1T    76384   97072   99969   66065   95370   99951
 2T    76384   97072   99969   66065   95370   99951
 4T    76384   97072   99969   66065   95370   99951
 8T    76384   97072   99969   66065   95370   99951

         End of test Mon Aug 14 11:18:27 2023

Pi 5/4 GCC8
        2 Ops/Word              32 Ops/Word
    KB  12.8     128   12800    12.8     128   12800

 1T     2.93    2.60    1.10    3.10    3.02    1.52
 2T     2.60    3.32    1.25    3.08    2.87    1.22
 4T     2.51    5.29    1.41    3.08    3.29    1.33
 8T     2.61    3.89    1.66    3.20    2.50    1.63

 Pi 5 GCC 12 DP MP-MFLOPS2 64 Bit gcc 12 Double Precision Tue Oct  3 10:00:48 2023

        2 Ops/Word              32 Ops/Word
 KB     12.8     128   12800    12.8     128   12800     Maximum  MFLOPS
 MFLOPS                                                   GFLOPS per MHz
 1T     4713    4740     562   10748   10727    8440       10.7     4.5
 2T     9355    9554     491   21389   21515    7875       21.5     9.0
 4T    17485   18403     468   41704   42464    7499       42.5    17.7
 8T    16645   18592     543   41049   41910    8596
 Results x 100000, 0 indicates ERRORS
 1T    39991   44914   98518   35119   36721   97642
 2T    39991   44914   98518   35119   36721   97642
 4T    39991   44914   98518   35119   36721   97642
 8T    39991   44914   98518   35119   36721   97642

        End of test Tue Oct  3 10:01:24 2023

Pi 5 GCC 12/8
        2 Ops/Word              32 Ops/Word
    KB  12.8     128   12800    12.8     128   12800

 1T     1.01    1.15    1.90    1.02    1.05    1.71
 2T     1.11    1.03    1.47    1.03    1.12    1.41
 4T     1.19    1.06    1.17    1.06    1.00    1.14
 8T     1.16    1.18    1.18    1.06    1.45    1.12
```

OpenMP-MFLOPS Benchmarks below or Go To Start

OpenMP-MFLOPS - OpenMP-MFLOPS64g8 and g12, notOpenMP-MFLOPS64g8 and g12

This benchmark carries out the same single precision calculations as the MP-MFLOPS Benchmarks but, in addition, calculations with eight operations per data word. There is also notOpenMP-MFLOPS single core version, compiled from the same code and carrying out identical numbers of floating point calculations, but without an OpenMP compile directive. Again, gcc 12 compilations were run for longer times that resulted in different "First Results" sumchecks.

In this case, data sizes used were 400 KB, 4 MB and 40 MB where, with the Pi 5, only the first would be expected to provide a full service from L1 or L2 caches and the second with possible impact of L3 cache. With the GCC 8 full OpenMP version, Pi5/Pi4 performance gains were around 3.0 times at 8 and 32 Operations per word at 400 KB, with most others lower due to data size or fewer operations. At 400 KB Pi 5 GCC 12 performance was 3.2 times faster than GCC 8 at 2 operations per word and slightly faster on the other measurements.

Maximum 4 core performance was 80.1 GFLOPS from GCC 12, at 3.73 times that for a single core, nearly the same as that for MP-MFLOPS.

Pi 4 GCC 8 OpenMP MFLOPS64g8 Tue May 26 12:06:36 2020

Test	4 Byte Words	Ops/ Word	Repeat Passes	Secs	MFLOPS	First Results	All Same	MP/ notMP
Data in & out	100000	2	2500	0.093	5389	0.92954	Yes	1.64
Data in & out	1000000	2	250	0.795	629	0.99255	Yes	1.21
Data in & out	10000000	2	25	0.784	638	0.99925	Yes	1.00
Data in & out	100000	8	2500	0.115	17455	0.95712	Yes	3.11
Data in & out	1000000	8	250	0.798	2507	0.99552	Yes	1.16
Data in & out	10000000	8	25	0.880	2273	0.99955	Yes	0.95
Data in & out	100000	32	2500	0.332	24068	0.89022	Yes	3.54
Data in & out	1000000	32	250	0.849	9418	0.98809	Yes	1.45
Data in & out	10000000	32	25	0.933	8571	0.99880	Yes	1.31

End of test Tue May 26 12:06:42 2020

Pi 5 GCC 8 OpenMP MFLOPS64g8 Mon Aug 14 12:08:35 2023

Test	4 Byte Words	Ops/ Word	Repeat Passes	Secs	MFLOPS	First Results	All Same	Pi5/4 GCC8	MP/ notMP
Data in & out	100000	2	2500	0.054	9204	0.92954	Yes	1.71	1.00
Data in & out	1000000	2	250	0.439	1140	0.99255	Yes	1.81	0.80
Data in & out	10000000	2	25	0.618	809	0.99925	Yes	1.27	1.09
Data in & out	100000	8	2500	0.038	52914	0.95712	Yes	3.03	2.92
Data in & out	1000000	8	250	0.410	4880	0.99552	Yes	1.95	0.83
Data in & out	10000000	8	25	0.664	3014	0.99955	Yes	1.33	1.00
Data in & out	100000	32	2500	0.112	71522	0.89022	Yes	2.97	3.60
Data in & out	1000000	32	250	0.424	18865	0.98809	Yes	2.00	1.07
Data in & out	10000000	32	25	0.622	12853	0.99880	Yes	1.50	0.93

End of test Mon Aug 14 12:08:38 2023

Pi 5 GCC 12 OpenMP MFLOPSL64g12 Tue Oct 3 16:27:53 2023

Test	4 Byte Words	Ops/ Word	Repeat Passes	Secs	MFLOPS	First Results	All Same	Pi 5 GCC 12/8	MP/ notMP
Data in & out	100000	2	50000	0.339	29459	0.44935	Yes	3.20	3.10
Data in & out	1000000	2	5000	7.021	1424	0.86736	Yes	1.25	0.82
Data in & out	10000000	2	500	12.322	812	0.98519	Yes	1.00	0.80
Data in & out	100000	8	50000	0.634	63086	0.60398	Yes	1.19	3.46
Data in & out	1000000	8	5000	6.956	5750	0.91822	Yes	1.18	0.88
Data in & out	10000000	8	500	12.360	3236	0.99109	Yes	1.07	0.80
Data in & out	100000	32	50000	1.997	80104	0.36770	Yes	1.12	3.73
Data in & out	1000000	32	5000	6.891	23219	0.79898	Yes	1.23	1.18
Data in & out	10000000	32	500	12.294	13015	0.97639	Yes	1.01	0.79

End of test Tue Oct 3 16:28:54 2023

Single Core Results below

Some Pi5/Pi4 GCC 8 comparisons were different to those above, for the single core benchmark, at between 2.70 and 3. 22. Maximum performance was nearly 21.5 GFLOPS.

```
Pi 4 GCC 8      notOpenMP MFLOPS64g8 Tue May 26 12:07:34 2020

Test            4 Byte Ops/  Repeat  Secs MFLOPS   First  All
                 Words Word  Passes              Results Same

Data in & out    100000    2    2500 0.153  3278  0.92954 Yes
Data in & out   1000000    2     250 0.966   518  0.99255 Yes
Data in & out  10000000    2      25 0.782   640  0.99925 Yes

Data in & out    100000    8    2500 0.356  5612  0.95712 Yes
Data in & out   1000000    8     250 0.926  2160  0.99552 Yes
Data in & out  10000000    8      25 0.840  2381  0.99955 Yes

Data in & out    100000   32    2500 1.176  6800  0.89022 Yes
Data in & out   1000000   32     250 1.228  6515  0.98809 Yes
Data in & out  10000000   32      25 1.225  6529  0.99880 Yes

                End of test Tue May 26 12:07:42 2020

Pi 5 GCC 8     notOpenMP MFLOPS64g8 Mon Aug 14 12:04:30 2023

Test            4 Byte Ops/  Repeat  Secs MFLOPS   First  All      Pi5/4
                 Words Word  Passes              Results Same       GCC8

Data in & out    100000    2    2500 0.054  9236  0.92954 Yes       2.82
Data in & out   1000000    2     250 0.350  1429  0.99255 Yes       2.76
Data in & out  10000000    2      25 0.675   740  0.99925 Yes       1.16

Data in & out    100000    8    2500 0.111 18092  0.95712 Yes       3.22
Data in & out   1000000    8     250 0.340  5888  0.99552 Yes       2.73
Data in & out  10000000    8      25 0.666  3002  0.99955 Yes       1.26

Data in & out    100000   32    2500 0.402 19891  0.89022 Yes       2.93
Data in & out   1000000   32     250 0.456 17563  0.98809 Yes       2.70
Data in & out  10000000   32      25 0.579 13810  0.99880 Yes       2.12

                End of test Mon Aug 14 12:04:33 2023

Pi 5 GCC 12  notOpenMP MFLOPSL64g12 Tue Oct  3 16:31:00 2023

Test            4 Byte Ops/  Repeat  Secs MFLOPS   First  All       Pi 5
                 Words Word  Passes              Results Same   GCC 12/8

Data in & out    100000    2   50000 1.053  9493  0.44935 Yes       1.03
Data in & out   1000000    2    5000 5.732  1745  0.86736 Yes       1.22
Data in & out  10000000    2     500 9.859  1014  0.98519 Yes       1.37

Data in & out    100000    8   50000 2.194 18228  0.60398 Yes       1.01
Data in & out   1000000    8    5000 6.121  6535  0.91822 Yes       1.11
Data in & out  10000000    8     500 9.872  4052  0.99109 Yes       1.35

Data in & out    100000   32   50000 7.449 21479  0.36770 Yes       1.08
Data in & out   1000000   32    5000 8.121 19701  0.79898 Yes       1.12
Data in & out  10000000   32     500 9.698 16498  0.97639 Yes       1.19

                End of test Tue Oct  3 16:32:01 2023
```

OpenMP-MemSpeed Benchmarks below or Go To Start

OpenMP-MemSpeed264g8 and g12, NotOpenMP-MemSpeed64g8 and g12

This is the same program as the single core MemSpeed benchmark, but with increased memory sizes and compiled using OpenMP directives. The same program was also compiled without these directives (NotOpenMP-MemSpeed64). Although the source code appears to be suitable for speed up by parallelisation, many of the test functions are slower using OpenMP.

Complete output for the Pi 4 is shown below, but just the first few results for the others. The first two lines of single core results are also included to show that the OpenMP options used were clearly unsuitable for this program.

```
Pi 4 GCC 8
     Memory Reading Speed Test OpenMP 64 Bit gcc 8 by Roy Longbottom

               Start of test Tue May 26 12:14:39 2020

  Memory   x[m]=x[m]+s*y[m] Int+ x[m]=x[m]+y[m]       x[m]=y[m]
  KBytes    Dble   Sngl  Int32   Dble   Sngl  Int32   Dble   Sngl  Int32
    Used    MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S
 1 Core
       4   15001   4010   4387  15087   4406   4400  11211   9061   9061
       8   15532   3990   4394  15567   4386   4394  11629   9315   9314
 4 Cores
       4    7749   8500   8716   7451   8520   8533  39508  18586  18589
       8    8198   8669   8874   8148   8678   8691  38972  18863  18861
      16    8023   8499   8335   7895   8355   8507  38305  19003  19004
      32    9034   8517   8619   9127   8550   8522  37928  19071  18409
      64    8652   8201   8178   8565   8223   8093  25191  17494  17508
     128   11397  11616  11715  11345  11649  11029  13861  14097  14170
     256   18242  18745  18195  17417  18605  18019  12535  12637  12623
     512   17580  18467  18787  18010  18414  18321  12900  13180  13121
    1024    8043  10172  11540  12510  10220  12082   9800   9586   9857
    2048    4816   6807   6850   6922   6805   6666   3137   3372   3369
    4096    7029   6846   6881   7017   5145   6801   2776   3124   3112
    8192    2428   7085   7124   7068   7134   6904   2571   3092   3112
   16384    7133   7152   7328   7008   3445   7178   2473   3099   3104
   32768    2656   7643   7669   7802   7616   7559   2043   3112   3104
   65536    7995   6523   2572   7059   6514   6485   2431   2955   3036
  131072    1981   7273   7327   1878   3615   7267   2538   2968   2976

                End of test Tue May 26 12:15:06 2020
```

```
Pi 5 GCC 8
     Memory Reading Speed Test OpenMP 64 Bit gcc 8 by Roy Longbottom

               Start of test Mon Aug 14 11:42:10 2023

  Memory   x[m]=x[m]+s*y[m] Int+ x[m]=x[m]+y[m]       x[m]=y[m]
  KBytes    Dble   Sngl  Int32   Dble   Sngl  Int32   Dble   Sngl  Int32
    Used    MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S
 1 Core
       4   50151   6872   7511  50254   7170   7181  37548  18867  25383
       8   50904   6848   7485  48915   7202   7487  38102  19038  25477
 4 Cores
       4   31324  14321  12707  28712  14606  21136  27075  18075  18075
       8   28580  13022  13365  32094  14657  21740  26558  13931  16817
      16   27074  19393  19847  32121  19067  24532  35440  24095  23527
      32   37880  31590  31455  34779  32095  29027  37245  22243  24984
      64   23823  29763  30980  30310  28829  28209  23569  27625  24428

                End of test Mon Aug 14 11:42:37 2Pi 5 GCC 12
```

```
Pi 5 GCC 12
     Memory Reading Speed Test OpenMP 64 Bit gcc 12 by Roy Longbottom

               Start of test Thu Sep 28 22:43:26 2023

  Memory   x[m]=x[m]+s*y[m] Int+ x[m]=x[m]+y[m]       x[m]=y[m]
  KBytes    Dble   Sngl  Int32   Dble   Sngl  Int32   Dble   Sngl  Int32
    Used    MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S   MB/S
 1 Core
       4   54368  65257  65165  53930  60045  60975  37606  25361  25384
       8   54564  65580  65162  55228  61180  60995  37829  25015  25010
 4 Cores
       4   31314  14584  13443  31523  14625  21373  26964  17800  17883
       8   29471  14672  13405  32067  14677  21719  27561  18585  18540
      16   32013  19352  19797  32164  19549  25666  36645  25085  25423
      32   43228  38115  33331  42989  38653  39254  49341  30903  30892

                End of test Thu   Sep-28   22:4351 2023
```

Single Core Results below

Single Core Benchmark - Again a complete output is provided plus limited results and comparisons. As expected, the latter are similar to those from the original MemSpeed included above. Here, maximum Pi5/4 comparison was 13.9 or L3 cache versus RAM speed.

As before, GCC 12 provided corrections for the GCC 8 fault, now indicating Pi 5 GCC 12/8 performance gains of up to 8.5 times for single precision calculations.

Pi 4 GCC 8

Memory Reading Speed Test notOpenMP 64 Bit gcc 8 by Roy Longbottom

Start of test Tue May 26 12:18:16 2020

Memory	x[m]=x[m]+s*y[m] Int+			x[m]=x[m]+y[m]			x[m]=y[m]		
KBytes	Dble	Sngl	Int32	Dble	Sngl	Int32	Dble	Sngl	Int32
Used	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S
4	15001	4010	4387	15087	4406	4400	11211	9061	9061
8	15532	3990	4394	15567	4386	4394	11629	9315	9314
16	15707	3998	4376	15770	4388	4393	11801	9447	9444
32	14552	3885	4245	14761	4186	4260	11627	9488	9495
64	12272	3855	4211	12089	4196	4220	8866	8606	8630
128	12321	3867	4217	12148	4182	4215	8221	8296	8292
256	12318	3871	4219	12134	4206	4219	8092	8231	8229
512	12118	3870	4218	12195	4211	4218	8077	8209	8226
1024	3224	3738	4032	3701	4009	4066	5387	5529	5331
2048	1945	3474	3615	1949	3598	3612	2848	2860	2945
4096	1940	3442	3610	1941	3406	3607	2614	2595	2597
8192	1951	3425	3637	1954	3617	3644	2595	2581	2583
16384	1962	3330	3467	1965	3443	3469	2588	2575	2564
32768	2003	3364	3303	1997	3292	3303	2503	2554	2557
65536	2005	2588	2937	2011	2930	2621	2577	2565	2566
131072	2024	2021	2025	2013	2014	2018	2586	2572	2570

End of test Tue May 26 12:18:42 2020

Pi 5 GCC 8

Memory Reading Speed Test notOpenMP 64 Bit gcc 8 by Roy Longbottom

Start of test Mon Aug 14 11:34:27 2023

Memory	x[m]=x[m]+s*y[m] Int+			x[m]=x[m]+y[m]			x[m]=y[m]		
KBytes	Dble	Sngl	Int32	Dble	Sngl	Int32	Dble	Sngl	Int32
Used	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S
4	50151	6872	7511	50254	7170	7181	37548	18867	25383
64	50862	6800	7423	50901	7140	7426	36297	19013	25373
256	32627	6790	7153	32638	7183	7276	34872	19156	25339
1024	30004	6804	7283	30354	7171	7122	23523	18525	23493
8192	2992	6089	5571	2005	5255	6448	4794	5279	5340

End of test Mon Aug 14 11:34:52 2023

Pi 5/4 GCC8

4	3.34	1.71	1.71	3.33	1.63	1.63	3.35	2.08	2.80
64	4.14	1.76	1.76	4.21	1.70	1.76	4.09	2.21	2.94
256	2.65	1.75	1.70	2.69	1.71	1.72	4.31	2.33	3.08
1024	9.31	1.82	1.81	8.20	1.79	1.75	4.37	3.35	4.41
2048	12.94	1.91	1.98	13.90	1.98	2.04	6.95	5.99	4.05
8192	1.53	1.78	1.53	1.03	1.45	1.77	1.85	2.05	2.07

Pi 5 GCC 12

Memory Reading Speed Test notOpenMP 64 Bit gcc 12 by Roy Longbottom

Start of test Thu Sep 28 22:42:10 2023

Memory	x[m]=x[m]+s*y[m] Int+			x[m]=x[m]+y[m]			x[m]=y[m]		
KBytes	Dble	Sngl	Int32	Dble	Sngl	Int32	Dble	Sngl	Int32
Used	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S	MB/S
4	54368	65257	65165	53930	60045	60975	37606	25361	25384
64	52501	65304	65319	53250	59544	59850	37508	25373	25401
256	33354	63081	63764	33718	60298	60351	35597	25397	25398
2048	22287	52312	53008	22349	50665	49230	11449	12273	16589
8192	3087	6050	6120	3132	6038	6491	6902	6608	6778

End of test Thu Sep 28 22:42:35 2023

Pi 5 GCC 12/8

4	1.08	9.50	8.68	1.07	8.37	8.49	1.00	1.34	1.00
64	1.03	9.60	8.80	1.05	8.34	8.06	1.03	1.33	1.00
256	1.02	9.29	8.91	1.03	8.39	8.29	1.02	1.33	1.00
2048	0.89	7.88	7.42	0.82	7.10	6.68	0.58	0.72	1.39
8192	1.03	0.99	1.10	1.56	1.15	1.01	1.44	1.25	1.27

JavWhetstone Benchmark below or Go To Start

Java Whetstone Benchmark - whetstc.class

The Java benchmarks comprise class files that were produced some time ago. But source codes are available to renew the files. Performance can vary significantly using different Java Virtual Machines.

Pi 5 performance gains, over the Pi 4, were beteen 1.94 and 3.81.

