

Qutaiba Saleh, Pekka Varis and Jonathan Bishop

摘要

本应用手册介绍了一系列用于测量 AM62Ax 系列器件的各种组件性能的基准。一些标准基准包含在 Linux SDK 中，而其他基准可以从它们各自的托管网站下载。此外，还包含有关如何执行测试和如何分析测试结果的说明。

内容

1 引言	2
1.1 更改 Cortex-A53 时钟频率.....	3
2 处理器内核基准测试	3
2.1 Dhrystone.....	3
3 计算和存储系统基准测试	4
3.1 存储器带宽和延迟.....	4
3.1.1 LMBench.....	4
3.1.2 STREAM.....	7
3.1.3 临界存储器访问延迟.....	7
3.2 CoreMark-Pro.....	7
3.3 快速傅里叶变换.....	8
3.4 加密基准测试.....	9
4 应用基准测试	10
4.1 机器学习推理.....	10
5 参考文献	10

插图清单

图 1-1. AM62Ax 功能方框图.....	2
图 3-1. 存储器读取延迟.....	6

表格清单

表 2-1. Dhrystone 基准测试.....	4
表 3-1. LMBench 结果.....	4
表 3-2. 存储器读取延迟结果.....	6
表 3-3. 流基准测试.....	7
表 3-4. A53、C7x、R5F MCU 和 R5F WKUP 的临界存储器访问延迟.....	7
表 3-5. CoreMark®-Pro 结果.....	8
表 3-6. NE10 CFFT 基准测试.....	8
表 3-7. 对称加密和安全哈希 (单位为 Mbit/s).....	9
表 3-8. 公钥加密基准测试.....	9
表 4-1. 所有测试执行的平均时间.....	10

商标

Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. CoreMark® is a registered trademark of Embedded Microprocessor Benchmark Consortium. 所有商标均为其各自所有者的财产。

1 引言

AM62Ax 包含最多四个具有 64 位架构的 Arm®-Cortex®-A53 内核、一个 Cortex-R5F MCU 内核、一个 Cortex-R5F 器件管理内核、一个具有 MMA 的单核 C7x256v DSP 以及各种其他加速器（例如视频和视觉处理、具有广泛外设和网络选项的显示接口）。它支持 LPDDR4 32 位宽度，速度为 3200MT/s。图 1-1 是 AM62Ax 的功能方框图。有关详细信息，请参阅 [AM62Ax Sitara 处理器数据表](#)。

