Application Note

Sitara™ AM62x Benchmarks

Andrew Shutzberg

ABSTRACT

This application report contains benchmarks for the AM62x family of devices.

Table of Contents

List of Tables

Trademarks

1 Introduction

These benchmarks are measured on the Cortex®-A53 cores. For updated benchmarks see Performance Guide section and the HMI and 3D Out Of Box Demo in the Processor SDK for AM62x. The benchmarks have been run on the AM62x processor, comprised of a Quad-Core 64-bit Arm®-Cortex A53 microprocessor, Single-core Arm Cortex-R5F MCU and an Arm Cortex-M4F MCU. The key parameters of the evaluation board are 1.2 GHz clock speed for the Cortex-A53 cores, and a 16-bit wide DDR4 at a speed of 1600MT/s.

2 Processor Core Benchmarks

This section contains benchmarks contained within an Arm Cortex processor core. Synthetic benchmarks included are for example Dhrystone.

2.1 Dhrystone

Dhrystone is a core only benchmark that runs from warm L1 caches in all modern processors. It scales linearly with clock speed. The score calculated by normalizing the time it takes the benchmark loop to run by the reference 1 MIPS machine score of 1757. Even though the benchmark was introduced in 1984 by Reinhold P. Weicker, Dhrystone still gets used in embedded processing. It is common to further normalize to DMIPS/MHz/core as the score scales linearly with clock speed. For standard Arm cores, the DMIPS/MHz is identical to the same compiler and flags. Dhrystone is a single core benchmark, a simple sum of multiple cores running the benchmark in parallel is sometimes used. The aggregate score for AM62 with four A53 cores at 1.2 GHz (14228 DMIPS).

Table 2-1. Dhrystone Benchmarks

	Cortex-A53 (1.2 GHz)
Dhrystones	62500000
Normalized Dhrystones (divide by 1757 reference for 1MIPS)	3557
DMIPS/MHz each core	3
Compiler and flags	GCC 9.2 -march=ARMv8 -O3
Operating System	Linux 5.10 (2021 LTS)

3 Compute and Memory System Benchmarks

This section contains benchmarks involving the Arm Cortex processor core and the memory system of the System-on-Chip (SoC). Synthetic benchmarks included are for example LMbench and CoreMark-Pro. Math function benchmarks include functions such linear algebra and fast fourier transforms (FFT).

3.1 Memory Bandwidth and Latency

STREAM and a subset of LMBench are benchmarks to measure achieved memory bandwidth and latency from software.

3.1.1 LMBench

LMBench is a suite of microbenchmarks for processor cores and operating system primitives. The memory bandwidth and latency related tests are most relevant for modern embedded processors. The results vary a little (< 10%) run to run.

LMBench benchmark *bw_mem* measures achieved memory copy performance. With parameter *cp* it does an array copy and *bcopy* parameter uses the runtime glibc version of *memcpy()* standard function. The glibc uses a highly optimized implementation that utilizes, for example, SIMD resulting in higher performance. The size parameter equal to or smaller than the cache size at a given level measures the achievable memory bandwidth from software doing a typical for loop or *memcpy()* type operation. Typical use is for external memory bandwidth calculation. The bandwidth is calculated as byte read and written counts as 1, which is roughly half of STREAM copy result. Table 3-1 shows the measured bandwidth and the efficiency compared to theoretical wire rate. The wire rate used is the DDR MT/s rate times the width divided by two (read and write making up a copy both consume the bus). The benchmark further allows creating parallel threads with -P parameter. To get the maximum multicore memory bandwidth, create the same amount of threads as there are cores available for the operating system, which is 4 for AM62x Linux (-P 4).

Table 3-1. LMBench Results

	Description	Arm Cortex-A53, DDR4-1600MT/s-16 Bit	DDR4 Efficiency
bw_mem -P 2 8M bcopy	*quad core, glibc memcpy*	1222MB/s	76%
bw_mem 8M bcopy	*single core, glibc memcpy*	887MB/s	55%
bw_mem -P 4 8M cp	*quad core, inline copy loop*	731MB/s	46%
bw_mem 8M cp	*single core, inline copy loop*	590MB/s	37%

LMBench benchmark *lat_mem_rd* is used to measure the observed memory access latency for external memory (DDR4 on AM62x) and cache hits. The two arguments are the size of the transaction (64 in the screenshot below) and the stride of the read (512). These two values are selected to measure the latency to caches and external memory, not the processor data prefetchers or other speculative execution. For access patterns, the prefetching will work, but this benchmark is most useful to measure the case when it does not. The left column is the size of the data access pattern in megabytes, right column is the round trip read latency in nanoseconds. As a summary for Arm Cortex-A53 read latency to:

- L1D is 3.5 ns
- L2 latency is 12 ns
- For access to DDR4-1600 latency is 209 ns

The below is a run with DDR4:

```
root@am62xx-evm:~# lat_mem_rd 64 512
"stride=512
0.00049 2.503
0.00098 2.504
0.00195 2.503
0.00293 2.503
0.00391 2.503
0.00586 2.503
0.00781 2.504
0.01172 2.503
0.01562 2.503
0.02344 2.520
0.03125 2.562
0.04688 7.673
0.06250 8.980
0.09375 10.190
0.12500 10.772
0.18750 11.374
0.25000 11.675
0.37500 11.969
0.50000 12.784
0.75000 140.541
1.00000 179.407
1.50000 192.142
2.00000 197.091
3.00000 202.542
4.00000 205.342
6.00000 207.528
8.00000 208.155
12.00000 209.024
16.00000 209.193
24.00000 209.510
32.00000 209.754
48.00000 209.919
64.00000 209.947
```