```
 Pi 4 Whetstone Benchmark Java Version, May 22 2020, 14:24:09

                                                1 Pass
 Test                 Result     MFLOPS    MOPS  millisecs

N1 floating point  -1.124750137     521           0.0369
N2 floating point  -1.131330490     481           0.2792
N3 if then else     1.000000000             236   0.4378
N4 fixed point     12.000000000            1320   0.2386
N5 sin,cos etc.     0.499110132              48   1.7348
N6 floating point   0.999999821     276           1.9520
N7 assignments      3.000000000             320   0.5772
N8 exp,sqrt etc.    0.825148463              25   1.4640

MWIPS                              1488           6.7205

  Operating System    Linux, Arch. aarch64, Version 4.19.118-v8+
  Java Vendor         Debian, Version  11.0.7
  CPU null

 Pi 5 Whetstone Benchmark Java Version, Aug 24 2023, 23:25:17

                                              1 Pass    Pi 5/4
Test                  Result     MFLOPS    MOPS  millisecs

N1 floating point  -1.124750137    1232           0.0156     2.37
N2 floating point  -1.131330490    1048           0.1282     2.18
N3 if then else     1.000000000             715   0.1448     3.02
N4 fixed point     12.000000000            2559   0.1231     1.94
N5 sin,cos etc.     0.499110132             183   0.4550     3.81
N6 floating point   0.999999821     554           0.9730     2.00
N7 assignments      3.000000000             624   0.2960     1.95
N8 exp,sqrt etc.    0.935364604              63   0.5920     2.47

MWIPS                              3666           2.7277     2.46
```

JavaDraw Benchmark below or Go To Start

JavaDraw Benchmark - JavaDrawPi.class

The benchmark uses small to rather excessive simple objects to measure drawing performance in Frames Per Second (FPS). Five tests draw on a background of continuously changing colour shades, each test adding to the load.

The first runs of this benchmark on the Pi 5 indicated that it was much slower than the Pi 4 on the more demanding functions. Sometime later I reran the benchmark on the Pi 4, using the OS acquired for the Pi 5, and that also produced the slow results. Using this OS, the Pi 5 average performance was around twice as fast.

```
Pi 4 Java Drawing Benchmark, May 22 2020, 14:25:15
           Produced by javac 1.8.0_222

 Test                                Frames      FPS

 Display PNG Bitmap Twice Pass 1       833     83.26
 Display PNG Bitmap Twice Pass 2      1001    100.05
 Plus 2 SweepGradient Circles          994     99.39
 Plus 200 Random Small Circles         836     83.54
 Plus 320 Long Lines                   380     37.98
 Plus 4000 Random Small Circles         95      9.44

        Total Elapsed Time  60.1 seconds

 Operating System    Linux, Arch. aarch64, Version 4.19.118-v8+
 Java Vendor         Debian, Version  11.0.7
 null, null CPUs
```

```
Pi 4 Java Drawing Benchmark, Dec 2 2023, 10:01:16
           Produced by javac 1.8.0_222

 Test                                Frames      FPS

 Display PNG Bitmap Twice Pass 1       469     46.86
 Display PNG Bitmap Twice Pass 2       561     56.06
 Plus 2 SweepGradient Circles          523     52.21
 Plus 200 Random Small Circles          31      2.97
 Plus 320 Long Lines                    13      1.22
 Plus 4000 Random Small Circles          2      0.18

        Total Elapsed Time  62.5 seconds

 Operating System    Linux, Arch. aarch64, Version 6.1.47-v8+
 Java Vendor         Debian, Version  17.0.8
 null, null CPUs
```

```
Pi 5 Java Drawing Benchmark, Aug 26 2023, 15:06:26
           Produced by javac 1.8.0_222

 Test                                Frames      FPS   Pi5/Pi4

 Display PNG Bitmap Twice Pass 1      1000     99.96     2.13
 Display PNG Bitmap Twice Pass 2      1077    107.66     1.92
 Plus 2 SweepGradient Circles         1010    100.99     1.93
 Plus 200 Random Small Circles          63      6.16     2.07
 Plus 320 Long Lines                    26      2.50     2.05
 Plus 4000 Random Small Circles          4      0.32     1.78

        Total Elapsed Time  63.1 seconds

 Operating System    Linux, Arch. aarch64, Version 6.1.32-v8+
 Java Vendor         Debian, Version  17.0.8
 null, null CPUs
```

```
Pi 5 Java Drawing Benchmark, Aug 26 2023, 15:15:27
           Produced by javac openjdk 17.0.8

 Test                                Frames      FPS

 Display PNG Bitmap Twice Pass 1      1014    101.33
 Display PNG Bitmap Twice Pass 2      1067    106.59
 Plus 2 SweepGradient Circles         1028    102.70
 Plus 200 Random Small Circles          61      6.04
 Plus 320 Long Lines                    25      2.47
 Plus 4000 Random Small Circles          4      0.33

        Total Elapsed Time  62.3 seconds

 Operating System    Linux, Arch. aarch64, Version 6.1.32-v8+
 Java Vendor         Debian, Version  17.0.8
 null, null CPUs
```

OpenGL Benchmark below or Go To Start

64 Bit OpenGL Benchmark - videogl64C10, videogl64C12

In 2012, I approved a request from a Quality Engineer at Canonical, to use this OpenGL benchmark in the testing framework of the Unity desktop software. The program can be run as a benchmark, or selected functions, as a stress test of any duration.

The benchmark measures graphics speed in terms of Frames Per Second (FPS) via six simple and more complex tests. The first four tests portray moving up and down a tunnel including various independently moving objects, with and without texturing. The last two tests, represent a real application for designing kitchens. The first is in wireframe format, drawn with 23,000 straight lines. The second has colours and textures applied to the surfaces.

As a benchmark, it was run using the following script file format, the first command needed to avoid VSYNC, allowing FPS to be greater than 60.

```
 export vblank_mode=0
./videogl64CXX Width 320, Height 240, NoEnd
./videogl64Cxx Width 640, Height 480, NoHeading, NoEnd
./videogl64Cxx Width 1024, Height 768, NoHeading, NoEnd
./videogl64Cxx Width 1920, Height 1080, NoHeading
```

The original benchmark was compiled using freeglut3 but, more recently, this was not available for new systems. The gcc12 version was compiled without this but will not run on my Pi 4, Similarly, the gcc10 program is incompatible with the Pi 5.

Performance comparisons indicate that the Pi 5 was between 2.9 and 5.2 times faster than the Pi 4, with an average of 4.0 times over the 24 measurements. The GLUT variety was recompiled on the Pi 4, using GCC 12. The average Pi5 gain then became 2.5 times.

```
Pi 4 gcc 10
GLUT OpenGL Benchmark 64 GCC 10, Wed Sep 20 00:48:11 2023

         Running Time Approximately 5 Seconds Each Test

Window Size  Coloured Objects  Textured Objects  WireFrm  Texture
   Pixels        Few      All      Few      All  Kitchen  Kitchen
 Wide  High      FPS      FPS      FPS      FPS      FPS      FPS

  320   240    727.7    413.0    219.7    131.9     42.8     28.9
  640   480    388.6    281.9    189.2    118.0     42.5     28.1
 1024   768    144.0    141.2    129.8     96.9     41.6     26.8
 1920  1080     54.1     50.2     52.7     56.7     38.4     23.9

                  End at Wed Sep 20 00:50:26 2023

Pi 5 gcc 12
GLUT OpenGL Benchmark 64 Bit GCC 12, Thu Oct 26 14:52:15 2023

         Running Time Approximately 5 Seconds Each Test

Window Size  Coloured Objects  Textured Objects  WireFrm  Texture
   Pixels        Few      All      Few      All  Kitchen  Kitchen
 Wide  High      FPS      FPS      FPS      FPS      FPS      FPS

  320   240   3184.7   1554.8    894.7    474.2    224.0    120.0
  640   480   1441.4    956.8    819.1    442.2    220.4    116.7
 1024   768    624.6    493.7    474.7    364.0    199.1    106.4
 1920  1080    221.4    198.6    194.4    165.8    137.9     87.6

                  End at Thu Oct 26 14:54:28 2023

Pi 5/4 Comparison
Window Size  Coloured Objects  Textured Objects  WireFrm  Texture
   Pixels        Few      All      Few      All  Kitchen  Kitchen
 Wide  High      FPS      FPS      FPS      FPS      FPS      FPS

  320   240      4.4      3.8      4.1      3.6      5.2      4.2
  640   480      3.7      3.4      4.3      3.7      5.2      4.2
 1024   768      4.3      3.5      3.7      3.8      4.8      4.0
 1920  1080      4.1      4.0      3.7      2.9      3.6      3.7

 #####################################################################
Pi 4
 GLUT OpenGL Benchmark 64 Bit GCC 12, Sat Dec  2 11:35:48 2023

         Running Time Approximately 5 Seconds Each Test

Window Size  Coloured Objects  Textured Objects  WireFrm  Texture
   Pixels        Few      All      Few      All  Kitchen  Kitchen
 Wide  High      FPS      FPS      FPS      FPS      FPS      FPS

  320   240   1137.1    517.1    308.3    159.7     93.5     49.6
  640   480    579.0    356.8    283.9    150.5     92.5     48.7
 1024   768    239.5    200.9    203.4    134.7     84.9     45.3
 2032  1080     92.8     74.3     93.6     81.1     75.2     37.6

                  End at Sat Dec  2 11:38:02 2023
```

I/O Benchmarks below or Go To Start

DriveSpeed and LanSpeed I/O Benchmarks

Two varieties of I/O benchmarks are provided, one to measure performance of main and USB drives, and the other for LAN and WiFi network connections. The programs write and read three files at two sizes (defaults 8 and 16 MB), followed by random reading and writing of 1KB blocks out of 4. 8 and 16 MB and finally, writing and reading 200 small files, sized 4, 8 and 16 KB. Run time parameters are provided for the size of large files and file path. The same program code is used for both varieties, the only difference being file opening properties. The drive benchmark includes extra options to use direct I/O, avoiding data caching in main memory, but includes an extra test with caching allowed. .

As found during previous tests on 64 bit systems and accessing the system SD card, DriveSpeed with Direct I/O failed, indicating "Error writing file". Later it was established that this also applied to external drives with Ext type format but operated correctly formatted as FAT32. A limitation of the latter (at 64 bits) is that file sizes must be less than 4096 MB.

The best option for measuring 64 bit performance, using these benchmarks, is to run LanSpeed, specifying large files that cannot be cached for reading. However, random and small file reading functions are likely to be accessing cached data.

DriveSpeed Benchmark FAT32 - DriveSpeed64v2g8 and g12

The first of the following results are for Pi 4 and Pi 5, both with 8 GB RAM, exercising the same high speed flash drive via USB 3, using 1GB and 2 GB files.

Average Pi 5 gains were around 1.5 times for writing and reading large files, somewhat less writing to cache and nearly 4 times reading from cache, representing RAM speed. The Pi 5 results indicated a slower speed on random reading then much faster on reading small files, where more of the data appears to have been cached.

As during the Pi 4 tests, a starting large file parameter of 2048 KB failed to execute the second part at 4096 KB. Below indicates a successful run at 4094 KB.

```
Pi 4   DriveSpeed RasPi 64 Bit gcc 8 Wed May 27 11:43:43 2020

 Selected File Path: /media/pi/PATRIOT1/
 Total MB  120832, Free MB  114614, Used MB    6218

                         MBytes/Second
  MB   Write1   Write2   Write3    Read1    Read2    Read3

1024    27.78    21.39    21.43   270.32   278.81   274.98
2048    21.40    21.14    21.44   275.79   273.14   319.95
 Cached
   8    40.27    42.81    42.81  1206.64  1068.72  1031.56

 Random         Read                       Write
 From MB         4        8       16         4        8       16
 msecs      0.004    0.004    0.184      4.33     4.00     4.04

 200 Files      Write                      Read                    Delete
 File KB         4        8       16         4        8       16     secs
 MB/sec       0.03     0.07     0.14    261.45    11.19    84.39
 ms/file    119.60   119.05   119.64      0.02     0.73     0.19    2.477


Pi 5   DriveSpeed RasPi 64 Bit gcc 8 Mon Sep  4 16:50:50 2023

 Selected File Path:
 /media/roy/PATRIOT/test/
 Total MB  120832, Free MB  113866, Used MB    6966

                         MBytes/Second
  MB   Write1   Write2   Write3    Read1    Read2    Read3

1024    30.89    31.14    38.40   349.35   376.91   375.03
2048    42.62    42.11    34.53   377.20   378.08   375.97
 Cached
   8    50.11    52.44    53.78  2327.93  4688.75  6184.63

 Random         Read                       Write
 From MB         4        8       16         4        8       16
 msecs      0.005    0.005    0.233     13.34    12.74    13.10

 200 Files      Write                      Read                    Delete
 File KB         4        8       16         4        8       16     secs
 MB/sec       0.03     0.07     0.13    386.06   667.63   950.87
 ms/file    123.74   124.04   123.19      0.01     0.01     0.02    3.234


 Pi 5 at 4094 KB
                         MBytes/Second
  MB   Write1   Write2   Write3    Read1    Read2    Read3

4094    42.74    38.90    45.55   372.93   349.44   376.49
```

Performance Monitor for above next or Go To Start

Performance Monitor - The following provides vmstat examples handling large files, confirming the benchmark large file data transfer speeds and that the data was actually written to and read from the drive at the benchmark reported time.

```
 Pi 5 VMSTAT Writing and Reading Large Files - volumes in kB, speeds in kB/second
      %CPU utilisation us + sy, 100% means 4 cores being used

 procs  -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
  r  b   swpd    free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
  1  1      0 7260884  22404 399188    0    0  1121  1288  179  284  1  1 93  5  0
  1  1      0 7260884  22404 399188    0    0     0 40005 3082 6308  0  4 74 23  0
  1  1      0 7260884  22404 399188    0    0     0 41030 3651 6074  0  3 74 23  0
  1  1      0 7260884  22404 399188    0    0     0 43080 3839 6375  0  3 75 22  0
  1  1      0 7260884  22404 399188    0    0     0 41033 3807 6275  0  3 74 22  0

  1  1      0 7260884  22404 399188    0    0 355824     0 3879 9207  1  9 73 17  0
  1  1      0 7260884  22404 399188    0    0 355320     0 2824 7807  1  9 73 17  0
  1  1      0 7260884  22404 399188    0    0 364544     0 2728 5560  1  9 72 17  0
  1  1      0 7260884  22404 399188    0    0 364540     0 4022 5513  0  8 73 18  0
```

LanSpeed Benchmark below or Go To Start

Pi 5 LanSpeed Benchmark - LanSpeedt64g8 and g12- Wired LAN and WiFi

As indicated above, this benchmark is effectively the same as DriveSpeed, but with Direct I/O not specified. Following are data transfer speeds to a PC via gigabit LAN, 2.4 GHz WiFi and 5 GHz WiFi, plus measurement from a Pi 400 to confirm the same performance levels.

The parameter for large file sizes was intended to be large enough to avoid local caching and some were included to use data sizes of 4 GB or 16 GB in one case. Random access tests access small files that are clearly cached for reading. The many small files used could involve some caching but indicate some consistency.

```
                                 MBytes/Second To PC
                MB    Write1   Write2   Write3    Read1    Read2    Read3

Wifi 2.4GHz   1024      5.27     5.56     5.69     6.16     5.92     5.72
WiFi 5GHz     1024     11.47    11.85    12.83    11.86    11.12    11.31
LAN 1Gbps 1  16384     55.25    51.88    54.17   114.38   116.13   114.81
LAN 1Gbps 2   4096     53.83    49.33    54.38   113.70   109.48   113.51
LAN Pi 400    4096     62.19    62.11    61.27   102.43   104.56   102.60

                          Milliseconds To PC
Random          Read                       Write
From MB           4        8       16        4        8       16

Wifi 2.4GHz  0.002    0.002    0.002     8.48     8.15     7.79
WiFi 5GHz    0.002    0.002    0.002    14.52    21.38    21.96
LAN 1Gbps 1  0.002    0.002    0.002     5.04     1.45     0.98
LAN 1Gbps 2  0.002    0.002    0.002     1.71     1.37     1.38
LAN Pi 400   0.005    0.005    0.005     1.43     1.13     1.18

                          MBytes/Second To PC
 200 Files         Write                      Read
 File KB             4        8       16        4        8       16

Wifi 2.4GHz       0.33     0.62     0.92     0.52     0.66     1.21
WiFi 5GHz         0.11     0.16     0.34     0.14     0.83     0.52
LAN 1Gbps         1.43     2.39     3.13     4.06     8.28    15.30
LAN 1Gbps 2       1.59     1.53     4.80     4.41     7.78    16.67
LAN Pi 400        0.68     2.46     3.55     3.91     6.17    12.45
```

Performance Monitor for above next or Go To Start

Raspbeerry Pi Performance Monitor - First example below is for VMSTAT that does not include network data transfer speeds. This is for LAN 2 test writing and reading the first part, comprising three 2048 MB files. This ends up using most of the 8 GB RAM as a cache, where data appears read from the network. CPU utilisation was mainly low but he maximum of 14% is for 4 cores or 56% of one core (if you want to calculate CPU time).

PC Performance Monitor - In some cases network data transfer speeds could be confirmed on the Windows PC, using Task Manager Performance display and Perfmon detailed tables. However, this became confusing due to deferred writing to the PC disk, with overlapped reading. Also, Perfmon data collector could not keep up with the volume of data, missing output in time slots and indicating unobtainable speeds in a following slot. Also, transferring the largest files could produce a complete overload of the PC, with a dead keyboard. An example of Perfmon results is provided below.

The PC was a four core 3 GHz CPU running under Windows 7. The statistics show significant time waiting for I/O and utilisation of up to all four cores. The second example shows network traffic, disk drive data transfers and CPU utilisation, where a 25% recording represents 100% of one core.

The important considerations for the Pi 5 are confirmation of data transfer speeds measured by the benchmark. Then, the indication that, on reading, no disk involvement was indicated but was supplied from PC RAM based cache and on writing, saving to disk was involved that might have reduced measured speed. In the bigger picture it seemed that all data had not been written to disk when reading began.