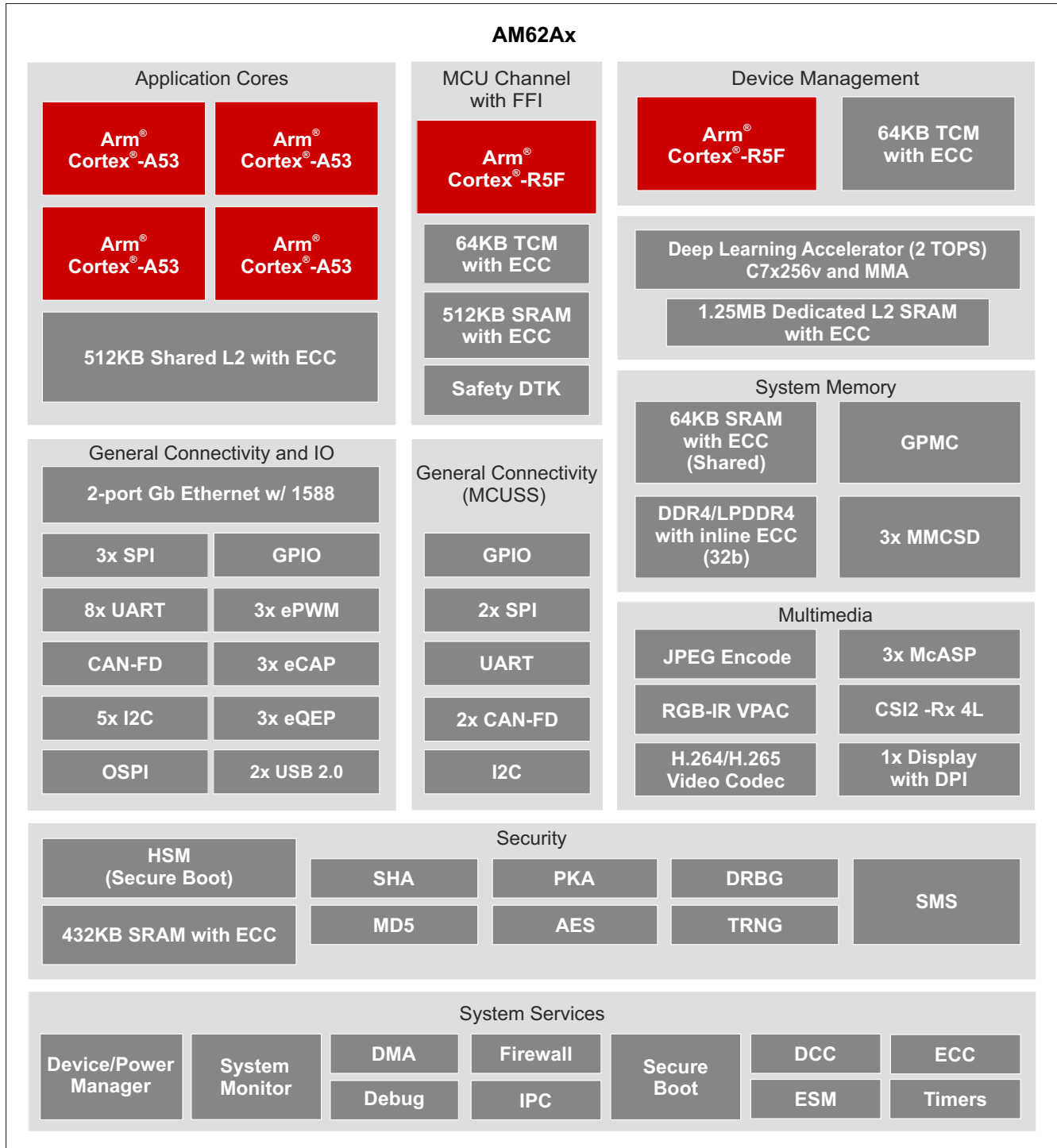


图 1-1. AM62Ax 功能方框图

本文档介绍了在 AM62Ax 处理器上测量的一系列行业标准和应用特定基准。这些测试侧重于 Arm-Cortex-A53 内核和 LPDDR4 存储器的性能，以及一些针对 Arm-Cortex-R5F MCU、C7x DSP 内核和其他存储器元件的应用特

定基准测试。评估板的关键参数包含：Cortex-A53 内核的 1.25GHz 和 1.4GHz 时钟速度，C7x DSP 的 1.0GHz 时钟速度，以及速度为 3200MT/s 的 32 位宽 LPDDR4。大多数标准基准已经包含在 SDK 中，可以直接执行，而其他基准可以从各自的官方托管网站下载。所有基准测试均使用 Linux SDK 8.06 实现。

1.1 更改 Cortex-A53 时钟频率

AM62Ax 上 A53 内核的最大频率取决于器件的速度等级和施加的 VDD_CORE 电压。AM62A SK-LP EVM Rev E2 支持 0.75V VDD_CORE 模式运行，从而支持在最大内核频率 1.25GHz 下运行。为了获得器件的峰值性能，修改了设置以在 1.4GHz 下运行 AM62Ax 上的 A53 内核。

2 处理器内核基准测试

本节包含 Arm Cortex 处理器内核基准测试。包括综合基准测试，例如 Dhrystone。

2.1 Dhrystone

Dhrystone 基准测试侧重于处理器内核性能。它在所有现代处理器中均采用热 L1 缓存运行。该测试程序随时钟速度的增加而线性增加。虽然该基准测试于 1984 年由 Reinhold P. Weicker 引入，但 Dhrystone 至今仍用于嵌入式处理。业界已采用 VAX 11/780 作为参考 1 MIPS 机器。VAX 11/780 每秒可达到 1757 Dhrystones。计算分数时，通过参考 1 MIPS 机器的分数 (1757)，对运行基准循环的时间进行标准化。由于分数随时钟速度的增加而线性增加，通常进一步标准化为 DMIPS/MHz/内核。对于标准 Arm 内核，在相同的编译器和标志中，DMIPS/MHz 将是相同的。Dhrystone 是一个单核基准测试，有时会使用多个简单内核并行运行此基准测试。

