



Qutaiba Saleh, Barath Ramesh, Jonthan Bishop and Andrew Shutzberg

摘要

本应用手册介绍了一系列用于测量 AM62Dx 系列器件的各种组件性能的基准。FREERTOS-SDK 中包括一些标准基准测试，而其他基准测试可以从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	3
1.1 将二进制文件加载到 AM62D.....	3
2 处理器内核基准测试	4
2.1 C7x DSP 基准测试.....	4
2.1.1 快速傅里叶变换.....	4
2.1.2 数字信号处理.....	5
2.1.3 数学运算.....	6
2.2 在 A53 内核中执行的 Dhrystone.....	7
3 存储器系统基准测试	8
3.1 临界存储器访问延迟.....	8
3.2 UDMA : DDR 至 DDR 数据复制.....	8
3.3 C7x DRU 性能 : 通过 DMA 进行块复制.....	9
4 应用特定的基准测试	11
4.1 SBL 引导时间.....	11
4.2 IPC 性能.....	11
4.3 闪存.....	12
4.4 应用特定延迟.....	12
5 总结	13
6 参考资料	13

插图清单

图 1-1. AM62Dx 功能方框图.....	3
图 3-1. DRU、SE 和 SA 数据移动示例.....	10

表格清单

表 2-1. C7x 上的 FFT 性能.....	4
表 2-2. 执行复数 FFT 的 C7x DSP 与 Arm CPU.....	5
表 2-3. C7x DSP 上的 FIR 滤波器性能.....	5
表 2-4. C7x DSP 上的级联双二阶滤波器性能.....	6
表 2-5. C7x DSP 上的点积性能.....	6
表 2-6. C7x DSP 上的数学运算性能.....	6
表 2-7. Dhrystone 基准测试.....	7
表 3-1. A53、C7x、R5F MCU 和 R5F WKUP 的临界存储器访问延迟.....	8
表 3-2. UDMA 通道类别.....	8
表 3-3. UDMA : DDR 至 DDR 块复制.....	9
表 3-4. DRU 性能 : 从 DDR 到 C7x 和从 Bmoack 到 DDR 的数据移动.....	10
表 4-1. SBL 设置详细信息.....	11
表 4-2. SBL 引导时间.....	11

表 4-3. IPC 消息通知延迟.....	11
表 4-4. IPC 消息传输延迟.....	11
表 4-5. EMMC 和 OSPI 性能.....	12
表 4-6. 应用特定延迟性能.....	12

商标

FreeRTOS™ is a trademark of Amazon Web Services, Inc..

Code Composer Studio™ is a trademark of Texas Instruments.

Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Linux® is a registered trademark of Linus Torvalds.

所有商标均为其各自所有者的财产。

1 引言

AM62Dx 是一个片上系统 (SOC)，包含多达四个具有 64 位架构的 Arm®-Cortex®-A53 内核和一个具有矩阵乘法加速器 (MMA) 的 C7x DSP (C7504)。DSP 包含具有 256 位定点和浮点向量的标量和向量计算单元。C7504 提供高达 40GFLOPS 的向量计算，而 MMA 提供高达 2TOPS 的计算。AM62D 还包含一个 Cortex-R5F MCU 内核、一个 Cortex-R5F 器件管理内核以及各种外设和联网选项，例如具有时间敏感网络 (TSN) 的 3 端口千兆位以太网交换机。此选项可用于启用音频网络功能，例如以太网音频视频桥接 (AVB) 和 Dante，而 McASP 等外设可启用多通道 I2S 和 TDM 音频输入和输出。AM62D 支持宽度为 32 位的 LPDDR4 存储器，运行速度为 3733MT/s。图 1-1 是 AM62Dx 的功能方框图。有关详细信息，请参阅 [AM62Dx Sitara 处理器数据表](#)。