```
LAN 1Gbps 2 VMSTAT initial part writing and reading three 2048 MB files.

procs -----------memory--------- ---swap-- ----io----  -system-- ------cpu-----
r  b  swpd    free   buff  cache   si   so   bi   bo    in    cs  us sy id wa st
Power On
1  0     0 7096944  29968 646800    0    0 4147 1026   859  1470  8  6 74 13  0
Write
1  0     0 1613712  32944 6076752   0    0  203   51  1406  1245  1  2 89  8  0
2  1     0 1352208  32944 6339728   0    0    0    0  3962  3469  0  2 75 23  0
3  0     0   58304   4192 7665904   0    0  175   44  1311  1122  1  2 90  7  0
Read
1  1     0 2727744   944  5000080   0    0  152   38  2153  1921  1  3 87  9  0
3  0     0 1480192   960  6244480   0    0    0    0 38445 42406  0 10 65 25  0
1  2     0  347872   960  7377648   0    0 1472   28 39595 42997  1 13 60 26  0
Write
2  1     0   52176  2688  7674272   0    0  148   37  2458  2198  1  3 87  9  0
1  1     0   94448  2688  7635744   0    0  148   37  2519  2253  1  3 87  9  0

 ##############################################################################

 PC Perfmon

        Comms               Disk
        Mbytes/second       Mbytes/second    %CPU
 Second Received    Sent    Read  Written

     11       50       0       0       90      49
     12       49       0       0        0      47
     13       50       0       0       88      55
     14       49       0       0        0      46
     15       49       0       0       89      45
    To
     45       37       0       0        0      36
     46     1461       4       0       99      34
     82        3       0       0       40      49
     83       79       0       0       41      56
     86      178       0       0       58      90
     94        0       5       0       43      85

     95        1     122       2       64      42
     96        1     120       1        1      36
     97        1     122       0       56      32
     98        1     121       0        0      35
     99        1     120       0       49      31
```

USB and SD Card Benchmarks below or Go To Start

LanSpeed Benchmark - Pi 5 USB Drives and Operating System SD Card

In most cases, as Direct I/O was not supported, LanSpeed was executed using large files that avoid caching.

These tests were run to confirm that the hardware could support 64 bit type file sizes and to show any major differences. It was found that 4096 MB could not be supported using FAT32 format, but such as 4096 MB was fine. Also, at 2048 MB, the 8 GB RAM might cache all the data.

MBytes/Second

	MB	Write1	Write2	Write3	Read1	Read2	Read3
USB3 HD FAT1	2048	98.07	80.66	74.72	306.43	9209.88	8687.44
USB3 HD Ext2	4096	158.98	28.25	113.34	38.47	143.80	114.56
USB3 HD Ext3	4096	122.73	26.33	61.23	48.78	122.24	109.04
USB3 HD Ext4	4096	164.59	81.99	19.61	103.72	143.48	120.17
Pi 5 SD	4096	27.95	20.58	19.20	43.45	104.53	92.26
SD USB boot	2048	52.82	20.68	20.41	10305.38	11463.08	11496.93
	4096	30.06	20.52	20.60	42.12	260.46	97.04

Milliseconds

Random	Read			Write		
From MB	4	8	16	4	8	16
USB3 HD FAT1	N/A as failed to write 4096 MB					
USB3 HD Ext2	0.002	0.002	0.002	44.90	15.38	16.10
USB3 HD Ext3	0.002	0.002	0.002	54.50	40.68	45.18
USB3 HD Ext4	0.002	0.002	0.002	52.50	45.27	51.93
Pi 5 SD	0.002	0.002	0.002	3.96	3.60	3.68
SD USB boot	0.002	0.002	0.002	6.83	4.24	3.90

MBytes/Second

200 Files	Write			Read		
File KB	4	8	16	4	8	16
USB3 HD FAT1	N/A					
USB3 HD Ext2	141.38	37.47	63.37	587.85	592.36	834.73
USB3 HD Ext3	64.24	21.61	35.24	310.16	601.22	927.89
USB3 HD Ext4	129.74	55.08	104.42	423.15	473.34	465.93
Pi 5 SD	78.41	95.12	194.19	554.82	732.07	1189.95
SD USB boot	106.88	121.88	309.35	596.63	789.24	1504.37

New Benchmark More Files next or Go To Start

New Benchmark More Files - LANSpeed64Long

Having encountered VMSTAT performance monitoring problems on running my LANSpeed program, I found that my original Linux version, LANSpeed64Long, avoided this, when compiled for the Raspberry Pi. This writes and reads five large files, followed by other tests, including some for random access and handling numerous small files. As with the earlier program, measured performance can be influenced by caching, sometimes in an unexpected way. Using extra large files helps to avoid the latter. Following is an example of results and sample details from VMSTAT system monitor.

```
 Current Directory Path:
 /home/???????
 Total MB  119699, Free MB  102167, Used MB   17531

 Linux LAN Speed Test 64-Bit Version 1.2, Wed Sep 20 13:38:14 2023


  4096 MB File         1          2          3          4          5
 Writing MB/sec      35.46      35.54      35.53      35.49      35.61
 Reading MB/sec     198.94     153.10      92.52      92.67      92.66

 Running Time Too Long At 793 Seconds - No More File Sizes
 ---------------------------------------------------------------------
 8 MB Cached File      1          2          3          4          5
 Writing MB/sec     895.98     859.22     817.44     770.10    1032.07
 Reading MB/sec    3337.54    6467.72    6574.06    6768.83    6643.57

 ---------------------------------------------------------------------
 Bus Speed Block KB     64        128        256        512       1024
 Reading MB/sec   13574.63   15329.45   16213.07   14365.65    9021.80

 ---------------------------------------------------------------------
 1 KB Blocks File MB >    2      4      8     16     32     64    128
 Random Read  msecs    0.40   0.44   0.45   0.45   0.45   0.45   0.45
 Random Write msecs    4.50   4.63   4.60   4.64   4.58   4.68   4.58

 ---------------------------------------------------------------------
 500 Files   Write             Read            Delete
 File KB     MB/sec  ms/File   MB/sec  ms/File  Seconds
       2       0.42     4.85   357.91     0.01    0.012
       4       0.82     5.01   636.20     0.01    0.012
       8       1.64     5.00  1224.07     0.01    0.013
      16       2.91     5.62  1288.33     0.01    0.033
      32       5.51     5.94  2573.57     0.01    0.014
      64       9.22     7.11  4727.86     0.01    0.015
     128      15.04     8.72  5015.65     0.03    0.019
     256      22.87    11.46  5514.21     0.05    0.024
     512      30.27    17.32  6487.64     0.08    0.061
    1024      34.50    30.39  5629.98     0.19    0.054
    2048      36.80    56.99 11498.58     0.18    0.087

VMSTAT Samples Large Files

procs  -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
Before Start
 1  0      0 6245248  54480 1069568  0    0     0     0  199  275  0  0 100  0  0
Write
 1  1      0   41088  76480 7254656    0    0     0 34584  714 1313  0  2 75 23  0
 1  1      0   41088  76480 7254656    0    0    16 35656 2310 4149  0  2 73 24  0
 1  1      0   41088  76480 7254656    0    0     0 36656 1830 3219  1  3 72 23  0
 1  1      0   41088  76480 7254656    0    0    16 34584 2012 3287  6  4 68 22  0
Read
 1  1      0   59568  76624 7238688    0    0 90112     0  812 1778  1  1 75 24  0
 1  1      0   59568  76624 7238688    0    0 90112     0  738 1661  1  2 74 24  0
 1  1      0   59568  76624 7238688    0    0 90624     0  667 1524  0  1 75 24  0
 1  1      0   59568  76624 7238688    0    0 90112     0  559 1479  0  1 75 24  0
```

New Benchmark Large Files next or Go To Start

New Benchmark Large Files

These mainly involved 4096 MB files with smaller ones limited by FAT formatting, available free space or slower WiFi. Approximate vmstat reported performance is also shown. This helps to highlight benchmark results affected by caching.

The first benchmark results were for boot drives, including SD cards, flash drives and hard disk drives, with some from a USB card reader and a USB hub. The other results are for LAN, WiFi and an attached USB flash drive, booted from the SD card. The main use is to demonstrate variations in performance.

```
 Boot Drive              File         1        2        3        4        5     VMSTAT
                                                                                MB/sec
 32 GB SD        Writing MB/sec     17.31    17.59    17.69    17.64    17.52       17
 3072 MB File    Reading MB/sec    106.05  8253.16   103.94    90.49    90.38       90

 128 GB SD       Writing MB/sec     35.46    35.54    35.53    35.49    35.61       36
                 Reading MB/sec    198.94    153.1    92.52    92.67    92.66       90

 128 GB SD USB   Writing MB/sec     39.04    38.86    39.14    38.98    38.98       39
                 Reading MB/sec    132.76    297.8    97.62    97.54    97.12

 32 GB Flash     Writing MB/sec     45.32    51.26    45.14    39.56    40.95       37
 SanDisk         Reading MB/sec     347.2   764.03   263.08   259.51   256.98      250

 128 GB Flash    Writing MB/sec     65.18    59.06    55.93    51.48    44.54   20to70
 PATRIOT         Reading MB/sec    529.24   880.72   283.78   358.71   357.57      350

 Disk USB        Writing MB/sec     19.00    20.76    21.03    19.03    16.37       20
                 Reading MB/sec    187.19   390.54   115.75   103.51    91.63      125

 Disk USB HUB    Writing MB/sec     19.36    20.97    19.67    14.24    18.25       20
                 Reading MB/sec    206.35   221.78    86.34   111.81   104.16      120

 SD Booted

 GB LAN          Writing MB/sec     36.31    36.92    36.69    36.94    37.18      N/A
                 Reading MB/sec    113.61    112.8   113.33   113.87   114.18

 5 GHz WiFi      256 MB File            1        2        3        4        5
                 Writing MB/sec     24.82    19.87    17.58    24.74     19.8      N/A
                 Reading MB/sec     12.13    11.47    11.53    11.67     9.18

 USB Drive FAT32 Writing MB/sec     30.21    30.01    30.06    30.18    30.16       29
 3072 MB File    Reading MB/sec    304.19   9936.6   343.77   311.99   309.92      290

 USB Drive Ext3  Writing MB/sec Cannot open data file for writing
 Use sudo        Writing MB/sec     30.56    30.35    30.39    30.37    30.23       30
                 Reading MB/sec    385.17   877.37   311.63   303.94   303.83
```

New Benchmark Small Files Next or Go To Start

New Benchmark Small Files, Booting Time, Volts and Amps

Performance measure are for writing and reading small files and random access, again demonstrating wide variations. The latter is also identified in measured booting time (from inserting the power plug to the full display, including warnings). One of the flash drives was particularly slow at 97 seconds. This drive had also produced unusually slow results during earlier tests.

I have two meters that measure USB voltage and current. One was connected to measure power in and the other USB 3 power out. The main power supply voltage did not appear to vary much, during these tests, and current was well within the 3 available Amps. The disk drive produced the most impact, falling to below 5 volts when connected by a USB hub. Even then, the benchmark ran successfully to the end.

500 Files Write MB/sec

	32 GB SD	128 GB SD		32 GB	128 GB	Disk	Disk	Gbps	5 GHz	FAT32	Ext3
File KB	Board	Board	USB	USB Dr	USB Dr	USB	USB HUB	LAN	WiFi	USB	USB
2	0.38	0.42	0.45	0.42	0.02	0.05	0.05	0.65	0.11	0.02	0.36
4	0.74	0.82	0.90	0.68	0.19	0.15	0.09	1.11	0.38	0.04	0.63
8	1.61	1.64	1.75	2.04	0.15	0.30	0.19	1.93	0.93	0.08	1.42
16	2.74	2.91	3.11	2.67	0.95	0.46	0.40	4.24	1.77	0.15	2.89
32	3.22	5.51	5.92	4.58	1.12	0.83	0.81	7.06	3.27	0.30	5.51
64	8.06	9.22	9.88	8.92	4.66	1.64	1.58	12.41	5.71	0.60	8.45
128	9.48	15.04	16.17	10.08	4.24	3.21	3.11	17.79	8.14	1.18	13.01
256	12.46	22.87	24.02	14.43	12.69	6.35	6.03	23.18	11.43	2.29	18.55
512	15.43	30.27	31.96	20.40	21.03	11.42	11.33	27.59	13.07	4.28	23.51
1024	16.31	34.50	38.04	32.05	36.48	17.08	16.03	33.55		7.60	27.54
2048	18.15	36.80	41.70	47.85	46.68	28.00	27.30	35.39		12.35	30.07
Random Access millisecs V = Variable											
Read	0.47	0.45	0.61	0.45	0.44V	1.10V	1.52	0.67V	18.77	0.40	0.38
Write	3.20	4.60	4.65V	1.89	16.55V	43.33V	48.80	2.08V	16.23	2.77	4.80
Boot Secs	21	21	30	21	97	46	44	N/A	N/A	N/A	N/A
Power Volts and Amps											
Main V	5.20	5.28	5.21	5.24	5.20	5.18	5.21	5.16	5.18	5.18	5.17
Main A	0.87	0.92	1.13	1.09	0.98	1.21	1.52	1.10	0.85	0.91	0.93
USB V	N/A	N/A	5.11	5.12	5.10	5.04	4.97	N/A	N/A	5.11	5.11
USB A	N/A	N/A	0.28	0.24	0.14	0.44	0.83	N/A	N/A	0.14	0.14

Drive Stress Test Next or Go To Start

Drive Stress Test - burnindrive264g12

The program uses 64 KB block sizes, with 164 variations of data patterns or a minimum file size of 10.25 MB. Larger files can be produced via a run time multiplication parameter, in this case 16 for for 164 MB files. Four of these written then read sequentially for 12 minutes, but with the choice of files randomised. Finally, each block/data pattern is reread continuously for a second, at full bus speed from disk drives that cache the data. On reading, file number and data values are compared and errors reported.

Note that measured speeds are generally slower than from DriveSpeed benchmark, covered earlier, as data transfers are based on using smaller 64 KB blocks.

The following provides summary Pi 5 results including MB/second performance calculations. The tests exercised the main SD drive, LAN, WiFi and USB 3. Devices on the latter were for a hard drive with Ext2, Ext3, Ext4 and FAT32 partitions and three flash drives. The LAN and WiFi tests were also run on a Pi 400 to confirm the similar performance. No errors were detected.

A gigabit LAN connection was used and WiFi reported as 5 GHz, with the former around 5 times faster on writing and up to 10 times reading. There were performance variations on the various solid state drives that could affect certain applications. One of the disk drive tests, using the Ext3 partition, had inexplicable slow speeds and, when repeated, somewhat slower than the other partitions on writing. Note the much faster transfer speeds with repeated reading of 64 KB blocks, indicating cached data and bus speed.

Source	Write Seconds	MB/sec	Read Passes	Minutes	MB/sec	Blocks Number	Repeated Minutes	MB/sec
Comms								
LAN Pi 5 to PC	19.3	34.0	156	12.06	35.4	99360	2.79	37.1
LAN Pi 400 to PC	20.2	32.6	132	12.37	29.2	80900	2.79	30.2
WiFi Pi 5 to PC	99.6	6.6	20	14.41	3.8	12540	3.61	3.6
WiFi Pi 400 to PC	101.7	6.5	20	12.78	4.3	14720	3.66	4.2
SD OS Card	41.7	15.7	260	12.03	59.1	174960	2.76	66.0
USB 3 Flash Drive								
Flash 1	20.7	31.7	328	12.01	74.6	179200	2.76	67.6
Flash 2	8.0	82.0	352	12.06	79.8	219400	2.75	83.1
Flash 3	145.2	4.5	136	12.12	30.7	89860	2.77	33.8
USB HD								
FAT32 Partition	8.4	78.1	268	12.15	60.3	408280	2.75	154.7
Ext 2 Partition	8.9	73.7	272	12.03	61.8	432060	2.74	164.3
Ext 3 Partition	1320	0.5	100	12.14	22.5	427360	2.74	162.5
Ext 3 Repeat	11.8	55.6	256	12.09	57.9	431820	2.74	164.2
Ext 4 Partition	9.0	72.9	284	12.10	64.2	432200	2.74	164.3

BurnInDrive Stress Test With Performance Monitoring or Go To Start

BurnInDrive Stress Test With Performance Monitoring

Following are details of a run handling four 2624 MB files, along with associated results from vmstat performance monitor and my CPU Voltage, MHz and Temperature recorder. The tests were run using the Ext3 partition.

First below are the program results with faster writing speeds than above, reading speeds a little slower and repeat reading similar. These might be due to handling larger files.

Second are the sample vmstat results (size numbers are KB) with nothing strange on 8 GB memory utilisation. There were variations in bo writing and bi reading speeds but essentially confirm program measurements. Percentage user + system CPU utilisation was low (note that such a 25% reflects 100% of one core and 100% indicates four core fully utilised).

Finally are samples of the environment measurements that were effectively constant. Results are provided for the start, middle and end of the tests. With ondemand CPU frequency scaling being used, a constant 1500 MHz was indicated for most of the time.

This test was run later on a Pi 4 where writing was 9% slower, reading 6%, repeat reading 18% with similar for CPU utilisation. See results below.