Dhrystone (版本 2.1, C 语言) 基准测试包含在 SDK 中。只需运行 *dhrystone* 命令即可执行它。由于执行时间短，建议运行大量迭代测试以测量准确的结果。在为 Arm-Cortex-A53 实施的测试中使用了 1 亿多次迭代。下面的代码块展示了用于 Dhrystone 基准执行的终端打印输出的简短版本。

```
root@am62axx-evm:~# dhrystone
Dhrystone Benchmark, Version 2.1 (Language: C)
Program compiled without 'register' attribute
Please give the number of runs through the benchmark: 100000000
Execution starts, 100000000 runs through Dhrystone
Execution ends
Final values of the variables used in the benchmark:
.
.
.
Microseconds for one run through Dhrystone:      0.1
Dhrystones per Second:                            7142857.0
```

表 2-1 展示了此基准测试的结果，其中包含编译器和操作系统详情。AM62Ax 具有四个以 1.25GHz 和 1.4GHz 运行的 A53 内核，其总分数分别为 14,880 DMIPS 和 16,261 DMIPS。

表 2-1. Dhrystone 基准测试

	Arm-Cortex-A53(1.25GHz)	Arm-Cortex-A53 (1.4GHz)
Dhrystones/s	6,535,947	7,142,857
标准化 Dhrystones (除以参考 1 MIPS 机器的分数 1757)	3,720	4,065
每个核心的 DMIPS/MHz	~3	~3
编译器和标志	GCC 9.2 -march=ARMv8 -O3	
操作系统	Linux 5.10 (2021 LTS)	

3 计算和存储系统基准测试

本节包含涉及 Arm-Cortex 处理器内核和片上系统 (SoC) 存储系统的基准测试。包括综合基准测试，例如 LMBench 和 CoreMark-Pro。数学函数基准测试包括快速傅里叶变换 (FFT)。

3.1 存储器带宽和延迟

LMBench 和 STREAM 的子集是用于测量实现的存储器带宽和软件延迟的基准测试程序。

3.1.1 LMBench

LMBench 是一套适用于处理器内核和操作系统基元的微基准测试工具。存储器带宽和延迟相关测试非常适用于现代嵌入式处理器。每次运行的结果略有不同 (<10%)。

LMBench 基准测试 `bw_mem` 测量实现的存储器复制性能。其使用参数 `cp` 复制数组，`bcopy` 参数使用运行时 `glibc` 版本的 `memcpy()` 标准函数。利用 SIMD 等实现更高性能，在实施高度优化的基础上进行 `glibc` 实践。等于或小于给定级别高速缓存大小的 `size` 参数可测量进行典型的 `for` 循环或 `memcpy()` `type` 操作的软件可实现的存储器带宽。通常用于计算外部存储器带宽。带宽根据字节读写 (每读写 1 字节计为 1) 计算，结果约为 STREAM 复制结果的一半。此基准测试还允许利用 `-P` 参数创建并行线程。为了获得较大核存储器带宽，需要创建的线程数量应等同于操作系统可用的内核数量，即 4 个 (AM62x Linux (-P 4))。为了显示 AM62Ax 的完整性能特征，LMBench 测试是在内核数量和时钟频率的完整阶乘组合上实施的。下面的代码块展示了执行 LMBench 命令的终端打印输出。

```

root@am62axx-evm:~# bw_mem 8M bcopy
8.00 2125.96
root@am62axx-evm:~# bw_mem -P 2 8M bcopy
8.00 3408.74
root@am62axx-evm:~# bw_mem -P 4 8M bcopy
8.00 3884.24

root@am62axx-evm:~# bw_mem 8M cp
8.00 1108.49
root@am62axx-evm:~# bw_mem -P 2 8M cp
8.00 1671.98
root@am62axx-evm:~# bw_mem -P 4 8M cp
8.00 1976.17
  
```

表 3-1 展示了相对于理论线速测得的带宽和效率。使用的线速计算方式为：LPDDR4 MT/s 速率 x 宽度 ÷ 2 (构成复制的读取和写入均会消耗总线)。

$$Efficiency = \frac{Measured\ Speed}{\frac{LPDDR4\ MT/s \times Width}{2}} = \frac{Measured\ Speed}{\frac{3200 \times 4\ B}{2}} = \frac{Measured\ Speed}{6400} \quad (1)$$

表 3-1. LMBench 结果

命令	说明	1.25GHz 下的 Arm-Cortex-A53、LPDDR4-3200MT/s 32 位	LPDDR4 效率	1.4GHz 下的 Arm-Cortex-A53、LPDDR4-3200MT/s 32 位	LPDDR4 效率
Bw_mem 8M bcopy	单核, <code>glibc memcpy</code>	2,058 MB/s	32%	2,125 MB/s	33%