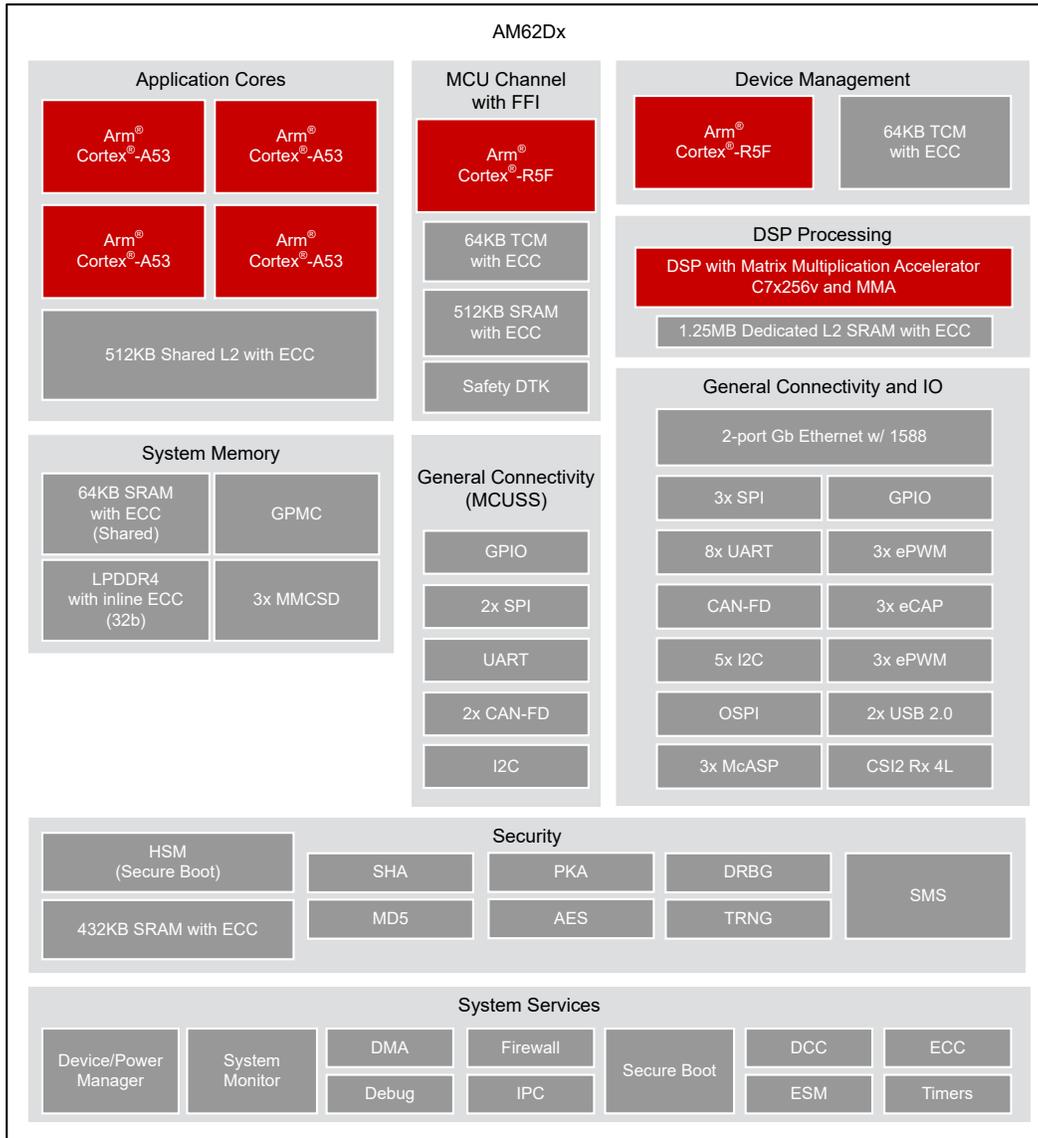


图 1-1. AM62Dx 功能方框图

本文档介绍了在 AM62Dx 处理器上测量的一系列业界通用的应用特定基准，这些测试侧重于 Arm-Cortex-A53 内核、C7x DSP 和 LPDDR4 存储器的性能。每项测试中都说明了关键的内核时钟和存储器速度参数。本文档中的测试在 FreeRTOS 或裸机上执行。这些基准测试的大多数已经包括在 AM62D 的 FreeRTOS-SDK 中。