3.1.2 STREAM

STREAM is a microbenchmark for measuring data memory system performance without any data reuse. It is designed to miss on caches and exercise the data prefetcher and speculative accesses. It uses double precision floating point (64 bit), but in most modern processors the memory access is the bottleneck. The four individual scores are copy, scale as in multiply by constant, add two numbers, and triad for multiply accumulate. For bandwidth, a byte read counts as one and a byte written counts as one resulting in a score that is double the bandwidth LMBench. Table 3-2 shows the measured bandwidth and the efficiency compared to theoretical wire rate. The wire rate used is the DDR MT/s rate times the width. To get overall maximum achieved throughput the command used is *stream -M 16M -P 4-N 10*, which means two parallel threads and 10 iterations.

Table 3-2. Stream Benchmarks

	DDR4-1600MT/s-16-Bit Bandwidth	DDR4-1600MT/s-16-Bit Efficiency
copy	2448MB/s	77%
scale	2372MB/s	74%
add	2491MB/s	78%
triad	2493MB/s	78%

3.2 CoreMark®-Pro

CoreMark-Pro tests the entire processor, adding comprehensive support for multicore technology, a combination of integer and floating-point workloads, and data sets for utilizing larger memory subsystems. The components of CoreMark-Pro utilizes all levels of cache with an up to 3MB data memory footprint. Many, but not all of the tests, are also using pthreads to allow utilization of multiple cores. The score scales with the number of cores but is always less than linear (dual core score is less than 2x single core).

CoreMark-Pro should not be confused with the smaller CoreMark which, like Dhrystone, is a microbenchmark contained in L1 caches of a modern processor.

Table 3-3. CoreMark®-Pro Results

	Arm Cortex-A53
Single Core	795
Dual Core	1330
Quad Core	2030

3.3 Fast Fourier Transform

Fast Fourier Transform (FFT) is a multiply accumulate heavy building block in many applications. Table 3-4 shows a 1024-point single precision floating point complex FFT execution time. The Arm Cortex-A53 benchmark uses the implementation from Ne10 library, which leverages the Advanced SIMD or NEON acceleration of Cortex A53.

Table 3-4. NE10 CFFT Benchmark

	1024 pt float CFFT Execution Time (single thread / core)
Arm Cortex-A53 (1.2 GHz)	23 microseconds

3.4 Cryptographic Benchmarks

The AM62x Linux SDK includes an openssl cryptographic library that can used by applications. It is also used by some HTTPS, ssh, and netconf implementations to get access to optimized implementation of cryptographic functions. For the highest performance, the higher-level interface provided by the EVP library should be used. Table 3-5 shows a set of selected benchmarks of software observed performance run on AM62x. Command run was *openssl speed -elapsed -evp <cryptographic mode> -multi 4*. This is utilizing all four A53 cores using four threads.

Table 3-5. Symmetric Cryptography and Secure Hash in Mbit/s

	Frame Size (bytes)					
	16	**64**	**256**	**1024**	**8192**	**16384**
aes-128-gcm	2080	5944	11300	14772	16084	16128
aes-256-gcm	1956	5453	9966	12684	13668	13720
aes-128-ctr	215	481	1815	6181	21258	25584
sha256	14	58	227	868	5144	8193
sha512	14	57	203	655	1717	1932
chacha20-poly1305	1196	2550	5101	5824	6154	6173

Further benchmarks for public key cryptography are shown in Table 3-6. Tests can be run with command *openssl speed -elapsed <algorithm> -multi 4* .

Table 3-6. Public Key Cryptography Benchmarks

RSA	size	512	1024	2048	3072	4096
	sign/second	13469	2892	443	146	65
	verify/second	166015	57372	16521	7646	4382
ECDSA	curve	nistp224	nistp256	nistp521	nistk233	nistb233
	sign/second	927	4075	185	706	690
	verify/second	1468	6735	260	378	370

4 Graphics Processing Unit Benchmarks

The 3D graphics core on AM62x have specifications as shown in Table 4-1.

Table 4-1. AM62x GPU Specifications

Vendor	**Imagination Technologies**
GPU Core	AXE-1-16M
Pixel Fillrate (Mpixels/sec)	500
GFLOPS	8
3D API	OpenGL ES 3.1

4.1 Glmark2 and Kanzi

The results shown in Table 4-2 and Table 4-3 were gathered from the Glmark2 and Kanzi performace tests.

Table 4-2. Glmark2 Results

Test	**Score**
glmark2-es2-drm --off-screen	57

Table 4-3. Kanzi Results

Test	Score
Screen Resolution	1920 x 1080
Digital Cockpit	101
Car Shading	50
Fast Cluster	244

4.2 GFXBench5

The results shown in Table 4-4 were gathered from the GFXBench5 performance tests.

Table 4-4. GFXBench5 Results

Test	FPS
Manhattan Offscreen	1.53
Trex Offscreen	2.75
Egypt Offscreen	8.47

5 References

- CoreMark-Pro
- STREAM McCalpin, John D. "STREAM: Sustainable Memory Bandwidth in High Performance Computers", a continually updated technical report (1991-2007), available at: http://www.cs.virginia.edu/stream/
- Ne10 math library
- hosted models at tensorflow.org
- OpenSSL

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265