表 3-1. LMBench 结果 (continued)

命令	说明	1.25GHz 下的 Arm-Cortex-A53、LPDDR4-3200MT/s 32 位	LPDDR4 效率	1.4GHz 下的 Arm-Cortex-A53、LPDDR4-3200MT/s 32 位	LPDDR4 效率
bw_mem -P 2 8M bcopy	双核, glibc memcpy	3,300 MB/s	52%	3,408 MB/s	53%
bw_mem -P 4 8M bcopy	四核, glibc memcpy	3,816 MB/s	60%	3,884 MB/s	61%
Bw_mem 8M cp	单核, 内联复制循环	1,076 MB/s	17%	1,108 MB/s	17%
bw_mem -P 2 8M cp	双核, 内联复制循环	1,659 MB/s	26%	1,671 MB/s	26%
bw_mem -P 4 8M cp	单核, 内联复制循环	1,952 MB/s	31%	1,976 MB/s	31%

LMBench 基准测试 `lat_mem_rd` 用于测量外部存储器 (AM62Ax LPDDR4) 观察到的存储器存取延迟和高速缓存命中率。有两个参数, 分别是事务大小 (64, 如以下代码块所示) 和读取跨度 (512)。选择这两个数值来测量高速缓存和外部存储器的延迟, 而不是处理器数据预取器或其他推测性执行的延迟。存取模式可实现预取, 但此基准测试特别适用于无法实现预取的存取模式下的相关测量。

下面的代码块展示了执行 `lat_mem_rd` 命令的终端打印输出。左列是数据存取模式的大小 (单位为兆字节), 右侧是往返读取延迟 (单位为纳秒)。此命令在 1.25GHz 和 1.4GHz 的 Arm-Cortex-A53 时钟频率下执行。

```

root@am62axx-evm:~# lat_mem_rd 64 512
"stride=512
0.00049 2.146
0.00098 2.146
0.00195 2.146
0.00293 2.146
0.00391 2.146
0.00586 2.146
0.00781 2.146
0.01172 2.146
0.01562 2.146
0.02344 2.146
0.03125 2.202
0.04688 6.725
0.06250 7.711
0.09375 8.725
0.12500 9.229
0.18750 9.748
0.25000 10.009
0.37500 22.217
0.50000 23.840
0.75000 88.270
1.00000 116.937
1.50000 133.405
2.00000 135.724
3.00000 137.268
4.00000 137.974
6.00000 138.512
8.00000 138.841
12.00000 139.033
16.00000 139.102
24.00000 139.097
32.00000 139.165
48.00000 139.163
64.00000 139.261

```

图 3-1 展示了 1.25GHz 和 1.4GHz 下存储器延迟结果的连接散点图。基于存储器块大小 (x 轴), 该图可分为三个区域。第一个区域是被存取的存储器块小于 L1 缓存时。可以假定数据完全位于 L1 内部, 因此该区域中的这种延迟是 L1 缓存延迟的近似估计值。第二个区域是被访问的存储器块大于 L1 但小于 L2 缓存时。该区域中的延迟是 L1、L2 和 LPDDR4 延迟的混合。可以假设该区域中间的延迟是 L2 延迟的近似表示。第三个区域是访问的存储器块大于 L2 缓存时。该区域中的最后一个读数反映了 LPDDR4 延迟。