1.1 将二进制文件加载到 AM62D

AM62D 的 FREERTOS-SDK 文档说明了对于在 AM62D EVM 上加载和运行裸机和 FreeRTOS™ 二进制文件所需的软件和工具进行设置的步骤。下面列出了步骤摘要。有关更多详细信息，请参阅 FREERTOS-SDK 文档。

1. 下载并安装 FREERTOS-SDK。
2. 下载并安装所有必需的软件和工具：
 - a. SysConfig
 - b. GCC AARCH64 编译器
 - c. C7000-CGT 编译器工具链
 - d. Python3
 - e. OpenSSL
 - f. Mono 运行时
 - g. Code Composer Studio™ (CCS)
 - h. TI CLANG 编译器工具链
3. 使用 `make -sj4 all` 构建 SDK
4. EVM 设置：
 - a. 将 EVM 设置为 UART 引导模式。
 - b. 使用 UART micro USB 端口将 EVM 连接到 PC。
 - c. 使用 UniFlash 刷写初始化二进制文件。若要从 FFTLIB、MATHLIB、DSPLIB 和 VXLIB 等库中运行 C7x 内核，请使用 SBL null。
5. 使用 CCS 将 .out 二进制文件加载到 EVM (此方法用于在 C7x DSP 上运行裸机内核)：
 - a. 为 AM62D 设置 XDS110 目标配置。仅在第一次运行时执行此步骤。
 - b. 将 EVM 更改为 OSPI NOR 引导模式。
 - c. 使用 JTAG micro USB 端口将 EVM 连接到 PC。
 - d. 启动在上一步中创建的 XDS110 配置。
 - e. 连接到 C7XSS_0。
 - f. 将 .out 二进制文件加载到 C7x 中。
 - g. 运行。
6. 对于 Dhystone 中的 FreeRTOS 示例，将 .appimage 文件刷写到器件中。

2 处理器内核基准测试

本节介绍 AM62D 中处理内核 (包括 C7x DSP 和 Arm Cortex 处理器内核) 的基准测试。包括合成基准测试，例如 Dhystone。

2.1 C7x DSP 基准测试

本节介绍了 C7x DSP 的基准测试，该 DSP 使用的内核在 DSP 算法中至关重要。使用 C7x 库 (如 FFTLIB 和 DSPLIB) 进行基准测试。这些库包含适用于大多数常用运算 (如有限脉冲滤波和点积) 的优化功能。AM62D 以源代码形式分发的 FreeRTOS-SDK 中包括这些库。每个库的文档文件夹中包括有关构建和运行这些模块的说明。

所有 C7x DSP 的性能测试都是在裸机环境下进行，并且测试时已经预先加载 L2 高速缓存。有关在 AM62D EVM 上加载和运行裸机二进制文件的信息，请参阅节 1.1。以下各小节中的基准测试根据包含这些基准测试的 C7x 库进行分类。

2.1.1 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。FFTLIB 库包含许多用于 FFT 的内核，这些内核具有各种批次大小和数据类型。仅在使用 C7x 时才实施这些内核。这些内核尚未使用 MMA。AM62D 的 FREERTOS-SDK 中包括 FFTLIB。该库的文档包含有关构建和运行这些内核的说明。表 2-1 展示了在 C7x 处理器上执行的复杂输入/复杂输出 FFT 的性能数据，数据类型包括 16 位复整数和 32 位浮点复数，且涵盖了不同的向量和批次大小。根据在 C7x 时钟频率为 1.0GHz 时执行 FFT 的 C7x 时钟周期总数以及处理每个批次的持续时间来衡量性能。

表 2-1. C7x 上的 FFT 性能

FFT 参数		16 位复整数复数输入和复数输出的性能 [周期]		32 位浮点复数输入和复数输出的性能 [周期]	
FFT 大小	批次大小：	总计 [周期数]	每批时间 [μs]	总计 [周期数]	每批时间 [μs]
128	128	12806	0.1	24221	0.18
256	64	13944	0.21	24850	0.38
512	32	13793	0.43	26826	0.83

表 2-1. C7x 上的 FFT 性能 (续)

FFT 参数		16 位整数复数输入和复数输出的性能 [周期]		32 位浮点复数输入和复数输出的性能 [周期]	
1024	16	16012	1.0	29783	1.86
2048	8	15981	1.99	31650	3.95
4096	4	17112	4.27	34834	8.70
8192	2	17063	8.53	36304	18.15