```
                    Write          Read                   Blocks Repeated
Source              Seconds  MB/sec  Passes Minutes  MB/sec  Number Minutes  MB/sec

Ext 3 Partition       129.2    81.2      16   13.99    50.0  419020    2.74   159.3
Pi 4 Ext3             142.2    73.8      16   14.81    47.2  345680    2.75   130.9

VMSTAT
procs -----------memory---------- ---swap-- -----io---- -system--    ------cpu-----
 r  b   swpd    free   buff  cache   si   so    bi    bo   in   cs    us sy id wa st
WRITE
 1  1      0 6901476 137524 682832    0    0     0 77806 8123 11887   1  6 74 20  0
 2  0      0 6901476 137524 682832    0    0     8 90292 9889 13562   1  7 74 18  0
READ
 1  1      0 6901476 137524 682832    0    0 32538    46 3377 5344    0  1 75 24  0
 1  1      0 6901476 137524 682832    0    0 60064    16 7630 10652   3  2 72 24  0
REPEAT
 1  1      0 6868408 149372 699428    0    0 162170    3 19231 25503  0  4 72 24  0
 1  1      0 6868408 149372 699428    0    0 162144    3 17290 25480  0  4 72 23  0

ENVIRONMENT
 Seconds
    0.0   ARM MHz=1500, core volt=0.9067V, CPU temp=37.3°C, pmic temp=38.4°C
  453.6   ARM MHz=1500, core volt=0.9067V, CPU temp=38.9°C, pmic temp=38.4°C
  897.4   ARM MHz=1500, core volt=0.9067V, CPU temp=38.9°C, pmic temp=38.6°C
```

Disk Drive Errors and Crashes next or Go To Start

Disk Drive Errors and Crashes - Power Supply Problems

I have two 1TB USB 3 disk drives. The first crash occurred in attempting to run the new benchmark on both disk drives when connected to the USB hub via one USB port. It would have been obvious, if I had looked up the specification. That indicated a maximum of 900 mA, where up to 660 mA on one drive had been observed. It seems that a 5 Amps power supply would not help in running this sort of activity, but should be using a powered USB hub.

The second crash was running two disk drive benchmarks with one on the hub, plus my 4 thread integer CPU stress test. This time the crash appeared to be due to the power demand being greater than the 3 Amps supply. 3.06 Amps was indicated shortly before the crash.

Before the next crash I successfully ran two copies of my burnindrive264g12 stress test on separate USB ports. Then, with one of these and one integer stress test, the last measurements before the screen went blank were a data transfer failure reported by my program and a power input recording of 2.72 Amps. Following is a report from the last failing test session, indicating the seriousness of the situation, reading the wrong file and corrupted data.

Later tests were run using a 4 amps power supply. At the time of testing, the official 5 amps power supply was not available.

```
Selected File Path:
/media/raspberrypi/EXT3/
Total MB  348052, Free MB  348052, Used MB       0

Storage Stress Test ARM 64 Bit v2.0 gcc 8, Fri Oct  6 21:28:44 2023

File size 2624.00 MB x 4 files, minimum reading time 12.0 minutes

File 1 2624.00 MB written in   30.97 seconds
File 2 2624.00 MB written in   28.80 seconds
File 3 2624.00 MB written in   29.70 seconds
File 4 2624.00 MB written in   32.35 seconds


        Total  121.83 seconds, Elapsed  121.83 seconds

             Start Reading Fri Oct  6 21:30:46 2023

Error reading file 1

Wrong File Read szzztestz-820 instead of szzztestz1

Error reading file 2

Wrong File Read szzztestz-820 instead of szzztestz2

Error reading file 3

Pass   1 file szzztestz1 word  1, data error was FFFFFCCC expected FFFFCCCC
Pass   1 file szzztestz1 word  2, data error was FFFFFCCC expected FFFFCCCC

ERRORS found during reading tests, see above

             End of test Fri Oct  6 21:34:09 2023
```

Other System Crashes

The first tests carried out were run with the Pi 5 operating via a 2 amps power supply, without any real problems running the short duration benchmarks. However, there were reductions in performance on running a series of tests, due to temperature increases. I had a cheap cooling fan module used for Pi 4 tests that I fitted on top of the Pi 5, to connect for use when needed, such as for the following procedures.

High Performance Linpack - I attempted to build this benchmark, to continue using as a stress test. This takes an excessive amount of time to build, appearing to repetitively execute the code for tuning purposes for a particular computer. In view of the timescale, I ensured that the cooling fan was working.

The first attempt was left to run overnight, only to find, in the morning, that the system had crashed. A second attempt crashed after 7 hours. Later with a 3 amps power supply, it took 12 hours to build (but other required software was found to be incompatible).

Stress Test Crash - I had successfully run numerous of my floating point and integer stress tests using a data size parameter aiming to achieve maximum performance using L1 caches on all four CPU cores. Other runs with L2 cache sized data size occasionally crashed. Later these tests ran successfully using the 3 amps power supply, with similar temperature and CPU throttling levels.

Even later, with more demanding system stress tests, the 3 amps supply was found to be inadequate.

CPU Stress Testing Benchmarks next or Go To Start

CPU Stress Testing Benchmarks - MP-FPUStress64g8 and g12, MP-FPUStress64DPg8 and g12 MP-IntStress64g8 and g12

These are provided to help in determining parameters to use for a stress test. They run a series of floating point tests using 1, 2, 4 and 8 threads, with three different memory demands, with single precision and double precision versions. An integer program is also provided using 16 and 32 threads, accessing three similar memory sizes.

```
Pi 5 GCC 12 SP
 MP-Threaded-MFLOPS 64 Bit V2 gcc 12 Fri Sep 29 09:59:04 2023

            Benchmark 1, 2, 4 and 8 Threads

                       MFLOPS          Numeric Results
            Ops/   KB     KB    MB      KB     KB     MB
  Secs  Thrd Word 12.8   128  12.8    12.8    128   12.8

   0.4    T1    2 13111 12985  2003   40394  76395  99700
   0.8    T2    2 24716 26088  1849   40394  76395  99700
   1.2    T4    2 41053 45232  1847   40394  76395  99700
   1.5    T8    2 34398 44918  2141   40394  76395  99700
   2.2    T1    8 17572 17484  8265   54764  85092  99820
   2.8    T2    8 33483 35138  5731   54764  85092  99820
   3.2    T4    8 59976 69804  6737   54764  85092  99820
   3.6    T8    8 58659 69463  8481   54764  85092  99820
   5.3    T1   32 18265 18246 17917   35206  66015  99520
   6.3    T2   32 35625 36482 22484   35206  66015  99520
   7.0    T4   32 69359 72766 29572   35206  66015  99520
   7.6    T8   32 69370 66234 33184   35206  66015  99520

           End of test Fri Sep 29 09:59:12 2023

Pi 5 GCC 8 SP
  MP-Threaded-MFLOPS 64 Bit V2 gcc 8 Thu Aug 17 21:21:35 2023

            Benchmark 1, 2, 4 and 8 Threads

                       MFLOPS          Numeric Results
            Ops/   KB     KB    MB      KB     KB     MB
  Secs  Thrd Word 12.8   128  12.8    12.8    128   12.8

   0.4    T1    2 12746 12885  2029   40394  76395  99700
   0.8    T2    2 25127 24925  1791   40394  76395  99700
   1.2    T4    2 43633 45111  1797   40394  76395  99700
   1.6    T8    2 39439 44308  2151   40394  76395  99700
   2.2    T1    8 17069 17333  7672   54764  85092  99820
   2.7    T2    8 34070 34766  7170   54764  85092  99820
   3.2    T4    8 58695 69177  7229   54764  85092  99820
   3.6    T8    8 59622 65856  8346   54764  85092  99820
   5.3    T1   32 18202 18288 18037   35206  66015  99520
   6.2    T2   32 36321 36549 27452   35206  66015  99520
   6.9    T4   32 68760 73025 27221   35206  66015  99520
   7.5    T8   32 68598 72071 32869   35206  66015  99520

           End of test Thu Aug 17 21:21:42 2023

Pi 5 GCC 12 DP
  MP-Threaded-MFLOPS 64 Bit gcc 12 Fri Sep 29 10:05:24 2023

   Double Precision Benchmark 1, 2, 4 and 8 Threads

                       MFLOPS          Numeric Results
            Ops/   KB     KB    MB      KB     KB     MB
  Secs  Thrd Word 12.8   128  12.8    12.8    128   12.8

   0.9    T1    2  6570  6565  1003   40395  76384  99700
   1.9    T2    2 12052 13057   696   40395  76384  99700
   2.7    T4    2 22815 25654   831   40395  76384  99700
   3.5    T8    2 21088 25978   838   40395  76384  99700
   4.9    T1    8  8348  8388  3290   54805  85108  99820
   6.3    T2    8 15906 16532  2530   54805  85108  99820
   7.3    T4    8 23730 28755  2932   54805  85108  99820
   8.3    T8    8 30036 30142  3327   54805  85108  99820
  11.4    T1   32 10027  9975  9486   35159  66065  99521
  13.3    T2   32 19719 19508 12462   35159  66065  99521
  14.6    T4   32 40249 39892 13452   35159  66065  99521
  15.9    T8   32 38383 39453 13637   35159  66065  99521

           End of test Fri Sep 29 10:05:40 2023
```

Continued Below or Go To Start

```
 Pi 5 GCC 8 DP
   MP-Threaded-MFLOPS 64 Bit gcc 8 Thu Aug 17 21:29:32 2023

    Double Precision Benchmark 1, 2, 4 and 8 Threads

                          MFLOPS          Numeric Results
              Ops/   KB     KB     MB      KB      KB      MB
   Secs  Thrd Word 12.8    128   12.8    12.8     128    12.8

    0.9    T1    2  5832  5779    964   40395   76384   99700
    1.8    T2    2 11389 11537    891   40395   76384   99700
    2.6    T4    2 18744 21914    794   40395   76384   99700
    3.5    T8    2 18803 22948    842   40395   76384   99700
    4.7    T1    8  9375  9433   3984   54805   85108   99820
    5.9    T2    8 18190 18819   2758   54805   85108   99820
    6.8    T4    8 33842 37329   3233   54805   85108   99820
    7.7    T8    8 33857 34347   3393   54805   85108   99820
   10.9    T1   32  9633  9642   9458   35159   66065   99521
   12.7    T2   32 19227 19248 14292   35159   66065   99521
   14.0    T4   32 37215 38597 13208   35159   66065   99521
   15.4    T8   32 35943 36029 13288   35159   66065   99521

             End of test Thu Aug 17 21:29:47 2023

 Pi 5 GCC 12
   MP-Integer-Test 64 Bit v2-gcc12 Fri Sep 29 10:11:39 2023

       Benchmark 1, 2, 4, 8, 16 and 32 Threads

                      MB/second
                   KB     KB     MB            Same All
    Secs Thrds     16    160     16  Sumcheck   Tests

    1.5     1  18233 17590 13957  00000000    Yes
    1.1     2  36284 35095 13303  FFFFFFFF    Yes
    1.0     4  71208 73154 11228  5A5A5A5A    Yes
    1.0     8  64036 68274 11499  AAAAAAAA    Yes
    0.9    16  70658 71792 12459  CCCCCCCC    Yes
    0.5    32  69044 72425 26917  0F0F0F0F    Yes

             End of test Fri Sep 29 10:11:45 2023

 Pi 5 GCC 8
   MP-Integer-Test 64 Bit v2-gcc8 Thu Aug 17 21:32:43 2023

       Benchmark 1, 2, 4, 8, 16 and 32 Threads

                      MB/second
                   KB     KB     MB            Same All
    Secs Thrds     16    160     16  Sumcheck   Tests

    1.7     1  15193 15083 13106  00000000    Yes
    1.2     2  30256 30277 13472  FFFFFFFF    Yes
    1.0     4  58317 60842 11173  5A5A5A5A    Yes
    1.0     8  56279 54906 12132  AAAAAAAA    Yes
    0.9    16  54716 59296 13475  CCCCCCCC    Yes
    0.5    32  53649 59206 34738  0F0F0F0F    Yes