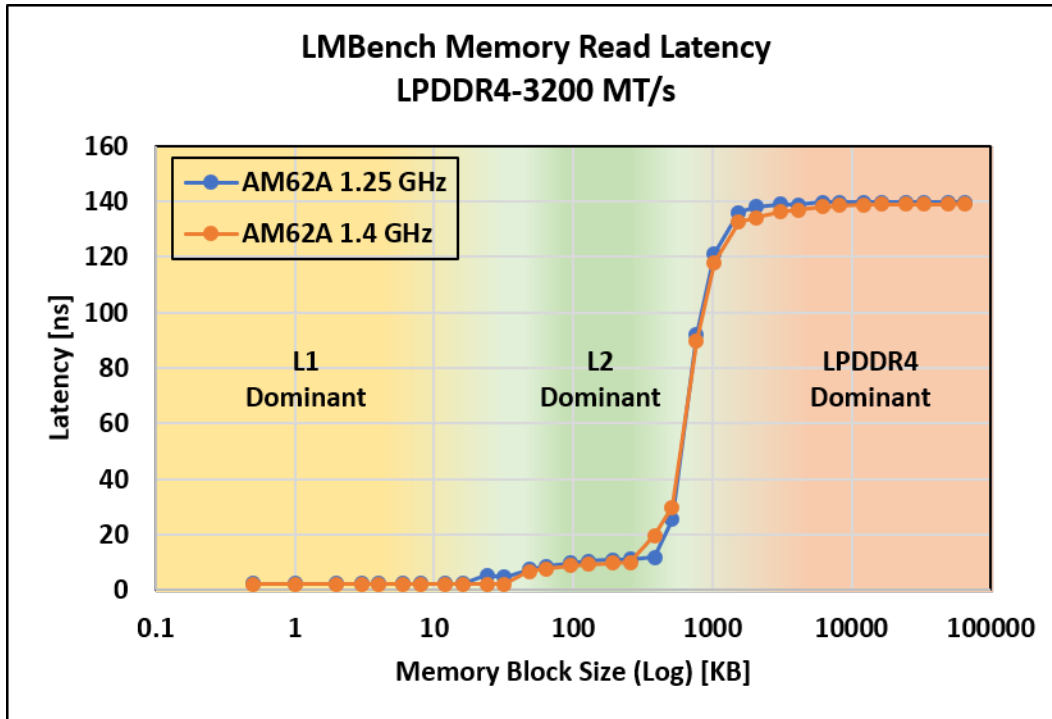


图 3-1. 存储器读取延迟

表 3-2 展示了 Arm-Cortex-A53 读取延迟的摘要。

表 3-2. 存储器读取延迟结果

存储器	1.25GHz 时的 Arm-Cortex-A53	1.4GHz 时的 Arm-Cortex-A53
L1 缓存	2.4ns	2.1ns
L2 缓存	10.3ns	9.2ns
LPDDR4-3200MT/s	140.2ns	139.2ns

3.1.2 STREAM

STREAM 是测量数据存储系统性能的微基准测试，无需任何数据重复。它旨在不命中高速缓存，执行数据预取和推测性存取。它使用双精度浮点（64 位），但在大多数现代处理器中，存储器访问是瓶颈。四个单项分数包括 copy（复制）、scale（乘常数）、add（数字相加）及 triad（乘法累加）。

- Copy：在不进行算术运算的情况下测量存储器传输速率， $a[i] = b[i]$
- Scale：包括一个简单的算术运算， $a[i] = k \times b[i]$
- 加法：除算术运算之外，还包含三个存储器存取， $a[i] = b[i] + c[i]$
- triad：将缩放和加法组合到一个运算中， $a[i] = b[i] + k \times c[i]$

对于带宽，每读取一个字节计数为 1，每写入一个字节计数为 1，得到的分数是 LMBench 带宽的两倍。表 3-3 展示了相对于理论线速测得的带宽和效率。使用的线速是 LPDDR4 MT/s 与宽度的乘积。为了获得总体最大吞吐量，使用命令 `stream -M 16M -P 4-N 10`，这意味着四个并行线程和 10 次迭代。在此测试中，Arm-Cortex-A53 时钟频率设置为 1.4GHz。

表 3-3. 流基准测试

	LPDDR4-3200MT/s-32 位带宽	LP DDR4-3200MT/s-32 位效率
复制	7,780 MB/s	61%
乘常数	7,815 MB/s	61%
add	6,868 MB/s	54%
triad	6,871 MB/s	54%

3.1.3 临界存储器访问延迟

本节提供从 AM62Ax 中的处理器到系统中的各种存储器目标的读取存储器访问延迟测量。此类测量是在 AM62Ax 平台上使用 SDK 中当前未包含的裸机芯片验证测试进行的。测试在 LPDDR4 之外的 A53、C7x 和 R5F 处理器上执行。每个测试包括一个由 8192 次迭代组成的循环，可读取总计 32KiB 的数据。每个测试的周期数被计数并除以相应的处理器时钟频率以获得延迟时间。表 3-4 展示了平均延迟结果。