为了展示使用 C7x DSP 相比于 ARM 的性能提升，将 C7x DSP 上执行 FFT 的时间与在 Arm-Cortex-A53 内核上执行相同类型 FFT 的时间进行比较。Arm 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。该库需要一个 Linux® 操作系统，因此在 AM62A 上实施该操作系统，AM62A 具有与 AM62D 相同的 Cortex-A53 内核。[表 2-2](#) 展示了 Arm-Cortex-A53 和 C7x DSP 上的 1024 点单精度浮点复数 FFT 执行时间。结果显示，Cortex-A53 和 C7x DSP 之间的执行速度约有 11 倍的提升。

表 2-2. 执行复数 FFT 的 C7x DSP 与 Arm CPU

	Arm-Cortex-A53 (1.4GHz) 单线程/内核	1.0 GHz 时的 C7x	C7x (1.0GHz) 与 A53 (1.4GHz) 相比实现的性能提升
1024 点复数 FFT 执行时间	19.4µs	1.8µs	约 11 倍

2.1.2 数字信号处理

DSPLIB 库包含几个在 C7x DSP 上实施的低级别数字信号处理函数。AM62D 的 FreeRTOS-SDK 中包括这个库。该库的文档文件夹中包括了有关重现和运行这些模块的说明。以下各小节展示了数字信号处理领域中一些常用函数的性能结果。

2.1.2.1 FIR

有限脉冲响应 (FIR) 是音频应用中常用的滤波器之一。DSPLIB 包含 FIR 实施，可以获取各种数据和滤波器大小。[表 2-3](#) 展示了在 AM62D 的 C7x DSP 上执行 FIR 滤波器的性能结果。

表 2-3. C7x DSP 上的 FIR 滤波器性能

数据类型	数据大小	输出大小	滤波器尺寸	EVM 周期数	抽头/周期
浮点	2048	1025	1024	74483	14.09
浮点	1151	1024	128	8803	14.89
浮点	1087	1024	64	4707	13.92

2.1.2.2 级联双二阶滤波器

DSPLIB_cascadeBiquad 对输入数据实施多通道多级级联。表 2-4 展示了在浮点输入向量上实施 32 通道级联双二阶滤波器的性能结果。该表还展示了与较旧的 C66x DSP 相比实现的性能提升。

表 2-4. C7x DSP 上的级联双二阶滤波器性能

数据类型	数据大小	通道数量	级数	EVM 周期数	周期数/双二阶滤波器	C66x 到 C7x 的性能提升
浮点	512	32	3	14772	0.3	4.57x
浮点	128	32	7	8494	0.3	7.46x

2.1.2.3 点积

DSPLIB_dotprod 内核计算两个向量的点积。内核支持的数据类型包括 int8、uint8、int16、uint16、int32、uint32、浮点和双精度。表 2-5 展示了针对各种数据类型在 AM62D 的 C7x DSP 上执行点积内核所获得的性能结果。虽然内核支持任何输入向量长度，但本文中仅报告 1024 个向量的结果。DSPLIB 的用户指南中列出了其他大小的性能。

表 2-5. C7x DSP 上的点积性能

数据类型	数据大小	EVM 周期数	周期/采样
Int8	1024	131	0.13
Int16	1024	132	0.13
Int16	16384	1112	0.07
整型	1024	199	0.19
浮点	1024	229	0.22
浮点	32768	4211	0.13
双精度	1024	352	0.34

2.1.3 数学运算

MATHLIB 库包含几个用于数学运算的低级别函数，例如在 C7x DSP 上实现的三角函数、幂函数和指数函数。AM62D 的 FreeRTOS-SDK 中包括这个库。该库的文档文件夹中包括了有关重现和运行这些模块的说明。表 2-6 展示了 MATHLIB 库中一些常用数学函数的性能结果。

表 2-6. C7x DSP 上的数学运算性能

功能	数据类型	数据大小	EVM 周期数	周期/采样
正弦	浮点	1024	1161	1.133
Cos	浮点	1024	1418	1.384
Tan	浮点	1024	4752	4.640
除法	浮点	1024	353	0.344
平方根	浮点	1024	611	0.596