             End of test Thu Aug 17 21:32:49 2023
```

Stress Tests - No Fan next or Go To Start

Floating Point and Integer Stress Tests - No Fan

Following are early gcc8 compiled result summaries for the first stress tests without a fan being fitted. They were for 15 minutes, using 1, 2 and 4 threads, measuring average performance over 10 seconds and samples of MHz, Volts and temperatures recordings within that period. The summaries are 5 sets of performance results at the beginning, middle and end, then minimum and maximum values of each column, plus maximum/minimum calculations. Note that, for more than 1 thread, share of data should fit in L1 caches of the utilised cores. Every test ran successfully but identified MHz throttling, with performance degradation between 23% and 55%, besides lower MHz due to throttling, and some voltage reductions. At the end of the integer 4 thread tests, temperatures of up to 90°C were recorded and some CPU clock speeds of 1000 MHz.

	Floating Point Stress Test 128 KB					Integer Stress Test 160 KB				
Seconds	MFLOPS	MHz	Volts	CPU °C	PMIC °C	MB/sec	MHz	Volts	CPU °C	PMIC °C
1 Thread										
0		2400	0.9065	68.6	61.8		2400	0.9065	71.9	64.8
10	18279	2400	0.9065	73.0	63.0	15128	2400	0.9065	77.4	66.0
20	18273	2400	0.9065	76.8	63.7	15132	2400	0.9065	78.5	66.8
30	18284	2400	0.9065	75.2	64.4	15094	2400	0.9065	79.0	67.4
40	18283	2400	0.9065	78.5	65.0	15095	2400	0.9065	81.8	68.1
50	18277	2400	0.9065	79.0	65.7	15117	2400	0.9065	82.3	68.9
420	16459	2201	0.7200	84.5	72.8	12906	2146	0.9065	85.1	73.3
430	16396	2146	0.9065	85.1	72.8	11522	1500	0.9065	84.0	73.0
440	16440	2256	0.9065	84.5	72.6	12905	1500	0.9065	84.5	73.3
450	14862	1500	0.9065	86.2	72.5	12437	1500	0.9065	84.5	73.2
460	15332	2146	0.9065	84.5	72.5	11505	1500	0.9065	85.1	73.0
860	15370	2256	0.9065	84.0	72.3	12181	1500	0.7200	85.1	73.6
870	15318	2201	0.9065	84.5	72.5	11929	2146	0.9065	84.0	73.3
880	17227	2201	0.7200	84.0	72.8	13275	2201	0.9065	84.5	73.2
890	16381	1500	0.9065	85.6	72.5	12913	1500	0.9065	84.0	73.4
900	16364	2201	0.7200	82.9	72.4	11974	1500	0.9065	84.5	73.2
Max	18284	2400	0.9065	86.2	72.8	15132	2400	0.9065	85.1	73.6
Min	14862	1500	0.72	68.6	61.8	11505	1500	0.72	71.9	64.8
Max/Min	1.23	1.60	1.26	1.26	1.18	1.32	1.60	1.26	1.18	1.14
2 Threads										
0		2400	0.9065	71.4	64.2		2400	0.9065	71.9	64.4
10	36520	2400	0.9065	79.0	66.8	30425	2400	0.9065	80.7	66.7
20	35794	2311	0.9065	84.0	68.1	29123	2256	0.9065	84.0	67.8
30	33156	2256	0.7200	84.5	69.3	28064	2256	0.9065	85.1	68.9
40	31361	2146	0.7200	85.1	70.0	25692	2201	0.9065	84.0	69.4
50	30525	2146	0.9065	85.1	70.8	25456	1500	0.9065	84.0	70.1
420	27102	1500	0.7200	84.5	73.5	21687	1500	0.7200	85.6	73.8
430	26742	2146	0.7200	85.1	73.5	20675	1500	0.9065	86.2	73.9
440	27006	1500	0.9065	85.6	73.4	20980	1500	0.7200	85.6	73.6
450	27092	2201	0.7200	85.6	73.5	21997	1500	0.7200	85.1	73.9
460	26822	1500	0.9065	85.6	73.3	20854	1500	0.7200	85.1	73.6
860	26691	2146	0.7200	85.1	73.9	21072	2146	0.7200	85.1	73.9
870	26989	1500	0.7200	85.1	73.9	21111	1500	0.7200	85.6	73.6
880	28018	1500	0.7200	85.1	73.9	21035	1500	0.9065	85.6	73.6
890	27595	1500	0.9065	85.6	73.9	21011	2256	0.7200	84.5	73.8
900	26449	2256	0.7200	85.1	74.0	21028	1500	0.7200	84.5	73.8
Max	36520	2400	0.9065	85.6	74.0	30425	2400	0.9065	86.2	73.9
Min	26449	1500	0.7200	71.4	64.2	20675	1500	0.7200	71.9	64.4
Max/Min	1.38	1.60	1.26	1.20	1.15	1.47	1.60	1.26	1.20	1.15
4 Threads										
0		2400	0.9065	71.4	64.3		2400	0.9065	70.8	64.3
10	61133	1500	0.9065	85.1	68.0	52566	2256	0.7200	83.4	68.1
20	52128	1500	0.7200	85.6	69.1	44870	1500	0.7200	84.5	69.2
30	50301	1500	0.7200	85.1	70.8	43266	2256	0.7200	85.1	70.0
40	49068	1500	0.9065	86.2	71.0	42129	2201	0.7200	84.5	71.2
50	48448	2201	0.9065	87.3	71.6	41617	1500	0.7200	85.1	71.4
420	45854	1500	0.7200	86.2	74.3	34701	1500	0.7200	89.5	76.6
430	45456	1500	0.7200	86.2	74.3	35108	1500	0.7200	88.4	76.6
440	45859	1500	0.7200	85.6	74.3	35034	1500	0.7200	90.0	76.6
450	45853	1500	0.7200	85.6	74.3	35099	1500	0.7200	88.9	76.5
460	45810	1500	0.7200	85.1	74.3	35176	1000	0.7200	89.5	76.6
860	45686	1500	0.7200	85.1	74.3	34503	1500	0.7200	88.9	76.8
870	45337	1500	0.7200	84.5	74.3	34056	1500	0.7200	90.0	77.0
880	46261	1500	0.7200	85.6	74.3	34053	1500	0.7200	88.9	76.6
890	45069	1500	0.7200	86.2	74.3	33955	1500	0.7200	89.5	77.0
900	45285	1500	0.7200	86.2	74.6	34188	1500	0.7200	90.0	76.9
Max	61133	2400	0.9065	87.3	74.6	52566	2400	0.9065	90.0	77.0
Min	45069	1500	0.7200	71.4	64.3	33955	1000	0.7200	70.8	64.3
Max/Min	1.36	1.60	1.26	1.22	1.16	1.55	2.40	1.26	1.27	1.20

Integer Stress Tests - With Fan next or Go To Start

Integer Stress Tests - With Fan

The fan came as part of a 2019 GeeekPi Acrylic Case for Raspberry Pi 4 Model B, probably not powerful enough for the Pi 5.

The results provided cover data from L1 and L2 caches, with a starting temperature around 40°C, in a room at 26°C to 27°C. One example made use of one thread, running continuously at full speed and reaching a maximum CPU temperature of 57.1°C. Similarly, one used two threads and ran at full speed, with temperature up to 70.3°C.

There are four examples using 4 threads with KB of data 128, 512, and two at 1024 (to show variations). These all have maximum CPU temperatures indicated as between 84.5°C and 85.1°C with MHz throttling, maximum speeds of around around 60 GB/second and minimum about 51 GB/second. Example using 1 and 2 threads indicated constant performance near 15 and 30 GB/second respectively, all at 2400 MHz.

	4 Threads 128 KB 4 x L1 Cache					4 threads 1024 KB 4 x L2 Cache				
Seconds	MB/sec	MHz	Volts	CPU °C	PMIC °C	MB/sec	MHz	Volts	CPU °C	PMIC °C
0		2400	0.9067	38.9	40.1		2400	0.9067	41.1	39.9
10	59953	2400	0.9067	57.6	43.8	60553	2400	0.9067	56.0	43.7
20	59448	2400	0.9067	67.0	47.3	60320	2400	0.9067	63.7	45.9
30	60019	2400	0.9067	70.8	50.0	59929	2400	0.9067	67.0	47.9
420	51124	2256	0.9067	84.5	62.2	53503	2256	0.9067	84.5	61.4
430	51011	2146	0.9067	84.5	62.2	53653	2256	0.9067	84.0	61.0
440	51219	2256	0.9067	84.5	62.4	53297	2146	0.9067	84.5	61.4
860	50943	2201	0.9067	84.5	62.1	53756	2201	0.9067	83.4	61.7
870	51446	2311	0.9067	84.0	62.3	53352	2146	0.9067	83.4	61.7
880	51378	2146	0.7200	82.3	61.9	54173	2201	0.9067	84.5	61.7
Max	60025	2400	0.9067	84.5	62.4	60553	2400	0.9067	84.5	61.7
Min	50943	2146	0.7200	38.9	40.1	53157	2146	0.7200	41.1	39.9
Max/Min	1.18	1.12	1.26	2.17	1.56	1.14	1.12	1.26	2.06	1.55

	4 Threads 512 KB 4 x L2 Cache					1 Thread 512 KB L2 Cache				
0		2400	0.9067	41.7	40.5		2400	0.9067	40.6	39.5
10	58969	2400	0.9067	59.8	44.9	14995	2400	0.9067	46.6	40.7
20	59611	2400	0.9067	66.4	47.2	15070	2400	0.9067	48.8	42.1
30	59488	2400	0.9067	70.8	50.0	15018	2400	0.9067	50.5	43.1
420	51217	1500	0.9067	84.0	62.1	15068	2400	0.9067	54.3	47.0
430	50975	2201	0.9067	85.1	61.5	15081	2400	0.9067	53.2	46.9
440	51841	2256	0.9067	84.0	62.3	15064	2400	0.9067	53.8	46.8
860	51128	2146	0.9067	85.1	61.3	15031	2400	0.9067	56.5	48.2
870	50938	2311	0.9067	84.5	62.1	15074	2400	0.9067	56.5	48.1
880	51460	2400	0.9067	84.0	61.7	15055	2400	0.9067	57.1	48.1
3560	51254	1500	0.9067	84.0	62.4	15038	2400	0.9067	56.5	47.8
3570	51414	2146	0.9067	85.1	61.8	15062	2400	0.9067	56.5	47.7
3580	51197	1500	0.9067	84.5	62.2	15051	2400	0.9067	56.5	47.7
Max	59611	2400	0.9067	85.1	62.4	15081	2400	0.9067	57.1	48.2
Min	50938	1500	0.72	41.7	40.5	14995	2400	0.9067	40.6	39.5
Max/Min	1.17	1.60	1.26	2.04	1.54	1.01	1.00	1.00	1.41	1.22

	2 Threads 512 KB 2 x L2 Cache					4 Threads 1024 KB 4 x L2 Cache				
0		2400	0.9067	39.5	40.0		2400	0.9065	41.1	39.7
10	30115	2400	0.9067	51.0	42.5	59776	2400	0.9065	57.6	44.2
20	30172	2400	0.9067	54.9	43.8	59619	2400	0.9065	67.0	47.0
30	30254	2400	0.9067	55.4	45.0	59773	2400	0.9065	70.8	49.7
420	30258	2400	0.9067	70.3	53.0	51820	2311	0.7200	84.0	62.0
430	30295	2400	0.9067	70.3	53.1	51644	2201	0.7200	82.9	61.3
440	30272	2400	0.9067	68.6	53.2	51512	2146	0.9065	84.5	62.1
860	30265	2400	0.9067	69.2	53.1	52739	2201	0.9065	83.4	61.7
870	30252	2400	0.9067	68.1	53.4	52652	2400	0.9065	84.5	61.5
880	30289	2400	0.9067	68.1	53.2	50956	2201	0.9065	84.5	61.8
3560	30274	2400	0.9067	69.7	53.2	51051	2311	0.9065	84.5	62.5
3570	30296	2400	0.9067	68.6	53.2	51008	2146	0.7200	82.3	62.5
3580	30246	2400	0.9067	68.6	53.2	51157	1500	0.9065	83.4	62.5
Max	30296	2400	0.9067	70.3	53.4	59812	2400	0.9065	84.5	62.5
Min	30115	2400	0.9067	39.5	40.0	50776	1500	0.7200	41.1	39.7
Max/Min	1.01	1.00	1.00	1.78	1.34	1.18	1.60	1.26	2.06	1.57

Floating Point Stress Tests - With Fan next or Go To Start

Floating Point Stress Tests - With Fan

Only two set of results are provided both using 4 threads, with the same data size of 512 KB, one with 2 floating point operations per data word, starting at 51.2 GFLOPS, and the other with 32 floating point operations per data word, starting at 72.3 GFLOPS. At the end of the 15 minutes runs, performance was indicated at 43.3 and 72.2 GFLOPS respectively, the slower one running at higher temperatures. The fastest near constant performance was confirmed by constant CPU MHz reports.

Estimating data flow from MFLOPS and Ops/Word indicates that the test with the slower CPU performance has a much higher data transfer speed and that can influence CPU temperatures.

	4 Threads 2 Ops/Word 512 KB 4 x L2					4 reads 32 Ops/Word 512 KB 4 x L2				
Seconds	MFLOPS	MHz	Volts	CPU °C	PMIC °C	MFLOPS	MHz	Volts	CPU °C	PMIC °C
0		2400	0.9067	41.7	41.2		1500	0.9067	40.0	40.6
10	51228	2400	0.9067	65.9	48.3	72366	2400	0.9067	59.3	44.6
20	50610	2400	0.9067	76.8	52.3	72350	2400	0.9067	67.0	47.3
30	50799	2400	0.9067	82.3	55.9	72370	2400	0.9067	70.3	49.3
40	51452	2201	0.9067	83.4	57.7	72348	2400	0.9067	71.9	51.2
50	50451	2256	0.9067	82.9	59.0	72212	2400	0.9067	74.1	52.6
420	43777	1500	0.9067	84.0	62.3	72348	2400	0.9067	81.2	58.9
430	43870	2400	0.9067	84.5	62.5	72381	2400	0.9067	81.2	58.9
440	43733	2201	0.9067	84.0	62.3	72617	2400	0.9067	80.7	58.9
450	43887	2146	0.9067	84.5	61.7	72201	2400	0.9067	80.7	58.8
460	43609	2201	0.9067	85.1	61.9	72229	2400	0.9067	81.2	58.9
860	43726	2366	0.9067	84.5	62.3	72294	2400	0.9067	81.2	59.2
870	43346	2201	0.9067	84.5	62.3	72465	2400	0.9067	81.2	59.1
880	44063	2146	0.9067	85.1	61.9	72257	2400	0.9067	81.8	59.3
890	43412	2201	0.9067	84.5	62.2	72173	2400	0.9067	81.2	59.2
900	43353	2146	0.9067	84.5	62.5	72163	2366	0.9067	81.2	59.2
Max	51452	2400	0.9067	85.1	62.5	72617	2400	0.9067	81.8	59.3
Min	43346	1500	0.9067	41.7	41.2	72163	1500	0.9067	40.0	40.6
Max/Min	1.19	1.60	1.00	2.04	1.52	1.01	1.60	1.00	2.05	1.46

Stress Test Parameters

The following show stress test run time parameters. The classifications can be upper or lower case and only the first character is interpreted.

```
./MP-FPUStress   Threads tt, Minutes mm, KB kk, Ops 00, Log ll
./MP-FPUStressDP Threads tt, Minutes mm, KB kk, Ops 00, Log ll
./MP-IntStress   Threads tt, Minutes mm, KB kk, Log ll
./RPiHeatMHzVolts2  Passes pp, Seconds ss, Log ll
vmstat ss pp

tt = Threads 1, 2, 4, 8, 16, 32, (64 FPU)       mm = Minutes greater than 0
kk = KBytes 12 to 15624                         oo = Operations Per Word 2, 8 or 32
ll = number added to log file name, 0 to 99     pp = Passes (at ss econd intervals)
ss = Second intervals
```

New Power Supply below or Go To Start

New 4 Amps Power Supply No Disk Crash

Earlier I reported that the Pi 5 crashed when running a stress test on a USB based disk drive along with one executing integer calculations via four threads. A 3 amps power supply was in use.

With no 5 amps power supplies being available, I investigated the Power over Ethernet (PoE) route. My existing Power Injector and Splitter were limited to providing 2.5 amps. There are lots of Injectors delivering 25 or 30 watts but I could not find a Splitter producing 5 amps at 5 volts. However, I acquired a GeeekPi Gigabit USB-C PoE Splitter 48V to 5V, 4A and YuanLey Gigabit PoE Injector 30W, PoE+.

They did not explode on connecting them and I was able to run those tests successfully, once with SD booting and disk on USB 3 and second booting and testing a disk on a USB 3 hub. My monitors typically indicated power in 5.2V 2.8A and USB supply 4.9V and 0.75A.

New INTitHOT Integer Stress Test below o or Go To Start

New Integer Stress Test - INTitHOT64g12

Above, I showed that my MP-BusSpeed benchmark could achieve a data transfer rate of 150 GB/second. I have now converted the particular procedures to work as a stress test, with variable options that operate at up to 168 GB/second. Later, 240 GB/second was obtained using L1 cache sized data. As the program is executing AND instructions, this demonstrated Terabit performance at 1.92 Tbps.

The tests identified three particular problems. With no fan, CPU temperature appeared to reach 90°C. Then, with a fan, current draw was indicated as being up to 2.3 amps. Also, in both cases there was significant CPU MHz throttling

Following is the C program function calculations and main disassembled code. It is effectively a read only test of 64 words, from a large array, executing AND instructions for a one word output. Each thread exercises a dedicated segment of the data, circulated on a round robin basis, reading all data every pass. The disassembly shows (I believe) loading data into eight pairs of quad word registers, then sixteen quad word AND operations.

In case of anybody is interested in running (or modifying), the program, the source and compiled codes, along with my environmental monitor are available from ResearchGate in INTitHOT.tar.xz.

Test Function Calculations

```
andsum1[t] = andsum1[t] & array[i   ] & array[i+1 ] & array[i+2 ] & array[i+3 ]
                        & array[i+4 ] & array[i+5 ] & array[i+6 ] & array[i+7 ]
                        & array[i+8 ] & array[i+9 ] & array[i+10] & array[i+11]
                        & array[i+12] & array[i+13] & array[i+14] & array[i+15]
                        & array[i+16] & array[i+17] & array[i+18] & array[i+19]
                        & array[i+20] & array[i+21] & array[i+22] & array[i+23]
                        & array[i+24] & array[i+25] & array[i+26] & array[i+27]
                        & array[i+28] & array[i+29] & array[i+30] & array[i+31]
                        & array[i+32] & array[i+33] & array[i+34] & array[i+35]
                        & array[i+36] & array[i+37] & array[i+38] & array[i+39]
                        & array[i+40] & array[i+41] & array[i+42] & array[i+43]
                        & array[i+44] & array[i+45] & array[i+46] & array[i+47]
                        & array[i+48] & array[i+49] & array[i+50] & array[i+51]
                        & array[i+52] & array[i+53] & array[i+54] & array[i+55]
                        & array[i+56] & array[i+57] & array[i+58] & array[i+59]
                        & array[i+60] & array[i+61] & array[i+62] & array[i+63];
```

Inner Loop Disassembly

```
.L128:
       ldp q31, q30, [x0]
add  w13,  w13,  1
ldp  q29,  q28,  [x0,  32]
ldp  q27,  q26,  [x0,  64]
ldp  q25,  q24,  [x0,  96]
ldp  q23,  q22,  [x0,  128]
ldp  q21,  q20,  [x0,  160]
ldp  q19,  q18,  [x0,  192]
ldp  q17,  q16,  [x0,  224]
add  x0,  x0,  256
and  v15.16b,  v15.16b,  v31.16b
and  v0.16b,  v0.16b,  v30.16b
and  v14.16b,  v14.16b,  v29.16b
and  v13.16b,  v13.16b,  v28.16b
and  v12.16b,  v12.16b,  v27.16b
and  v11.16b,  v11.16b,  v26.16b
and  v10.16b,  v10.16b,  v25.16b
and  v9.16b,  v9.16b,  v24.16b
and  v8.16b,  v8.16b,  v23.16b
and  v7.16b,  v7.16b,  v22.16b
and  v6.16b,  v6.16b,  v21.16b
and  v5.16b,  v5.16b,  v20.16b
and  v4.16b,  v4.16b,  v19.16b
and  v3.16b,  v3.16b,  v18.16b
and  v2.16b,  v2.16b,  v17.16b
and  v1.16b,  v1.16b,  v16.16b
cmp  w2,  w13
bhi  .L128
```

INTitHOT PI 5 and Pi 4 Maximum Speeds below o or Go To Start

INTitHOT PI 5 and Pi 4 Maximum Speeds - With Fan

The INTitHOT tests were run with the fan operational to demonstrate maximum speeds over the first few passes, using the same run time parameters on the Pi 5 and Pi 4. These accessed 64 KB using 1, 2 and 4 threads. Here, near constant elapsed times at all thread levels indicate high efficiency. This applied to the Pi 5 results. But, for an inexplicable reason, the Pi 4 failed to benefit from using 4 threads. Note that the latter system was booted and used via the Pi 5 OS SD card.

Pi 5 performance gains over Pi 4 results were 3.94 and 4.62 at 1 and 2 threads and maybe 10 times at 4 threads. Fastest Pi 5 performance was 240 Gigabytes per second, using 4 threads. This indicates the equivalent of 120 Giga Instructions Per Second (GIPS) or 60 Giga Integer Arithmetic Operations Per Second (GIAOPS).

Also below are maximum speeds using 9 data sizes between 64 and 16384 KB. This test was included in my benchmark, intended to measure bus speeds. In this case, the memory bus speed is indicated as 27 GB/second. Here, at 16 MB data size, each of the 4 threads would be cycling through dedicated segments of 4 MB. Maximum observed current draw was 2.3 amps at 512 KB data size, higher than at 64 KB but with slower performance.