表 3-4. A53、C7x、R5F MCU 和 R5F WKUP 的临界存储器访问延迟

存储器	Arm-Cortex-A53 (平均 ns)	C7x DSP (平均 ns)	Arm-Cortex-R5F MCU (平均 ns)	Arm-Cortex-R5F WKUP (平均 ns)
LPDDR4	137	154	202	172
OCSRAM MAIN	59	57	122	77
OCSRAM MCU	120	118	58	85
OCSRAM WKUP	210	189	203	156
C7X SRAM - 本地路径	不适用	20	不适用	不适用
C7X SRAM - 外部路径	80	不适用	151	103
R5F MCU TCM - 本地路径	不适用	不适用	1	不适用
R5F MCU TCM - 外部路径	143	144	不适用	120
R5F WKUP TCM - 本地路径	不适用	不适用	不适用	1
R5F WKUP TCM - 外部路径	112	108	120	不适用

测试是在 0.75V VDD_CORE 设置 (A53 : 1.25GHz, C7x DSP : 1.0GHz 和 R5 : 800MHz) 和 LPDDR4 @3200MT/s 下完成的。

3.2 CoreMark-Pro

CoreMark®-Pro 测试了整个处理器，增加了对多核技术，整数和浮点工作负载以及用于利用更大存储子系统的数据集的全面支持。CoreMark-Pro 的组件利用各种级别的高速缓存，数据存储容量高达 3MB。许多但并非所有测

试会使用 P 线程，以便允许执行多个内核。分数随内核数量的增加而增加，但总是低于线性增加（双核分数小于单核分数的 2 倍）。

请勿将 CoreMark-Pro 与更小巧的 CoreMark 混淆，后者和 Dhrystone 一样，都是包含在现代处理器 L1 缓存中的微基准测试。

CoreMark-Pro 未包含在 SDK 中。它可以从官方[托管网站](#)下载。在此测试中，直接克隆代码并将其内置在 AM62Ax EVM 中。所有正式的 CoreMark-Pro 规则都已得到满足，例如确保每个工作负载的执行时间至少是最小计时器分辨率的 1000 倍。[表 3-5](#) 展示了单核、双核和四核 A53 在 1.25GHz 和 1.4GHz 下的 CoreMark-Pro 结果。

表 3-5. CoreMark®-Pro 结果

	1.25GHz 时的 Arm-Cortex-A53 (iter/s)	并行缩放	1.4GHz 时的 Arm-Cortex-A53 (iter/s)	并行缩放
单核	837	1	965	1
双核	1,548	1.81	1,723	1.8
四核	2,465	2.9	2,694	2.82

3.3 快速傅里叶变换

快速傅里叶变换 (FFT) 是常见的信号处理算法之一。本节介绍了 Arm-Cortex-A53 和 C7x 处理器的 FFT 性能结果。[表 3-6](#) 展示了在 Arm-Cortex-A53 和 C7x DSP 上的 1024 点单精度浮点复杂 FFT 执行时间。Arm-Cortex-A53 基准测试使用了 Ne10 库，该库利用了 Cortex-A53 的高级 SIMD 或 NEON 加速。此库不包含在 SDK 中，但可以从[官方 Ne10 代码库](#)下载。在 C7x 上，SDK 中可用的 FFT 库用于对性能进行基准测试。在具有热 L2 缓存的裸机上执行的代码。

表 3-6. NE10 CFFT 基准测试

	1.25GHz 下的 Arm-Cortex-A53 (单线程/内核)	1.4GHz 下的 Arm-Cortex-A53 (单线程/内核)	1.0GHz 下的 C7x
1024 点复杂 FFT 执行时间	21.7 μ s	19.4 μ s	1.8 μ s

3.4 加密基准测试

AM62Ax Processor SDK Linux 包括一个 openssl 加密库，可提供加密运算的优化实现。某些应用（例如 HTTPS、ssh 和 netconf 实现）都采用了此库。为了获得优异的性能，应使用 EVP 库提供的较高级别的接口。表 3-7 展示了在 AM62Ax 上运行的一组选定的软件观察到的性能的部分基准测试。运行的命令是 `openssl speed -elapsed -evp <cryptographic mode> -multi 4`。这利用了全部四个 A53 内核，每个内核使用两个线程。在这些测试中，Arm-Cortex-A53 的时钟频率为 1.4GHz。openssl 命令的输出以 KB/s 为单位。为了满足所需的行业标准，表 3-7 中所报告的结果被转换为 Mb/s。

表 3-7. 对称加密和安全哈希 (单位为 Mbit/s)