2.2 在 A53 内核中执行的 Dhrystone

Dhrystone 基准测试侧重于处理器内核性能，并采用所有现代处理器的预加载 L1 高速缓存来运行。该基准测试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 1MIPS 机器的分数 (1757)，对基准测试循环运行所花的时间进行归一化。由于分数随时钟速度的增加而线性增加，通常进一步归一化为 DMIPS/MHz/内核。对于标准 Arm 内核，在相同的编译器和标志中，DMIPS/MHz 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) 基准测试作为 freeRTOS 工程包括在 FREERTOS-SDK 中。该工程位于 `<freertos-sdk root>/examples/kernel/freertos/dhrystone_benchmark`。该工程在其中一个 Arm-Cortex-A53 内核中执行。由于执行时间短，TI 建议运行大量迭代测试以测量准确的结果。默认情况下，该示例会运行 3000 万次迭代测试。修改代码以运行 1 亿次迭代。结果表明，3000 万和 1 亿次迭代的结果差别不大。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。

```

Image loading done, switching to application ...
Starting RTOS/Baremetal applications

[DHRYSTONE BENCHMARKING] Iterations           : 100000000
[DHRYSTONE BENCHMARKING] Threads             : 1
[DHRYSTONE BENCHMARKING] Dhrystones per second : 7602786.5

[DHRYSTONE BENCHMARKING] Iterations           : 100000000
[DHRYSTONE BENCHMARKING] Threads             : 2
[DHRYSTONE BENCHMARKING] Dhrystones per second : 7665291.5

[DHRYSTONE BENCHMARKING] Iterations           : 100000000
[DHRYSTONE BENCHMARKING] Threads             : 5
[DHRYSTONE BENCHMARKING] Dhrystones per second : 7652734.0

[DHRYSTONE BENCHMARKING] Iterations           : 100000000
[DHRYSTONE BENCHMARKING] Threads             : 10
[DHRYSTONE BENCHMARKING] Dhrystones per second : 7665313.

All tests have passed!!
  
```

表 2-7 展示了该基准测试的结果。具有四个 A53 内核的 AM62Ax 在 1.4GHz 下运行的汇总分数为 17,308DMIPS。

表 2-7. Dhrystone 基准测试

	Arm-Cortex-A53 (1.4GHz)
Dhrystones/秒	7,602,786.5
标准化 Dhrystones (除以参考 1MIPS 机器的 1757)	4,327
每个核心的 DMIPS/MHz	大约为 3
操作系统	freeRTOS

3 存储器系统基准测试

本节包含的基准测试侧重于片上系统 (SoC) 的各种存储器系统 (包括 LPDDR4、OCSRAM 和 SRAM) 在由 Arm-Cortex A53、Arm-Cortex-R5F 和 C7x DSP 等各种处理内核访问时的性能。

3.1 临界存储器访问延迟

本节提供从 AM62Dx 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。此类测量是在 AM62Dx 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 32KB 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 3-1 展示了平均延迟结果。请注意“本地路径”和“外部访问路径”

表 3-1. 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 (C7X 访问内部存储器)	不适用	20	不适用	不适用
C7X SRAM - 外部路径 (C7X 的内部存储器, 由外部内核访问)	80	不适用	151	103
R5F MCU TCM - 本地路径 (R5F MCU 访问内部存储器)	不适用	不适用	1	不适用
R5F MCU TCM - 外部路径 (R5F MCU 内部存储器, 由外部内核访问)	143	144	不适用	120
R5F WKUP TCM - 本地路径 (R5F WKUP 访问内部存储器)	不适用	不适用	不适用	1
R5F WKUP TCM - 外部路径 (R5F WKUP 内部存储器, 由其他内核访问)	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) 条件下完成。

3.2 UDMA : DDR 至 DDR 数据复制

本节提供了使用正常容量 (NC) UDMA 通道进行 DDR 至 DDR 块复制的测试结果和观察结论。有关详细信息, 请参阅表 3-2。

表 3-2. UDMA 通道类别

	说明
正常容量 (NC)	提供了基本数量的描述符和 TR 预取以及 Tx/Rx 控制和数据缓冲。非常适用于与片上存储器和 DDR 通信的大多数外设传输。缓冲区大小为 192B 时, 此 FIFO 深度允许每个传输周期进行 3 次数据突发为 64B 的读取事务。