```
 Pi 5                                    Pi 4

 INTitHOT 64 Bit gcc 12 Thu              INTitHOT 64 Bit gcc 12 Thu
 Oct 19 15:51:53 2023                    Oct 19 15:11:35 2023

 1 Threads. 64 KBytes, 500000            1 Threads. 64 KBytes, 500000
 Passes 1+ Minutes                       Passes 1+ Minutes

 Repeat   MB/second  Seconds             Repeat   MB/second  Seconds

    1         56796    0.58                 1         14418    2.27
    2         56612    0.58                 2         14412    2.27
    3         56704    0.58                 3         14404    2.27

####################################    ####################################

 INTitHOT 64 Bit gcc 12 Thu              INTitHOT 64 Bit gcc 12 Thu
 Oct 19 15:51:16 2023                    Oct 19 15:11:06 2023

 2 Threads. 64 KBytes, 500000            2 Threads. 64 KBytes, 500000
 Passes 1+ Minutes                       Passes 1+ Minutes

 Repeat   MB/second  Seconds             Repeat   MB/second  Seconds

    1        113194    0.58                 1         24510    2.67
    2        113663    0.58                 2         24415    2.68
    3        113272    0.58                 3         24412    2.68

####################################    ####################################

 INTitHOT 64 Bit gcc 12 Thu              INTitHOT 64 Bit gcc 12 Thu
 Oct 19 15:50:53 2023                    Oct 19 15:10:29 2023

 4 Threads. 64 KBytes, 500000            4 Threads. 64 KBytes, 500000
 Passes 1+ Minutes                       Passes 1+ Minutes

 Repeat   MB/second  Seconds             Repeat   MB/second  Seconds

    1        240850    0.54                 1         23839    5.50
    2        231406    0.57                 2         23832    5.50
    3        240861    0.54                 3         23836    5.50

####################################    ####################################

 Pi 5 4 Threads Maximum speeds         Power

 Passes    KB MB/sec    Secs             amps

 500000    64 240850    0.54 L1          1.8 to 1.9
 500000   128 165221    1.59 L2          1.9 to 2.0
 500000   256 168499    3.11             1.9 to 2.0
 500000   512 158777    6.64             2.1 to 2.3
  50000   512 158019    0.66             2.1 to 2.3
  50000  1024  73043    2.87 L3          1.8 to 1.9
  50000  2048  52050    8.06 L3          1.7 to 1.8
  50000  4096  32024   26.18 RAM         1.6 to 1.7
  50000  8192  30767   54.53             1.5 to 1.6
  50000 16384  26983  124.35             1.5 to 1.7
```

INTitHOT Stress Tests next or Go To Start

INTitHOT Stress Tests

The tests were all run for 15 minutes using 4 threads, covering two data sizes, 64 KB for the fastest via L1 caches and the hottest at 512 KB using L2 caches. In a table, each performance measurement is for the same pass count, where the time taken can increase due to CPU MHz throttling. The environmental monitor was run at the same time, sampling at 39 second intervals.

Later the full details are provided of the two test sessions running with the fan cooling disconnected and the default CPU frequency ondemand scaling setting used. Others with the performance setting were also run, providing similar long term variations in performance. Here, we have summaries of fan and no fan situations.

With no fan in use, there was significant CPU MHz throttling at both data sizes, less so at 64 KB with the higher KB/second data transfer speeds.

With fan cooling, the 64 KB test was not affected much by MHz throttling, suffering by a mere 5% degradation in performance, compared with 16% at 512 KB, with additional throttling but not that much increase in CPU temperature.

	MB/sec	Secs	MHz	Volts	CPU °C	PMIC °C
64 KB No Fan						
Min	150715	16.4	1500	0.7200	42.8	44.2
Max	240498	26.1	2256	0.9060	87.3	75.4
Average			1689	0.7492	84.0	71.5
512 KB No Fan						
Min	84743	29.0	1000	0.7200	47.7	47.3
Max	144811	49.5	2146	0.9060	90.0	77.4
Average			1380	0.7433	86.8	74.1
64 KB Fan						
Min	228738	32.7	2256	0.9067	41.7	39.9
Max	240414	34.4	2400	0.9067	84.0	60.1
Average			2306	0.9067	82.3	59.7
512 KB Fan						
Min	124143	29.2	1500	0.7200	41.7	43.0
Max	143845	33.8	2400	0.9060	85.6	62.5
Average			2193	0.8700	83.6	61.5

INTitHOT Stress Test 64 KB next or Go To Start

INTitHOT Stress Test 64 KB - No Fan

The fan was not successful in controlling the CPU temperature that reached 85.6°C, leading to a 14% reduction in measured performance. The temperature, CPU MHz and voltage had regular variations.

PI 5 Stress Test 64 KB, no fan, ondemand MHz scaling

INTitHOT Fri Oct 20 11:20:38 Temperature and CPU MHz Measurement
4 Threads 64 KB 15000000 Passes Start at Fri Oct 20 11:20:33 2023

Repeat	MB/sec	Secs	Seconds	MHz	Volts	CPU °C	PMIC °C
			0	1500	0.9060	42.8	44.2
1	240498	16.4	30	2256	0.9060	83.4	58.8
2	225209	17.5	60	1500	0.9060	85.6	65.4
3	195713	20.1	91	1500	0.9060	86.2	69.0
4	182682	21.5	121	1500	0.7200	84.5	71.4
5	172867	22.8	151	1500	0.7200	85.1	72.0
6	166663	23.6	182	1500	0.7200	85.1	72.5
7	163066	24.1	212	2146	0.7200	86.2	73.1
8	160312	24.5	242	1500	0.7200	84.5	73.9
9	158921	24.7	273	1500	0.7200	85.6	73.4
10	157789	24.9	303	1500	0.7200	85.1	73.8
11	156465	25.1	334	1500	0.7200	85.6	73.8
12	154721	25.4	364	1500	0.7200	85.6	73.8
13	155261	25.3	394	1500	0.7200	85.1	73.9
14	154156	25.5	425	1500	0.7200	86.2	74.2
15	153030	25.7	455	1500	0.7200	86.2	74.1
16	152971	25.7	485	1500	0.7200	86.2	74.5
17	153125	25.7	515	1500	0.7200	85.6	74.5
18	152132	25.9	546	1500	0.7200	85.6	74.5
19	152081	25.9	576	1500	0.7200	86.2	74.8
20	152261	25.8	606	1500	0.7200	86.2	74.8
21	151389	26.0	637	1500	0.7200	85.6	74.6
22	151139	26.0	667	1500	0.7200	86.7	74.9
23	151028	26.0	697	1500	0.7200	86.7	75.0
24	151525	26.0	728	1500	0.7200	86.2	75.1
25	151101	26.0	758	1500	0.7200	86.7	75.0
26	151200	26.0	788	1500	0.7200	86.2	75.2
27	151501	26.0	819	1500	0.7200	87.3	75.2
28	150845	26.1	849	1500	0.7200	86.7	75.4
29	150795	26.1	879	1500	0.7200	86.7	75.2
30	150715	26.1	910	1500	0.7200	87.3	75.2
31	151059	26.0	940	1500	0.9060	76.8	72.8
32	150767	26.1					
33	150751	26.1					
34	150959	26.1					

35	150927	26.1				
36	150783	26.1				
37	151009	26.0				
Min	150715	16.4	1500	0.7200	42.8	44.2
Max	240498	26.1	2256	0.9060	87.3	75.4
Average			1689	0.7492	84.0	71.5

INTitHOT Stress Test 512 KB next or Go To Start

INTitHOT Stress Test 512 KB - No Fan

This recorded the highest temperatures at 90°C and 42% reduction in MB/second, with lowest CPU frequency regularly at 1000 MHz. Voltage was mainly constant at 0.7200 along with temperature near the top end.

```
PI 5 Stress Test Detail - 512 KB,  no fan, ondemand MHz scaling

INTitHOT Fri Oct 20 10:49:05     Temperature and CPU MHz Measurement
4 Threads 512 KB 2000000 Passes  Start at Fri Oct 20 10:48:58 2023

  Repeat  MB/sec    Secs         Seconds     MHz   Volts  CPU °C PMIC °C
                                       0    1500  0.9060    47.7    47.3
       1  144811    29.0              30    1500  0.9060    84.5    62.8
       2  117807    35.6              60    1500  0.9060    86.7    67.7
       3  109939    38.2              91    2146  0.7200    85.1    70.3
       4  106055    39.6             121    1500  0.7200    85.6    71.3
       5  104401    40.2             152    1500  0.7200    85.6    72.2
       6  103921    40.4             182    1500  0.7200    85.1    72.6
       7  103770    40.4             212    1500  0.7200    86.7    73.1
       8  103705    40.4             243    1500  0.7200    87.8    74.1
       9  101765    41.2             273    1500  0.7200    87.8    74.9
      10   98730    42.5             303    1500  0.7200    88.9    75.3
      11   96339    43.5             334    1500  0.7200    89.5    75.8
      12   93876    44.7             364    1500  0.7200    89.5    76.0
      13   92469    45.4             394    1500  0.7200    90.0    76.0
      14   90528    46.3             425    1000  0.7200    89.5    76.2
      15   88594    47.3             455    1500  0.7200    88.9    76.3
      16   88113    47.6             485    1500  0.7200    88.4    76.6
      17   87023    48.2             515    1500  0.7200    90.0    76.5
      18   86581    48.4             546    1500  0.7200    90.0    77.0
      19   85699    48.9             576    1500  0.7200    89.5    77.1
      20   84743    49.5             606    1000  0.7200    88.9    77.0
      21   84760    49.5             637    1000  0.7200    90.0    77.0
                                     667    1000  0.7200    88.4    77.2
                                     698    1000  0.7200    88.4    77.2
                                     728    1500  0.7200    89.5    77.3
                                     758    1000  0.7200    89.5    77.2
                                     789    1000  0.7200    90.0    77.3
                                     819    1500  0.7200    90.0    77.2
                                     849    1000  0.7200    90.0    77.2
                                     880    1500  0.7200    89.5    77.4
                                     910    1000  0.7200    89.5    77.4
                                     940    1500  0.9060    75.7    73.0

     Min   84743   28.96                    1000  0.7200    47.7    47.3
     Max  144811   49.49                    2146  0.9060    90.0    77.4
 Average                                    1380  0.7433    86.8    74.1
```

32 Bit System Stress Tests below or Go To Start

System Stress Tests

All these tests were run for 30 minutes, exercising the CPU, graphics and data input/output and included my environment and VMSTAT performance monitors, the, latter to validate the program MBytes per second measurements and confirm that CPU utilisation was at the expected near 100% level. A script file was used to to ensure that the programs started in at the same time. In most cases, performance was measured or sampled every 60 seconds.

An example script file is below, also the commands to run the OpenGL program from a separate terminal, with VSYNC turned off to produce maximum frames per second (FPS).

```
Script File

lxterminal -e ./RPiHeatMHzVolts64 Passes 31 Seconds 60 Log 7 &
lxterminal -e ./INTitHOT64g12 threads 2, kBStress 64, Minutes 30, passCount 4000000, logNumber 7 &
lxterminal -e ./MP-FPUStress64g12 threads 2, kb 512, ops 32, Minutes 30, log 7 &
lxterminal -e sudo ./burnindrive264g12 Repeats 16, Minutes 27, Log 8, Seconds 1, F /media/raspberrypi/public/ray &
lxterminal -e sudo ./burnindrive264g12 Repeats 16, Minutes 27, Log 9, Seconds 1, F /media/raspberrypi/EXT3 &
lxterminal -e vmstat 60 30 . vmstat7.txt

Separate Terminal

export vblank_mode=0
./videogl64C12 Test 6 Minutes 30
```

Of particular note, the first set of tests identifies increases in CPU temperature up to 91.7°C, with no fan running.

A questionable more significant problem, during the second set of tests, was the disk program indicating errors and the drive temporarily dropping off line during a test with the fan operational. The errors were the same as on earlier runs using a 3 amps power supply, the present PoE connection supposedly providing 4 amps.

Monitoring the input power used and that supplied for the USB drive, indicated that consumption was fairly constant between 2 and 15 minutes testing time, providing the following typical meter readings. These suggest that the disk drive might be more vulnerable to failure when the CPU is fully loaded and CPU MHz throttling might be useful if danger can be predicted.

No Fan Poor CPU Performance				With Fan Good CPU Performance			
Power		USB		Power		USB	
Volts	Amps	Volts	Amps	Volts	Amps	Volts	Amps
5.26	1.75	5.06	0.53	5.20	2.60	4.94	0.53

Light System Stress Test below or Go To Start

Light System Stress Test

The first sessions involved INTitHOT64g12, using 4 threads accessing 512 KB data, with a pass count to control minimum running time. Then, with this test, total running time was specified as 30 minutes, leading to fewer results when the CPU MHz was throttled. These MB/second results were allocated at two minute intervals. Other inclusions were burnindrive264g12 to a USB3 disk drive, plus videogl64C12 accessing the most demanding display test, producing FPS results every 30 seconds, with results provided at 60 second intervals, as shown in the detailed tables below.

Following are two sets of results for one run with the fan in use and one without the fan. On the bright side, these and a number of other tests, using the same parameters, ran without any issues. But CPU MHz throttling occurred in all cases.

Summaries

Minimum values are often isolated examples and can often be ignored. Best scores shown at the head of the table are from standalone runs. Maximum benchmark performance measurements suffer from being noted a minute after start time. Averages indicate significant reductions for the integer and OpenGL tests but little difference on disk drive data transfer speeds.

Of particular note is the CPU temperature measurement of 91.7°C with the fan out of use.

	MHz	Volts	CPU °C	PMIC °C	Integer MB/sec	VMSTAT Disk KB/sec	OpenGL FPS
Best					145000	63000	102
512 KB FAN							
Average	2128	0.8878	82.8	61.8	97568	60368	65.3
Min	1500	0.7200	42.2	39.7	95281	59159	61.0
Max	2400	0.9058	85.1	63.2	106457	61815	69.0
512 KB NO FAN							
Average	1174	0.7260	88.7	77.0	55898	56081	40.0
Min	1000	0.7200	56.0	53.7	45528	19941	33.0
Max	2400	0.9058	91.7	79.5	79094	58095	58.0
Average No Fan							
%Reduction	45	18	7	20	43	7	39

Light Test With Fan below or Go To Start

Light Test With Fan

Note that CPU temperature is shown to be more than 84°C for most of the time.

512 KB FAN

Seconds	MHz	Volts	CPU °C	PMIC °C	Integer MB/sec	VMSTAT Disk KB/sec	OpenGL FPS
0	2400	0.9058	42.2	39.7			
60	2146	0.9058	84.5	59.5	106457	61815	69
120	2146	0.9058	84.0	62.2		60132	68
181	2201	0.9058	84.5	62.1		61054	66
241	2366	0.9058	84.0	62.5	97930	60130	65
301	2201	0.9058	85.1	62.4		60235	67
362	2256	0.9058	84.0	62.8		60548	64
422	2146	0.9058	84.0	62.5	96799	59701	65
482	2146	0.9058	84.0	63.1		60461	67
542	2201	0.9058	85.1	62.0		60175	66
603	2146	0.7200	84.0	63.0	96761	60006	65
663	2146	0.9058	85.1	61.9		61348	64
723	2311	0.9058	84.5	62.8		59479	67
784	2146	0.9058	84.5	62.9	97231	61585	64
844	2146	0.7200	82.9	62.8		59742	64
904	2146	0.9058	82.3	62.8		60262	66
965	1500	0.9058	84.5	62.8	96604	61429	67
1025	2366	0.9058	84.0	62.9		59341	65
1086	1500	0.9058	84.0	62.3		60804	64
1146	2201	0.9058	83.4	62.8	96213	59546	65
1206	2256	0.9058	84.0	62.8		59360	64
1267	2366	0.9058	84.5	63.2		61687	68
1327	1500	0.9058	84.5	63.0	96053		64
1387	2146	0.9058	84.5	62.8		59159	66
1447	2146	0.9058	85.1	61.9		60655	65
1508	1500	0.9058	84.5	62.9	96349		67
1568	2400	0.7200	81.8	62.7		60491	66
1629	2146	0.9058	85.1	62.1		59962	64
1689	2400	0.9058	85.1	62.1	95281		63
1749	2146	0.9058	84.0	62.3		60429	61
1809	2146	0.9058	84.5	62.9		60390	64
Average	2128	0.8878	82.8	61.8	97568	60368	65.3
Min	1500	0.7200	42.2	39.7	95281	59159	61.0
Max	2400	0.9058	85.1	63.2	106457	61815	69.0

Light Test No Fan below or Go To Start

Light Test No Fan

Note that the CPU is running at 1000 MHz for much of the time, with CPU temperature around 90°C and that for the Power Management Integrated Circuit more than 78°C.

512 KB NO FAN

Seconds	MHz	Volts	CPU °C	PMIC °C	MB/sec	KB/sec	FPS
0	2400	0.9058	56.0	53.7			
60	1500	0.7200	86.2	69.5	79094	19941	58
120	1500	0.7200	85.6	72.5		58012	52
181	1500	0.7200	87.8	73.9		57754	50
241	1500	0.7200	88.9	75.8	70129	56880	50
301	1500	0.7200	89.5	76.9		57616	48
362	1500	0.7200	89.5	77.0	64348	57313	45
422	1000	0.7200	90.6	77.1		57850	44
482	1500	0.7200	88.9	77.6	57341	57980	42
543	1000	0.7200	89.5	78.2		57245	44
603	1000	0.7200	90.0	78.1		57311	41
663	1000	0.7200	90.0	78.2	53759	57391	39
724	1000	0.7200	88.9	78.6		57486	37
784	1000	0.7200	89.5	78.1		57786	38
844	1000	0.7200	90.0	78.3	50933	57456	36
905	1000	0.7200	90.0	78.5		57914	37
965	1000	0.7200	90.6	78.7		56861	38
1025	1000	0.7200	90.0	78.6	49921	57428	37
1086	1500	0.7200	89.5	78.9		57705	36
1146	1000	0.7200	90.6	78.9		57445	38
1206	1000	0.7200	90.0	78.6	48803	57803	39
1267	1000	0.7200	90.0	78.9		57618	36
1327	1000	0.7200	90.0	79.1			36
1387	1000	0.7200	90.6	78.9	47790	57545	37
1448	1000	0.7200	90.0	78.5		58095	36
1508	1000	0.7200	90.6	79.4			34
1568	1000	0.7200	90.0	79.0	47234	57055	35
1629	1000	0.7200	91.7	79.1		57110	35
1689	1000	0.7200	91.1	79.5			34
1750	1000	0.7200	91.7	79.3	45528	56708	35
1810	1000	0.7200	91.7	79.4		56874	33
Average	1174	0.7260	88.7	77.0	55898	56081	40.0
Min	1000	0.7200	56.0	53.7	45528	19941	33.0
Max	2400	0.9058	91.7	79.5	79094	58095	58.0

Heavy System Stress Test below or Go To Start

Heavy System Stress Test

This session comprised INTitHOT64g12, with 2 threads at 64 KB, MP-FPUStress64g12 with 2 threads at 512 KB, burnindrive264g12 to a PC via Ethernet, burnindrive264g12 to a USB 3 disk drive and videogl64C12 as before. Detailed important results are provided for fan and no fan scenarios, with two for the former as the first one failed. Note that, compared with 4 thread results, those for 2 threads can be slower than expected as the main data source can be from L2 cache instead of L1.

On running these tests the main issue was that the second test failed due to data comparison failures on reading. The first indication was a system warning that the disk drive was no longer available but it was remounted. Following are examples of reported errors, similar to the earlier ones described above in Disk Drive Errors and Crashes. These were thought to have been caused by the inadequate 3 amps power supply. Also, see the comments in the initial System Stress Testing summary.