	帧大小 (字节)					
	16	64	256	1024	8192	16384
aes-128-gcm	2,348	6,796	12,960	16,89	18,512	18,556
aes-256-gcm	2,227	6,242	11,372	14,538	15,703	15,634
aes-128-ctr	246	546	2,059	6,902	21,045	24,888
sha256	16	64	257	988	5,820	8,705
sha512	16	63	22	756	1,943	2,175
chacha20-poly1305	1,366	2,933	5,79	6,665	7,018	7,020

公钥加密的进一步基准测试如表 3-8 中所示。使用命令 `openssl speed -elapsed -multi 4 <algorithm>` 可运行测试。

表 3-8. 公钥加密基准测试

RSA	大小	512	1024	2048	3072	4096
	签名/秒	15,589	3,357	515	169	75
	验证/秒	192,415	66,789	19,230	8,898	5,097
ECDSA	曲线	nistp224	nistp256	nistp521	nistk233	nistb233
	签名/秒	1,056	4,457	213	801	782
	验证/秒	1,701	7,786	302	430	420

4 应用基准测试

本节包含用于机器学习推理的应用级基准测试。

4.1 机器学习推理

EdgeAI 云将提供 AM62Ax 上 2TOPS 深度学习加速器 (具有 MMA 的 C7x 256v) 上的综合机器学习性能数字 : [Edge AI \(ti.com\)](#)。

TensorFlow Lite 用于在边缘深度学习推理中测试 Arm-Cortex-A53 处理器的性能。以下示例为运行两次 TensorFlow Lite 模型, 根据 ImageNet 数据库和 1000 种物体类别进行图像分类 (224 x 224 像素, 3 字节颜色)。选择量化 Mobilenetv1 和浮点 Mobilenetv2 作为共同基准, 可用于内插计算推理应用的性能。这些模型在 SDK 中不可用。TensorFlow Lite 分类器和模型 (1.15-R5.0) 是从[官方托管网站 tensorflow.org](#) 下载的。海军少将葛丽丝·霍普 (Grace Hopper) 的示例图像安装在文件系统中 ([可点击访问](#))。示例 label_image 程序在调用 TensorFlow Lite 之前裁剪并调整 bmp 图像大小为 224 x 224 像素。下面的代码块展示了 Mobilenetv1 (mobilenet_v1_1.0_224_quant.tflite) 和 Mobilenetv2 (mobilenet_v2_1.0_224.tflite) 模型对相同图像分辨率 (224x224x3) 进行推理执行的终端打印输出。

```
root@am62axx-evm:/usr/share/tensorflow-lite/examples# ./label_image -i grace_hopper.bmp -l
labels.txt -m mobilenet_v1_1.0_224_quant.tflite
Loaded model mobilenet_v1_1.0_224_quant.tflite
resolved reporter
invoked
average time: 56.945 ms
0.780392: 653 military uniform
0.105882: 907 Windsor tie
0.0156863: 458 bow tie
0.0117647: 466 bulletproof vest
0.00784314: 835 suit

root@am62axx-evm:/usr/share/tensorflow-lite/examples# ./label_image -i grace_hopper.bmp -l
labels.txt -m mobilenet_v2_1.0_224.tflite
Loaded model mobilenet_v2_1.0_224.tflite
resolved reporter
invoked
average time: 178.05 ms
0.911345: 653 military uniform
0.014466: 835 suit
0.0062473: 440 bearskin
0.00296661: 907 Windsor tie
0.00269019: 753 racket
```

为了对 Arm-Cortex-A53 处理器进行全面性能评估, 在 1.25GHz 和 1.4GHz 下执行了基准测试。表 4-1 展示了所有测试执行的平均时间。

表 4-1. 所有测试执行的平均时间

型号	1.25GHz 时的 Arm-Cortex-A53	1.4GHz 时的 Arm-Cortex-A53
Mobilenetv1 <i>mobilenet_v1_1.0_224_quant.tflite</i>	63.35ms	56.94ms
Mobilenetv2 <i>mobilenet_v2_1.0_224.tflite</i>	192.10ms	178.05ms

5 参考文献

- [CoreMark-Pro](#)
- STREAM McCalpin, John D. “STREAM : 高性能计算机中的可持续存储器带宽”, 持续更新技术报告 (1991-2007), 可从下述网址访问 : <http://www.cs.virginia.edu/stream/>
- [Ne10 数学库](#)
- 托管模型见 [tensorflow.org](#)
- [OpenSSL](#)
- 德州仪器 (TI) : [AM62Ax Sitara 处理器数据表](#)

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司