以下测量结果是通过使用 DDR 的 A53 上的裸机芯片验证测试收集的。传输描述符和环位于 DDR 中。测试在以下条件下完成：0.75V VDD_CORE、1.25Hz A53 内核和 3200MT/s LPDDR4。传输大小范围为 1KB 至 512KB。

NC UDMA 通道在缓冲区大小高达 512KB 时的传输容量和延迟如表 3-3 所示。

表 3-3. UDMA : DDR 至 DDR 块复制

缓冲区大小 [KB]	NC 通道带宽 [MB/s]	NC 通道延迟 [μ s]
1	108.99	8.96
2	182.88	10.68
4	262.69	14.87
8	341.90	22.85
16	403.02	38.77
32	448.35	69.7
64	472.91	132.16
128	485.30	257.57
256	491.92	508.21
512	495.12	1009.86

3.3 C7x DRU 性能：通过 DMA 进行块复制

C7x 内提供的数据路由单元 (DRU) 用于在 C7x 的 DDR 和 L2SRAM 之间传输数据，从而有效地支持 DMA。德州仪器 (TI) 信号处理 (TISP) 中间件库提供了几个示例，说明如何使用 DMA 对 C7x 的 DSPLIB 和 FFTLIB 中的各种内核进行打包。AM62D 的 FreeRTOS-SDK 中包括 TISP，并随附用于构建和运行示例的文档。TISP 中的 TISP_blockCopy 示例提供了在 C7x 的 DDR 和 L2SRAM 之间移动数据时的性能结果。在 TISP_blockCopy 示例中，我们将数据从 DDR 读取到 C7x 的 L2SRAM，同时将数据从 L2SRAM 写入 DDR。有一个块复制内核将相同的数据 (通过 DRU 读取到 L2SRAM) 从 L2SRAM 中的一个位置复制到 L2SRAM 中的另一个位置。块复制内核利用流引擎 (SE) 从 L2SRAM 读取数据。要写入通过 SE 读取的相同数据，内核会将 C7x 的写入路径应用到 L2SRAM，同时通过流地址 (SA) 生成器生成写入的地址偏移。下面列出了有关此示例的一些注释：

- DDR 规格：3200MT/s，每个事务 32 位，因此理论上的 DDR 带宽峰值为 12.8GB/s (4B x 3200MT/s)。
- DRU 传输属性设置为 4D。
- 一个通道将数据从 DDR 读取到 L2SRAM，同时另一个通道将数据从 L2SRAM 写入 DDR。图 3-1 展示了 TISP_blockCopy 示例的详细信息。在该示例中，我们通过 DMA 同时将 16MB 的数据写入到 DDR 并从 DDR 读取 16MB 的数据。总数据移动量为 32MB。
- 请注意，这里不涉及计算。