```
Read passes    74 x 4 Files x  164.00 MB in    14.03 minutes
Error reading file 1

Wrong File Read szzztestz-3 instead of szzztestz1
Error reading file 2

Pass  76 file szzztestz1 word  1, data error was FFFFFFFD expected FFFFFFFB
Pass  76 file szzztestz1 word  2, data error was FFFFFFFD expected FFFFFFFB
```

A summary of the three tests sessions follow. As indicated above power consumption was higher during the tests run with the fan operational, which reduced temperatures, enabling faster performance. Without the fan, MHz throttling, involving higher temperatures, reduced current demands with slower performance. It seems that power consumption was more important than system temperature when considering stability.

	MHz	Volts	CPU °C	PMIC °C	Integer MB/sec	Floating MFLOPS	OpenGL FPS	& VMSTAT Disk MB/s	Program LAN MB/s
Best	2400				114000	32000	102	63	36
Test 9 NO FAN									
Average	1239	0.7312	88.7	77.5	38696	12361	39	Mainly	27
Min	1000	0.7200	70.8	64.7	30093	9836	31	58-59	
Max	2400	0.9118	90.6	79.4	76652	22873	51		
Test 10 FAN									
Average	2288	0.9118	81.2	60.2	71940	24046	66	Error	27
Min	2146	0.9118	42.8	40.5	64379	22518	61		
Max	2400	0.9118	84.0	61.7	78453	27388	70		
Test 11 FAN									
Average	2276	0.9080	80.8	59.7	71794	24003	66	Mainly	27
Min	1500	0.7950	41.7	38.8	59602	20594	60	57-58	
Max	2400	0.9118	84.0	61.4	82481	26551	72		
Average No Fan									
%Reductions	46	19	9	23	46	49	41	-2	0

Heavy Test No Fan below or Go To Start

Heavy Test No Fan

At 100% CPU utilisation, the following measurements were similar to those during the No Fan Light System Test, with the CPU running at 1000 MHz for much of the time, temperatures around 90°C and that for the Power Management Integrated Circuit more than 78°C.

Test 9 NO FAN Second	MHz	Volts	CPU °C	PMIC °C	Integer MB/sec	Floating MFLOPS	OpenGL FPS	VMSTAT Disk MB/s
0	2400	0.9118	70.8	64.7				
60	1500	0.7200	85.6	72.5	76652	22873	51	0.3
120	1500	0.7200	86.2	74.1	50138	15511	50	41.9
180	1500	0.7200	88.4	75.8	44886	15027	48	58.8
240	1500	0.7200	89.5	76.6	49106	15012	46	58.1
300	1500	0.7200	88.9	77.2	44702	14215	45	59.6
360	1000	0.7200	90.0	77.5	41739	12596	43	58.5
420	1500	0.7200	89.5	77.6	41734	12524	43	59.3
480	1000	0.7200	90.0	77.7	40211	12041	42	58.1
540	1000	0.7200	90.0	78.0	39083	13329	41	58.4
600	1500	0.7200	89.5	78.2	37814	12529	38	58.3
660	1500	0.7200	90.0	78.2	36144	11875	38	58.5
720	1000	0.7200	89.5	78.3	35741	11720	36	58.2
780	1000	0.7200	90.6	78.5	37614	13467	38	58.5
840	1000	0.7200	89.5	78.7	33104	10712	35	57.6
900	1000	0.7200	90.0	78.6	39563	11029	38	58.6
960	1000	0.7200	90.0	78.4	37259	11448	38	58.2
1020	1000	0.7200	89.5	78.9	34469	11583	39	57.8
1080	1000	0.7200	90.0	78.3	35970	11306	38	57.4
1140	1500	0.7200	90.0	78.7	34045	12281	36	58.6
1200	1000	0.7200	90.0	78.4	35297	10928	38	59.1
1260	1500	0.7200	90.0	78.9	37365	12002	36	58.3
1320	1000	0.7200	90.0	78.5	34004	11252	36	58.2
1380	1000	0.7200	90.0	78.4	34892	11070	34	58.8
1440	1000	0.7200	90.0	78.7	36255	10274	37	58.8
1500	1000	0.7200	88.9	78.7	33912	11320	37	58.3
1560	1500	0.7200	89.5	79.0	33513	11426	35	58.7
1620	1000	0.7200	89.5	79.0	30093	10650	35	58.8
1680	1000	0.7200	89.5	79.4	32852	9836	32	58.7
1740	1000	0.7200	90.0	79.1	30465	10273	31	122.6
1800	1500	0.8769	85.1	77.1	32262	10709	32	146.5
Average	1239	0.7312	88.7	77.5	38696	12361	39	
Min	1000	0.7200	70.8	64.7	30093	9836	31	
Max	2400	0.9118	90.6	79.4	76652	22873	51	

Heavy Test With Fan below or Go To Start

Heavy Test With Fan - FAILED

As shown initially below, system behaviour did not appear to be much different to that, at the same point, during the later successful test. However, these are instantaneous measurements that can be different in the next picosecond. Also I did note USB power measurements of 4.8 volts at 0.53 amps, compared with 4.94 and 0.53 quoted above. But this might be due to infrequent manual sampling.

Tests 10 and 11 at 900 seconds

T11	900	2366	0.9118	83.4	61.0	61490	24333	68	58.1
T10	900	2256	0.9118	83.4	61.5	70134	22929	61	59.1

Test 10 FAN Second	MHz	Volts	CPU °C	PMIC °C	Integer MB/sec	Floating MFLOPS	OpenGL FPS	VMSTAT Disk MB/s	
0	2400	0.9118	42.8	40.5					
60	2400	0.9118	79.0	55.6	70918	25009	65	9.5	
120	2201	0.9118	82.3	59.7	73729	23355	68	42.9	
180	2366	0.9118	82.9	60.9	68151	24311	67	59.5	
240	2311	0.9118	83.4	61.0	70410	23307	67	59.7	
300	2146	0.9118	82.9	61.0	73093	23714	65	58.6	
360	2311	0.9118	82.3	61.3	69355	22632	64	59.1	
420	2311	0.9118	82.9	61.5	74376	23902	62	59.1	
480	2311	0.9118	83.4	61.0	64379	23731	63	59.2	
540	2201	0.9118	82.9	61.4	72430	22757	66	58.4	
600	2201	0.9118	83.4	61.2	67268	25440	65	58.9	
660	2256	0.9118	82.9	61.7	70452	22864	66	58.2	
720	2311	0.9118	83.4	61.5	66588	22796	64	59.0	
780	2256	0.9118	82.9	61.4	71766	22518	64	59.5	
840	2146	0.9118	84.0	61.7	69162	23801	65	59.0	
900	2256	0.9118	83.4	61.5	70134	22929	61	59.1	
960	2201	0.9118	82.9	61.2	75122	24518	61	31.5	
1020	2400	0.9118	82.9	61.4	74535	23855	64	0.1	FAILED
1080	2311	0.9118	82.9	61.0	74460	23832	62	0	
1140	2256	0.9118	82.9	61.0	71397	23861	64	0	
1200	2311	0.9118	83.4	61.0	75347	23264	64	0	
1260	2311	0.9118	82.3	61.0	72384	24361	62	0	
1320	2366	0.9118	83.4	61.5	74719	25401	70	2	
1380	2400	0.9118	82.3	61.2	71234	24356	69	0	
1440	2311	0.9118	83.4	61.4	73853	24652	67	0	
1500	2366	0.9118	82.9	61.3	71402	24619	66	0	
1560	2146	0.9118	84.0	61.4	78453	23417	70	0	
1620	2256	0.9118	84.0	61.0	71631	24961	70	0	
1680	2311	0.9118	82.9	61.0	74461	25101	69	0	
1740	2201	0.9118	83.4	61.3	73486	24737	69	0	
1800	2400	0.9118	70.3	57.1	73493	27388	68	0	
Average	2288	0.9118	81.2	60.2	71940	24046	66		
Min	2146	0.9118	42.8	40.5	64379	22518	61		
Max	2400	0.9118	84.0	61.7	78453	27388	70		

Second Heavy Test With Fan below or Go To Start

Second Heavy Test With Fan

Here, performance did not vary much but there was some CPU MHz throttling. Perhaps the official fan will avoid this and overcome observed undesirable power variations with the new 5 amps version

Test 11 FAN					Integer	Floating	OpenGL	VMSTAT
Second	MHz	Volts	CPU °C	PMIC °C	MB/sec	MFLOPS	FPS	Disk MB/s
0	2400	0.9118	41.7	38.8				
60	2400	0.9118	74.7	53.7	77484	26076	67	4.5
120	2400	0.9118	81.8	58.7	82481	25011	72	42.3
180	2400	0.9118	82.9	60.0	74579	26236	66	58.3
240	2366	0.9118	81.8	60.1	69930	23368	63	57.7
300	2311	0.9118	83.4	60.5	76266	22233	68	57.9
360	2311	0.9118	83.4	60.7	72493	25286	66	58.7
420	2311	0.9118	82.3	61.0	67909	23927	70	57.9
480	2311	0.9118	83.4	60.8	73526	25794	63	57.6
540	2256	0.9118	83.4	61.0	74888	26551	67	57.9
600	2366	0.9118	82.9	61.0	74110	23912	66	57.4
660	2256	0.9118	82.9	61.1	75024	25414	65	57.6
720	2256	0.9118	82.9	61.0	59602	25025	65	59.1
780	2256	0.9118	83.4	61.0	67930	22907	65	57.1
840	2256	0.9118	84.0	61.0	71962	24011	67	58.2
900	2366	0.9118	83.4	61.0	61490	24333	68	58.1
960	2311	0.9118	82.3	61.1	63462	22888	65	58.2
1020	2256	0.9118	83.4	61.0	67540	25537	68	57.3
1080	2256	0.9118	82.9	61.0	70804	23791	66	57.8
1140	2400	0.9118	83.4	61.0	71113	22011	64	57.5
1200	2256	0.9118	82.3	61.4	77050	23111	70	58.7
1260	2311	0.9118	83.4	61.0	73053	24148	63	57.7
1320	2256	0.9118	82.3	60.9	74469	23307	66	57.6
1380	2256	0.9118	83.4	61.2	72160	22726	66	58.2
1440	2256	0.9118	82.3	60.9	73994	24276	66	59.5
1500	2256	0.9118	83.4	61.0	72659	22260	67	56.9
1560	2256	0.9118	82.9	61.2	74870	21866	68	57.8
1620	2256	0.9118	83.4	61.0	76735	23945	66	57.5
1680	2201	0.9118	83.4	60.9	70727	20594	66	57.6
1740	2311	0.9118	83.4	61.2	65023	24760	63	123.7
1800	1500	0.7950	64.2	55.4	70479	24786	60	158.3
Average	2276	0.9080	80.8	59.7	71794	24003	66	
Min	1500	0.7950	41.7	38.8	59602	20594	60	
Max	2400	0.9118	84.0	61.4	82481	26551	72	

Second below or Go To Start

Firefox, Bluetooth and YouTube

Whilst looking at numbers for this report and other things, I had movies playing via the readily accessible YouTube at 1080p HD for a few hours. YouTube was accessed via Firefox with Bluetooth sound played on a rechargeable speaker. Examples of MHz, Volts and Temperatures, with ondemand frequency scaling, were :

```
Start at Fri Aug 25 10:33:03 2023

Using 361 samples at 10 second intervals

Seconds
   0.0   ARM MHz=1500, core volt=0.9065V, CPU temp=47.2°C, pmic temp=42.3°C
  10.0   ARM MHz=2400, core volt=0.9065V, CPU temp=48.3°C, pmic temp=42.5°C
  20.1   ARM MHz=2400, core volt=0.9065V, CPU temp=48.3°C, pmic temp=42.3°C
  30.2   ARM MHz=2400, core volt=0.9065V, CPU temp=48.8°C, pmic temp=42.7°C

1028.3   ARM MHz=1500, core volt=0.9065V, CPU temp=43.9°C, pmic temp=40.7°C
1038.4   ARM MHz=2400, core volt=0.9065V, CPU temp=46.6°C, pmic temp=41.0°C
```

Pi 5 bluetooth sound levels were not loud enough for me. They were significantly louder from a side by side Pi 400. This was from Youtube movies and local music from VLC media player.

Pi 5 The Vector Processor below or Go To Start

Pi 5 The Vector Processor including whetv64SPg12 and whetv64DPg12

During the 1980s and early 90s I was responsible for evaluating and acceptance testing of supercomputers for the UK government and those centrally funded for universities. For multiple user development the latter were particularly interested in vector versus scalar performance. I converted my Fortran scalar Whetstone benchmark to one where every test function could vectorize, with a default vector length of 256 words.

The vector version was finely tuned, hands on, on Cray 1 serial 1 that was at Didcot Rutherford Laboratory for a time. First real use was during factory and site trials of the first UK full scale Cray 1. Next was the first CDC Cyber 205 and last was attending user benchmark tests in Japan for ULCC at NEC and Fujitsu, where my benchmarks were also run.

I recompiled the scalar and vector C Whetstone benchmarks on the Pi 5, using gcc 12. The scalar results were effectively the same as those from gcc 8, quoted earlier in this topic. Results for the single and double precision vector version were as follows. Note that the N5 and N8 tests, with functions (both executed at DP) mainly determine the final rating.

The gcc 12 vector benchmark was also run on the Pi 4, to compare like with like. Then, for the three main MFLOPS measurements, the Pi 5 was effectively 3.1 times faster for both single and double precision operation. For both systems, double precision MFLOPS results were effectively half those at single precision, as expected with SIMD vector operation.

```
Pi 4 GCC 12 SP
Whetstone Vector Benchmark gcc 12 64 Bit Single Precision, Sun Dec 10 17:42:10 2023

Loop content          Result          MFLOPS  MOPS Seconds

N1 floating point    -1.13316142559051  2387           0.4
N2 floating point    -1.13312149047851  2407           2.8
N3 if then else       1.00000000000000        7428     0.7
N4 fixed point       12.00000000000000        1736     9.0
N5 sin,cos etc.       0.49998238682747          79    52.2
N6 floating point     0.99999982118607  2577          10.4
N7 assignments        3.00000000000000       10223     0.9
N8 exp,sqrt etc.      0.75002217292786          78    23.7

MWIPS                                   4955         100.0

Pi 4 GCC 12 DP
Whetstone Vector Benchmark gcc 12 64 Bit Double Precision, Sun Dec 10 17:47:48 2023

Loop content          Result          MFLOPS  MOPS Seconds

N1 floating point    -1.13314558088707  1164           0.7
N2 floating point    -1.13310306766606  1173           4.9
N3 if then else       1.00000000000000        7424     0.6
N4 fixed point       12.00000000000000        1735     7.8
N5 sin,cos etc.       0.49998080312724          76    47.0
N6 floating point     0.99999988868927  1295          18.0
N7 assignments        3.00000000000000        5325     1.5
N8 exp,sqrt etc.      0.75002006515491          83    19.4

MWIPS                                   4314         100.0

Pi 5 GCC 12 SP
Whetstone Vector Benchmark gcc 12 64 Bit Single Precision, Sat Oct  7 10:46:30 2023

Loop content          Result          MFLOPS  MOPS Seconds   Pi 5/4

N1 floating point    -1.13316142559051  7393           0.3     3.10
N2 floating point    -1.13312149047851  7365           2.0     3.06
N3 if then else       1.00000000000000       14169     0.8     1.91
N4 fixed point       12.00000000000000        2399    14.5     1.38
N5 sin,cos etc.       0.49998238682747         177    51.7     2.24
N6 floating point     0.99999982118607  8079           7.4     3.13
N7 assignments        3.00000000000000       26419     0.8     2.58
N8 exp,sqrt etc.      0.75002217292786         178    23.0     2.29

MWIPS                                  10975         100.3     2.21

Pi 5 GCC 12 DP
Whetstone Vector Benchmark gcc 12 64 Bit Double Precision, Sat Oct  7 10:50:40 2023

Loop content          Result          MFLOPS  MOPS Seconds   Pi 5/4

N1 floating point    -1.13314558088707  3603           0.5     3.10
N2 floating point    -1.13310306766606  3620           3.6     3.09
N3 if then else       1.00000000000000       14168     0.7     1.91
N4 fixed point       12.00000000000000        2399    12.9     1.38
N5 sin,cos etc.       0.49998080312724         172    47.5     2.25
N6 floating point     0.99999988868927  3998          13.3     3.09
N7 assignments        3.00000000000000       13172     1.4     2.47
N8 exp,sqrt etc.      0.75002006515491         183    20.0     2.21

MWIPS                                   9830          99.9     2.28
```

Example Of Vector Instructions Compiled below or [Go To Start](#)

Example Of Vector Instructions Compiled

These are for the first single precision test function for what is probably the key part. Maximum speed of operation would be a long sequence of fused multiply and add or subtract instructions (fmla or fmls) that can produce 8 results per clock cycle for each linked vector pipeline. The disassembled code has too many non-arithmetic instructions, resulting in just over 3 operations per clock cycle on the Pi 5.