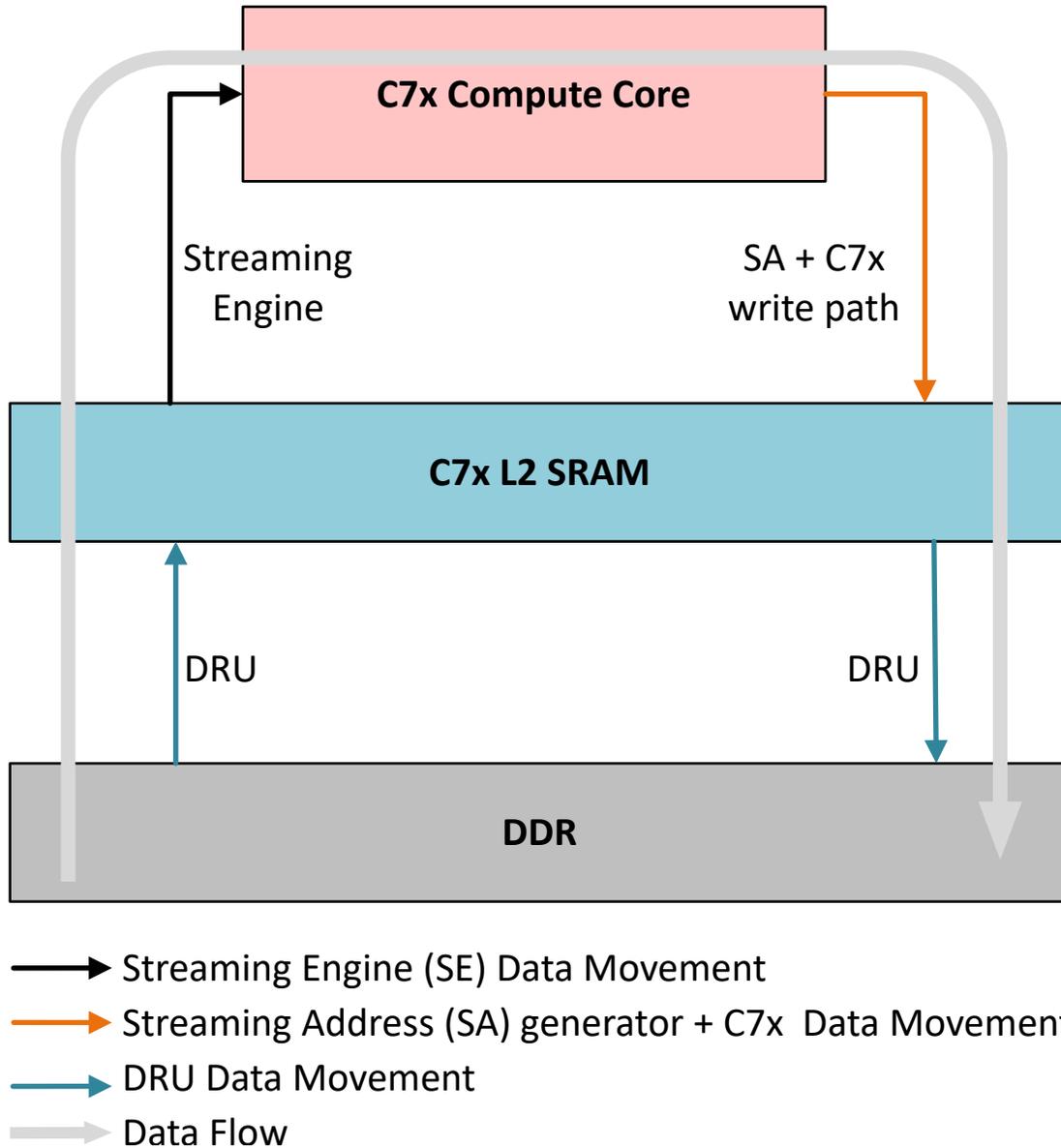


图 3-1. DRU、SE 和 SA 数据移动示例

表 3-4 展示了移动 16MB 数据的性能测量，实现 10.4GB/s 的带宽，效率为总 DDR 带宽的 81%。

表 3-4. DRU 性能：从 DDR 到 C7x 和从 Bmoack 到 DDR 的数据移动

数据类型	数据大小	EVM 周期数	数据传输	效率
浮点	2048x2048x4=16MB	3185174	5.2x2 = 10.4GB/s	10.4/12.8 = 81%

4 应用特定的基准测试

AM62D 的 FreeRTOS-SDK 为 IPC、OSPI 和 McASP 等各种硬件 IP 和软件驱动程序提供了几个额外的性能基准。FreeRTOS-SDK 文档提供了这些基准测试的结果。以下各小节中的表提供了一些基准测试的性能结果。有关最新的数据，请访问 FREERTOS-SDK 页面。

4.1 SBL 引导时间

表 4-1 展示了 SBL 设置详细信息。

表 4-1. SBL 设置详细信息

属性	详细信息
使用的软件或应用程序	sbl_ospi_multistage、ipc_rpmmsg_echo 和 HSM App 映像
由 stage1 SBL 引导的内核	r5f0-0
由 stage2 SBL 引导的内核	hsm-m4f0-0 mcu-r5f0-0 a530-0 c75ss0
由 stage1 加载的映像大小：r5f0-0	199KB
由 stage2 加载的 HSM-M4F 映像的大小	7.81KB
由 stage2 加载的 MCU-R5F 映像的大小	39.06KB
由 stage 2 加载的 A53 映像的大小	73.92KB
由 stage 2 加载的 C7x 映像的大小	144.82KB
由 stage2 加载的映像的总大小	144.82KB

在 HS-EMMC 器件上同时使用 OSPI 和 EMMC 时的 stage1 和 stage2 引导时间如表 4-1 所示。请注意，stage1 中的大部分时间都用于 DDR 初始化。

表 4-2. SBL 引导时间

阶段	166.667MHz 下的 OSPI [ms]	200.0MHz 下的 EMMC [ms]
Stage1	40.861	63.044
Stage2	33.912	52.372

4.2 IPC 性能

通过在各种处理内核之间发送 10,000 条通知并测量延迟，对 IPC Notify 性能进行基准测试。所有内核都从 DDR 运行，但来自 MSRAM 的 MCU-R5 除外，表 4-3 展示了平均通知延迟。

表 4-3. IPC 消息通知延迟

本地内核	远程内核	平均消息延迟 [ns]
mcu-r5f0-0	c75ss0	2094ns
mcu-r5f0-0	a530-0	1169ns
mcu-r5f0-0	r5f0-0	1689ns
a530-0	c75ss0	2082ns
c75ss0	r5f0-0	2065ns
a530-0	r5f0-0	1000ns

通过在处理器之间发送 1000 条消息并测量消息延迟来进行基准测试。表 4-4 展示了各种消息大小的平均和最大消息延迟。

表 4-4. IPC 消息传输延迟

本地内核	远程内核	消息大小	平均消息延迟 [μs]	最大延迟 [μs]
r5f0-0	a530-0	4	6.842	10

表 4-4. IPC 消息传输延迟 (续)

本地内核	远程内核	消息大小	平均消息延迟 [μs]	最大延迟 [μs]
r5f0-0	mcu-r5f0-0	4	8.933	12
r5f0-0	c75ss0	4	79.916	94
r5f0-0	a530-0	32	9.659	12
r5f0-0	a530-0	64	12.655	15
r5f0-0	a530-0	112	17.544	21
r5f0-0	mcu-r5f0-0	32	15.526	18
r5f0-0	mcu-r5f0-0	64	22.628	25
r5f0-0	mcu-r5f0-0	112	33.379	36
r5f0-0	c75ss0	32	89.926	104
r5f0-0	c75ss0	64	92.618	127
r5f0-0	c75ss0	112	113.281	128

4.3 闪存

表 4-5 展示了 EMMC 和 OSPI 针对各种数据大小的写入和读取速度。EMMC 设置为 HS200 模式，而 OSPI 使用 FLASH_CFG_PROTO_8D_8D_8D 协议且启用了 PHY 和 DMA。

表 4-5. EMMC 和 OSPI 性能

模式	数据大小 [MB]	写入速度 [Mbps]	读取速度 [Mbps]
EMMC	1	53.48	157.51
EMMC	4	109.84	169.34
EMMC	6	108.03	169.78
OSPI	1	0.43	283.77
OSPI	5	0.43	248.83
OSPI	10	0.43	284.96

4.4 应用特定延迟

表 4-6 展示了音频应用中各种关键功能的延迟。

表 4-6. 应用特定延迟性能

测试	详细信息	延迟
McASP (音频)	以 48kHz 运行, I2C 模式 RX 至 TX 引脚对引脚	792μs
引导延迟 (MCU R5)	从 MCU_PORz 测得 MCUR5 主控制器的 GPIO 切换时间	112ms
C7x 音频提示声时间	从 MCU_PORz 测得 包括 DAC 配置	185ms

5 总结

本文档介绍了 AM62D 处理器的一组基准测试。这些测试侧重于处理内核和存储器性能，并重点关注 C7x DSP。本文档还包括用于重现结果的步骤。

6 参考资料

- 德州仪器 (TI), [AM62Dx Sitara 处理器](#) 数据表
- [Ne10 数学库](#) 网页

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司