```
L11:   add     x0, x0, 16
       ldr     q4, [x0, -16]
       ldr     q0, [x0, 4816]
       ldr     q9, [x0, 9648]
       fadd    v4.4s, v0.4s, v4.4s
       ldr     q8, [x0, 14480]
       fadd    v4.4s, v4.4s, v9.4s
       fsub    v4.4s, v4.4s, v8.4s
       fmla    v0.4s, v1.4s, v4.4s
       fsub    v0.4s, v0.4s, v9.4s
       fadd    v0.4s, v0.4s, v8.4s
       fmul    v0.4s, v0.4s, v1.4s
       fneg    v2.4s, v0.4s
       mov     v5.16b, v0.16b
       mov     v3.16b, v0.16b
       fmla    v2.4s, v1.4s, v4.4s
       fmls    v5.4s, v1.4s, v4.4s
       fmla    v3.4s, v1.4s, v4.4s
       fadd    v2.4s, v2.4s, v9.4s
       mov     v4.16b, v5.16b
       fadd    v2.4s, v2.4s, v8.4s
       fmla    v4.4s, v2.4s, v1.4s
       fmla    v3.4s, v2.4s, v1.4s
       fadd    v4.4s, v4.4s, v8.4s
       fmls    v3.4s, v4.4s, v1.4s
       fmul    v3.4s, v3.4s, v1.4s
       fadd    v0.4s, v3.4s, v0.4s
       str     q3, [x0, -16]
       fmls    v0.4s, v2.4s, v1.4s
       fmla    v0.4s, v4.4s, v1.4s
       fmul    v0.4s, v0.4s, v1.4s
       fsub    v5.4s, v3.4s, v0.4s
       str     q0, [x0, 4816]
       fsub    v0.4s, v0.4s, v3.4s
       mov     v3.16b, v5.16b
       fmla    v3.4s, v2.4s, v1.4s
       mov     v2.16b, v3.16b
       fmla    v2.4s, v4.4s, v1.4s
       fmul    v2.4s, v2.4s, v1.4s
       fadd    v0.4s, v0.4s, v2.4s
       str     q2, [x0, 9648]
       fmla    v0.4s, v4.4s, v1.4s
       fmul    v0.4s, v0.4s, v1.4s
       str     q0, [x0, 14480]
       cmp     x0, x22
       bne     .L11
```

Comparison With Old Supercomputers

Following are Scalar and Vector Whetstone benchmark results for the original supercomputers. In the 1980s they provided a useful tool in confirming the choice for university work in dealing with multiple user access, typically with programs containing 90% vectorisable code. Then the choices depended on scalar versus vector performance and multiple processors versus multiple pipelines.

Pi 5 results are included and can look good on a per MHz basis. See the next page for comparisons, including for the benchmark originally used to validate performance of the first Cray 1 supercomputer.

		Scalar		Vector		Vector /Scalar	
	MHz	MWIPS	MFLOPS	MWIPS	MFLOPS	MFLOPS	DATE
Cray 1	80	16.2	5.9	98	47	8.0	1978
CDC Cyber 205	50	11.9	4.9	161	57	11.7	1981
Cray XMP1	118	30.3	11.0	313	151	13.7	1982
Cray 2/1	244	25.8	N/A	425	N/A		1984
Amdahl VP 500 #	143	21.7	7.5	250	103	13.8	1984
Amdahl VP 1100 #	143	21.7	7.5	374	146	19.5	1984
Amdahl VP 1200 #	143	21.7	7.5	581	264	35.3	1984
IBM 3090-150 VP	54	12.1	4.9	60	17	3.6	1986
(CDC) ETA 10E	95	15.7	6.5	335	124	19.2	1987
Cray YMP1	154	31.0	12.0	449	195	16.3	1987
Fujitsu VP-2400/4	312	71.7	25.4	1828	794	31.3	1991
NEC SX-3/11	345	42.9	17.0	1106	441	25.9	1991
NEC SX-3/12	345	42.9	17.0	1667	753	44.3	1991
# Fujitsu Systems							
Raspberry Pi 5 SP	2400	5843	1206	10986	7599	6.3	2023
Raspberry Pi 5 DP	2400	N/A	N/A	9816	3731	3.1	2023

PC and Pi Comparisons below or Go To Start

PC and Pi Performance Comparisons

The following results are for the original Classic Benchmarks, comprising Livermore Loops, Linpack 100 and Whetstone applications, for PCs from 1991 and the Pi 5. They tended to be produced by the latest compiler version, available at the time. These probably represent best case Pi 5 comparative performance, mainly better than the Core i5 CPU on a per MHz basis.

To be fair, the later MP-MFLOPS results, included below, reflect the other extreme via SIMD vector performance. However, my present compiling procedures might be confusing for a newbie. For the Pi 5, compiling parameters for all programs used were -O3 and -march=armv8-a for optimisation level 3 using armv8-a architecture. For Intel the method I adopted requires inclusion of compile directives for such as SSE, AVX, AVX2 or AVX512.

For those who only consider maximum performance, the Intel based PC MP-MFLOPS speeds are indicated as being far superior. But on a MFLOPS per MHz basis, the Pi 5 results were between Intel SSE and AVX measurements. Considering these and repeated runs, the Core i5 CPUs (on a laptop in this case) appear to be running at a lower MHz, using 4 threads or more.

Given an application mainly running 4 core vector MP-MFLOPS type code and a much smaller part executing the slow Whetstone scalar MFLOPS type functions, the Pi 5 can appear to be faster than that Core i5 PC. This is shown in the example (tongue in cheek) performance calculations shown below. Note the Pi 5 / Cray 1 comparisons, particularly Livermore Loops results, the benchmark originally run to validate required performance of the first Cray 1 system. Here, Gmean MFLOPS was the official average, where the Raspberry Pi 5 is indicated as being 194 times faster.

CPU	MHz	LLLOOPS MFLOPS Max	Gmean	Min	MFLOPS Linpack	MWIPS Whets	MFLOPS Whets	Device Year	LOOPS Gmean MFLOPS per MHz
Main Columns	V		V		V		V		
Cray 1	80	82.1	11.9	1.2	27	16.2	6.0	1978	0.15
Windows or Linux PCs									
AMD 80386	40	1.2	0.6	0.2	0.5	5.7	0.8	1991	0.02
80486 DX2	66	4.9	2.7	0.7	2.6	15	3.3	1992	0.04
Pentium	75	24	7.7	1.3	7.6	48	11	1994	0.10
Pentium	100	34	12	2.1	12	66	16	1994	0.12
Pentium	200	66	22	3.8		132	31	1996	0.11
AMD K6	200	68	22	2.7	23	124	26	1997	0.11
Pentium Pro	200	121	34	3.6	49	161	41	1995	0.17
Pentium II	300	177	51	5.5	48	245	61	1997	0.17
AMD K62	500	172	55	6.0	46	309	67	1999	0.11
Pentium III	450	267	77	8.3	62	368	92	1999	0.17
Pentium 4	1700	1043	187	19	382	603	146	2002	0.11
Athlon Tbird	1000	1124	201	23	373	769	161	2000	0.20
Core 2	1830	1650	413	40	998	1557	374	2007	0.23
Core i5	2300	2326	438	35	1065	1813	428	2009	0.19
Athlon 64	2150	2484	447	48	812	1720	355	2005	0.21
Phenom II	3000	3894	644	64	1413	2145	424	2009	0.21
Core i7 930	3066	2751	732	68	1765	2496	576	2010	0.24
Core i7 4820K	3900	5508	1108	88	2680	3114	716	2013	0.28
Core i5 1135G7	4150	7505	1387	92	3541	3293	802	2021	0.33
Linux PCs AVX New Compiler									
Core i7 4820K	3900	12878	2615	597	5098	5887	1174	2013	0.67
Core i5 1135G7	4150	19794	3568	943	6998	6477	1077	2021	0.86
Raspberry Pi	700	140	55	17	42	271	94	2013	0.08
Raspberry Pi 2B	900	248	115	42	120	525	244	2015	0.13
Raspberry Pi 3B	1200	436	184	56	180	725	324	2016	0.15
Raspberry Pi 4B	1500	1861	679	180	957	1883	415	2019	0.35
Raspberry Pi 4B 64b	1500	2491	730	212	1060	2269	476	2019	0.35
Raspberry Pi 5 64b	2400	10577	2308	734	4136	5843	1206	2023	0.96
Core i5 / Pi 5	1.73	1.87	1.55	1.28	1.69	1.11	0.89		0.90
Pi 5 / Cray 1	30	129	194	612	153	361	201		

##

MP-MFLOPS Threads	MHz	MFLOPS 1	2	4	8	MFLOPS/MHz 1	2	4	8
Core i7 SSE	3900	23355	46883	88776	119313	6.0	12.0	22.8	30.6
Core i7 AVX	3900	45459	91277	172443	184765	11.7	23.4	44.2	47.4
Core I5 SSE	4150	33273	64727	86194	119426	8.0	15.6	20.8	28.8
Core i5 AVX	4150	64946	128515	153955	225265	15.6	31.0	37.1	54.3
Core i5 AVX512	4150	94417	185785	324870	325915	22.8	44.8	78.3	78.5
Pi 5	2400	21519	42488	80947	85086	9.0	17.7	33.7	35.5

##

Performance Calculations

MOPS	i5 SSE MFLOPS	secs	i5 AVX MFLOPS	secs	Pi 5 MFLOPS	secs
5000	1077	4.64	1077	4.64	1206	4.15
50000	86194	0.58			80947	0.62
50000	153955			0.32		
Total		5.22		4.96		4.77

CPU Stress Tests Next or Go To Start

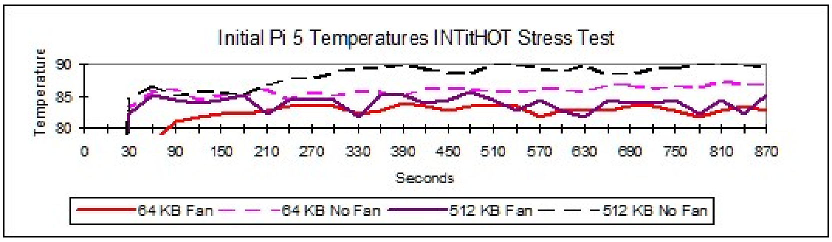
New 5 Amps Power Supply and Active Cooler

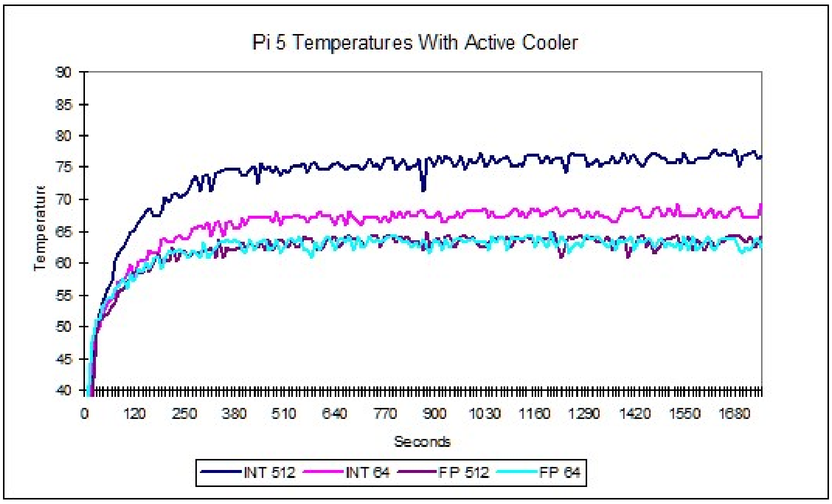
CPU Stress Tests

The fan on my new active cooler did not spin, I might have broken the JST connection on trying to insert the fiddly little thing. However, I have run some stress tests by plonking my cheap old Pi 4 fan on top of the dead new one. That and the new heatsink appear to do a good job and might be recommended as a useful backup arrangement.

Below are temperature graphs of my earlier integer and floating point tests using 64 KB and 512 KB of data. Maximum 4 thread performance was 73 GFLOPS for both floating point tests. For integers it was 240 GB/second at 64 KB then 160 GB/second at 512 KB, the latter being the hottest with data transfers reading from L2 cache as opposed to L1 at 64 KB.

The (part) active cooler graph indicates less than 80°C for all measurements, others demonstrating constant maximum CPU MHz and performance. The other graph only covers the integer tests, with and without the old Pi 4 fan. Then, using 64 KB with the fan, CPU MHz throttling was just about avoided. On running without an operational fan, it is commendable that the Pi 5 can continue running at those high temperatures, where throttled performance can be demonstrated that it is far superior to that from a super cooled Pi 4.





Heavy System Stress Test next or Go To Start

Heavy System Stress Test

This is a repeat of the above, comprising INTitHOT64g12, with 2 threads at 64 KB, MP-FPUStress64g12 with 2 threads at 512 KB, burnindrive264g12 to a PC via Ethernet, burnindrive264g12 to a USB 3 disk drive and videogl64C12. They were run with the Active Cooler enabled, initially using the new 5 amps power supply, then controlled by the 4 amps PoE arrangement. The two drive MB/second results are reading speeds, the second being for repetitive reading of the same blocks, representing bus speed where the drive has a buffer.

There were some differences in results of the two sessions at 5 amps, but nothing unusual for a mixed workload. The first test at 4 amps failed, as earlier, with disk reading errors being recorded, this time after 100 seconds. The second one at 4 amps ran successfully, essentially providing the same levels of performance as those at 5 amps. For the first 4 amps test, benchmark results, that were recorded, indicated slower performance.

There were noticeable differences in measured power where the input level was less than 5 volts, using the 4 amps supply. For some inexplicable reason, the failed test input current recording was particularly low.

An additional test was run excluding the floating point program, using the 4 amps power supply and 512 KB data size for INTitHOT via 4 threads. The latter is slower than at 64 KB but requiring a higher amperage and CPU temperature. Higher USB voltage might have helped in avoiding disk errors.

	Volts	Amps		INT MB/sec	MP MFLOPS	MHz	Volts	CPU °C	PMIC °C	OpenGL FPS	Drive MB/s	LAN MB/s
5A Supply												
Power	5.15	2.38	Min	62371	19494	2400	0.8833	37.8	40.0	59.0	52.8	35.1
USB	4.92	0.53	Avg	75234	24713	2400	0.8833	63.5	62.4	64.4	117.7	36.7
			Max	89243	28868	2400	0.8833	67.5	65.0	68.0		
Repeat			Min	63097	23625	2400	0.8833	38.4	40.1	60.0	58.5	28.6
			Avg	77075	25451	2400	0.8833	64.4	62.8	66.4	159.1	31.7
			Max	89625	27352	2400	0.8833	68.6	66.0	71.0		
4A Supply												
Power	4.88	1.98	Min	56159	18062	2400	0.7200	37.3	37.9	44.0	N/A	31.3
USB	4.71	0.54	Avg	63134	20087	2400	0.8567	51.5	49.9	56.6	N/A	N/A
FAILED			Max	69947	23773	2400	0.8840	59.8	57.2	70.0		
Repeat												
Power	4.84	2.39	Min	63472	22513	2400	0.8840	37.8	39.5	59.0	52.6	30.1
USB	4.71	0.54	Avg	76104	25127	2400	0.8840	59.4	58.4	64.7	159.0	32.2
			Max	84488	27214	2400	0.8840	62.6	60.7	70.0		
4A Supply												
Power	5.07	2.74	Min	95040		2400	0.8833	35.1	38.6	50.0	57.3	28.6
USB	4.81	0.53	Avg	100302		2400	0.8833	65.0	64.3	61.9	156.8	31.4
			Max	104684		2400	0.8833	69.2	67.2	66.0		

Solid State Hard Drive next or Go To Start

Solid State Hard Drive

I obtained another Pi 5 at the same time as the 5 amps power supply and active cooler. I had overstressed the original board creating a irrecoverable hardware failure. This occurred on plugging in a new Solid State Drive, where tests indicated power supply irregularities. It is a SanDisk 1TB Extreme Portable SSD, USB-C USB 3.2 Gen 2, External NVMe Solid State Drive up to 1050 MB/s, now with FAT32 and Ext3 partitions. I quite rightly completed all other proposed tests before returning to those for the SSD, this time with the active cooler in use.

I repeated the last heavy stress test via both the 5 amps and 4 amps power supplies. The results indicate around a 10% increase in USB current, with slightly faster operation at 4 amps but at a higher temperature. A few more runs would be required to determine the truth.

With these particular drives, SSD reading speed was around 2.45 times faster.

	Volts	Amps		INT MB/sec	MP MFLOPS	MHz	Volts	CPU °C	PMIC °C	OpenGL FPS	Drive MB/s
5A Supply SSD											
Power	5.12	2.74	Min	94755		2400	0.8838	36.7	40.2	60.0	146.7
USB	4.80	0.59	Avg	96325		2400	0.8838	64.8	64.6	64.7	166.1
			Max	109008		2400	0.8838	68.6	68.3	69.0	
4A Supply SSD											
Power	5.12	2.95	Min	109197		2400	0.8830	38.4	41.7	64.0	148.5
USB	4.84	0.59	Avg	111188		2400	0.8830	67.7	67.9	67.2	168.4
			Max	119425		2400	0.8830	71.9	71.1	70.0	

DriveSpeed and LanSpeed I/O Benchmarks

As indicated I/O above, there are two varieties of the original drive benchmark, DriveSpeed using Direct I/O and LANSpeed without that option. The former would not run via 64 bit OS software and extra large files have to be selected to avoid caching data using the latter.

First of the following results is for LanSpeed using Ext3 formatted files where one of the 4096 MB files appears to have been partially cached and not identified in vmstat sampling. Note that USB power consumption was up to 640 mA at 5.14 volts.

The second details are partial results running DriveSpeed on a FAT32 partition, where writing large files was slower than during the Ext3 test but similar on reading. The main observation is the exceptionally slow speed on handling small files, particularly on writing. Partition size was around 500 GB.

New Benchmark Large Files above indicates best USB 3 hard drive results like 30 MB/second writing and 310 MB/second reading. Results for that benchmark on the SSD were around 165 and 415 MB/second respectively.

```
   LanSpeed RasPi 64 Bit gcc 8 Tue Dec 26 12:49:03 2023

 Selected File Path: /media/raspberrypi/Ext3/ Total MB  491955, Free MB  491955

                       MBytes/Second
  MB   Write1   Write2   Write3    Read1    Read2    Read3

4096   491.86   393.63   360.86   416.77   937.70   420.40
8192   407.49   364.13   365.28   579.91   412.14   411.16

 Random         Read                       Write
 From MB           4        8       16        4        8       16
 msecs       0.002    0.002    0.002     0.52     0.49     0.48

 200 Files      Write                      Read                      Delete
 File KB           4        8       16        4        8       16      secs
 MB/sec     139.48    34.81   100.02   479.48   558.20  1353.81
 ms/file      0.03     0.24     0.16     0.01     0.01     0.01     0.019

             End of test Tue Dec 26 12:52:22 2023

procs  -----------memory---------- ---swap--  -----io---- -system-- ------cpu-----
 r  b   swpd    free   buff  cache   si   so     bi     bo   in   cs us sy id wa st
 1  3      0 6805744 182608 752752    0    0      0 413554 3775 2544  0 22 46 31  0
 2  2      0 6805744 182608 752752    0    0      0 401661 6715 8275  0 18 32 50  0
 1  3      0 6805744 182608 752752    0    0    123 382200 4824 5126  0 20 32 48  0
 1  3      0 6805744 182608 752752    0    0     13 332742 4379 4918  0 18 27 55  0
 1  3      0 6805744 182608 752752    0    0     66 363967 4509 4615  0 17 47 36  0
 2  2      0 6805744 182608 752752    0    0     46 345998 6905 9378  0 17 45 38  0
 2  0      0 6805744 182608 752752    0    0  85870 272317 4082 4434  0  4 55 41  0
 1  1      0 6805744 182608 752752    0    0 409245      0 3435  648  0  5 73 21  0
 1  1      0 6805744 182608 752752    0    0 381261      0 3076  616  0  5 74 20  0
 1  1      0 6805744 182608 752752    0    0 406957      3 3332  846  0  5 74 21  0
 2  0      0 6805744 182608 752752    0    0 414537      1 3147  597  0  5 74 21  0

   DriveSpeed RasPi 64 Bit gcc 8 Tue Dec 26 12:33:43 2023 /media/raspberrypi/FAT32/

                       MBytes/Second
  MB   Write1   Write2   Write3    Read1    Read2    Read3

1024   194.07   198.99   218.42   426.35   426.37   425.99

 200 Files      Write                      Read                      Delete
 File KB           4        8       16        4        8       16      secs
 ms/file    104.09   104.07   104.07     0.14     0.21     0.12     0.052
```

Go To Start