

## TMS320C6678 存储器访问性能

冯华亮

多核处理器

### 摘要

TMS320C6678 有 8 个 C66x 核，典型速度是 1GHz，每个核有 32KB L1D SRAM，32KB L1P SRAM 和 512KB LL2 SRAM；所有 DSP 核共享 4MB SL2 SRAM。一个 64-bit 1333MTS DDR3 SDRAM 接口可以支持 8GB 外部扩展存储器。

存储器访问性能对 DSP 上运行的软件是非常关键的。在 C6678 DSP 上，所有的主模块，包括多个 DSP 核和多个 DMA 都可以访问所有的存储器。

每个 DSP 核每个时钟周期都可以执行最多 128 bits 的 load 或 store 操作。在 1GHz 的时钟频率下，DSP 核访问 L1D SRAM 的带宽可以达到 16GB/S。

DSP 的内部总线交换网络，TeraNet，提供了 C66x 核(包括其本地存储器)，外部存储器，EDMA 控制器，和片上外设之间的互连总共有 10 个 EDMA 传输控制器可以被配置起来同时执行任意存储器之间的数据传输。

本文为设计人员提供存储器访问性能评估的基本信息；提供各种操作条件下的性能测试数据；还探讨了影响存储器访问性能的一些因素。

## 目录

1. 存储器系统简介.....	4
2. DSP 核, EDMA3, IDMA 拷贝数据的性能比较.....	6
3. DSP 核访问存储器的时延.....	9
3.1 DSP 核访问 LL2 的时延.....	9
3.2 DSP 核访问 SL2 的时延.....	11
3.3 DSP 核访问外部 DDR 存储器的时延.....	13
4. DMA 访问存储器的性能.....	17
4.1 DMA 传输的额外开销.....	17
4.2 EDMA 10 个传输引擎的区别.....	18
4.3 EDMA 带宽和传输灵活性的权衡.....	19
4.3.1 第一维大小 (ACNT) 的考虑 (突发数据块大小).....	19
4.3.2 二维传输的考虑 (传输优化).....	20
4.3.3 地址偏移的考虑.....	20
4.3.4 地址对齐.....	21
5. 多个主模块共享存储器的性能.....	22
5.1 多个主模块共享 SL2 的性能.....	22
5.2 多个主模块共享 DDR 的性能.....	25
5.2.1 多个 DSP 核共享 DDR 的性能.....	27
5.2.2 多个 EDMA 共享 DDR 的性能.....	30
6. 总结.....	33
参考文献.....	33

## 图

图 1 TMS320C6678 存储器系统.....	5
图 2 DSP 核访问 LL2.....	10
图 3 C66x 和 C64x+核在 LL2 上 Load/Store 的时延比较.....	11
图 4 DSP 核访问 SL2.....	12
图 5 DSP 核访问 SL2 和 LL2 的性能比较.....	13
图 6 DSP 核对 DDR Load/Store 的时延.....	16
图 7 ACNT 对 EDMA 效率的影响.....	19
图 8 线性 2D 传输.....	20
图 9 偏移量对 EDMA 带宽的影响.....	21
图 10 SL2 bank 组织结构.....	22
图 11 DDR bank 数据组织结构.....	25
图 12 多个主模块访问相同 DDR bank 的不同页.....	26
图 13 多个主模块访问不同 DDR bank 的不同页.....	27

## 表

表 1 1GHz C6678 上 C66x 核, IDMA 和 EDMA 的理论带宽.....	6
表 2 1GHz C6678 上各种存储器端口的理论带宽.....	6

表 3 DSP 核, EDMA 和 IDMA 数据拷贝的吞吐量比较 .....	7
表 4 EDMA CC0 传输的额外开销 .....	17
表 5 EDMA CC1 和 EDMA CC2 传输的额外开销 .....	17
表 6 EDMA 传输引擎之间的区别 .....	18
表 7 1GHz C6678 上 EDMA TC 吞吐量比较.....	18
表 8 多个 DSP 核共享 SL2 的性能.....	23
表 9 多个 EDMA 共享 SL2 的性能 .....	24
表 10 多个 DSP 核共享 DDR 的性能.....	27
表 11 DDR 优先级提升计数值的影响.....	29
表 12 多个 EDMA 共享 DDR 的性能.....	30
表 13 多个主模块访问相同 bank 的概率.....	33

## 1. 存储器系统简介

TMS320C6678 有 8 个 C66x 核，每个核有：

- ◆ 32KB L1D (Level 1 Data) SRAM，它和 DSP 核运行在相同的速度上，可以被用作普通的数据存储器或数据 cache。
- ◆ 32KB L1P (Level 1 Program) SRAM，它和 DSP 核运行在相同的速度上，可以被用作普通的程序存储器或程序 cache。
- ◆ 512KB LL2 (Local Level 2) SRAM，它的运行速度是 DSP 核的一半，可以被用作普通存储器或 cache，既可以存放数据也可以存放程序。

所有 DSP 核共享 4MB SL2 (Shared Level 2) SRAM，它的运行速度是 DSP 核的一半，既可以存放数据也可以存放程序。

TMS320C6678 集成一个 64-bit 1333MTS DDR3 SDRAM 接口，可以支持 8GB 外部扩展存储器，既可以存放数据也可以存放程序。它的总线宽度也可以被配置成 32 bits 或 16 bits。

存储器访问性能对 DSP 上软件运行的效率是非常关键的。在 C6678 DSP 上，所有的主模块，包括多个 DSP 核和多个 DMA 都可以访问所有的存储器。

每个 DSP 核每个时钟周期都可以执行最多 128 bits 的 load 或 store 操作。在 1GHz 的时钟频率下，DSP 核访问 L1D SRAM 的带宽可以达到 16GB/S。当访问二级(L2)存储器或外部存储器时，访问性能主要取决于访问的方式和 cache。

每个 DSP 核有一个内部 DMA (IDMA)，在 1GHz 的时钟频率下，它能支持高达 8GB/秒的传输。但 IDMA 只能访问 L1 和 LL2 以及配置寄存器，它不能访问外部存储器。

DSP 的内部总线交换网络，TeraNet，提供了 C66x 核(包括其本地存储器)，外部存储器，EDMA 控制器，和片上外设之间的互联。总共有 10 个 EDMA 传输控制器可以被配置起来同时执行任意存储器之间的数据传输。芯片内部有两个主要的 TeraNet 模块，一个用 128 bit 总线连接每个端点，速度是 DSP 核频率的 1/3，理论上，在 1GHz 的器件上每个端口支持 5.333GB/秒的带宽；另一个 TeraNet 内部总线交换网络用 256 bit 总线连接每个端点，速度是 DSP 核频率的 1/2，理论上，在 1GHz 的器件上每个端口支持 16GB/秒的带宽。

总共有 10 个 EDMA 传输控制器可以被配置起来同时执行任意存储器之间的数据传输。它们中的两个连接到 256-bit, 1/2 DSP 核速度的 TeraNet 内部总线交换网络；另外 8 个连接到 128-bit, 1/3 DSP 核速度的 TeraNet 内部总线交换网络。

图 1 展示了 TMS320C6678 的存储器系统。总线上的数字代表它的宽度。大部分模块运行速度是 DSP 核时钟的 1/n，DDR 的典型速度是 1333MTS(Million Transfer per Second)。

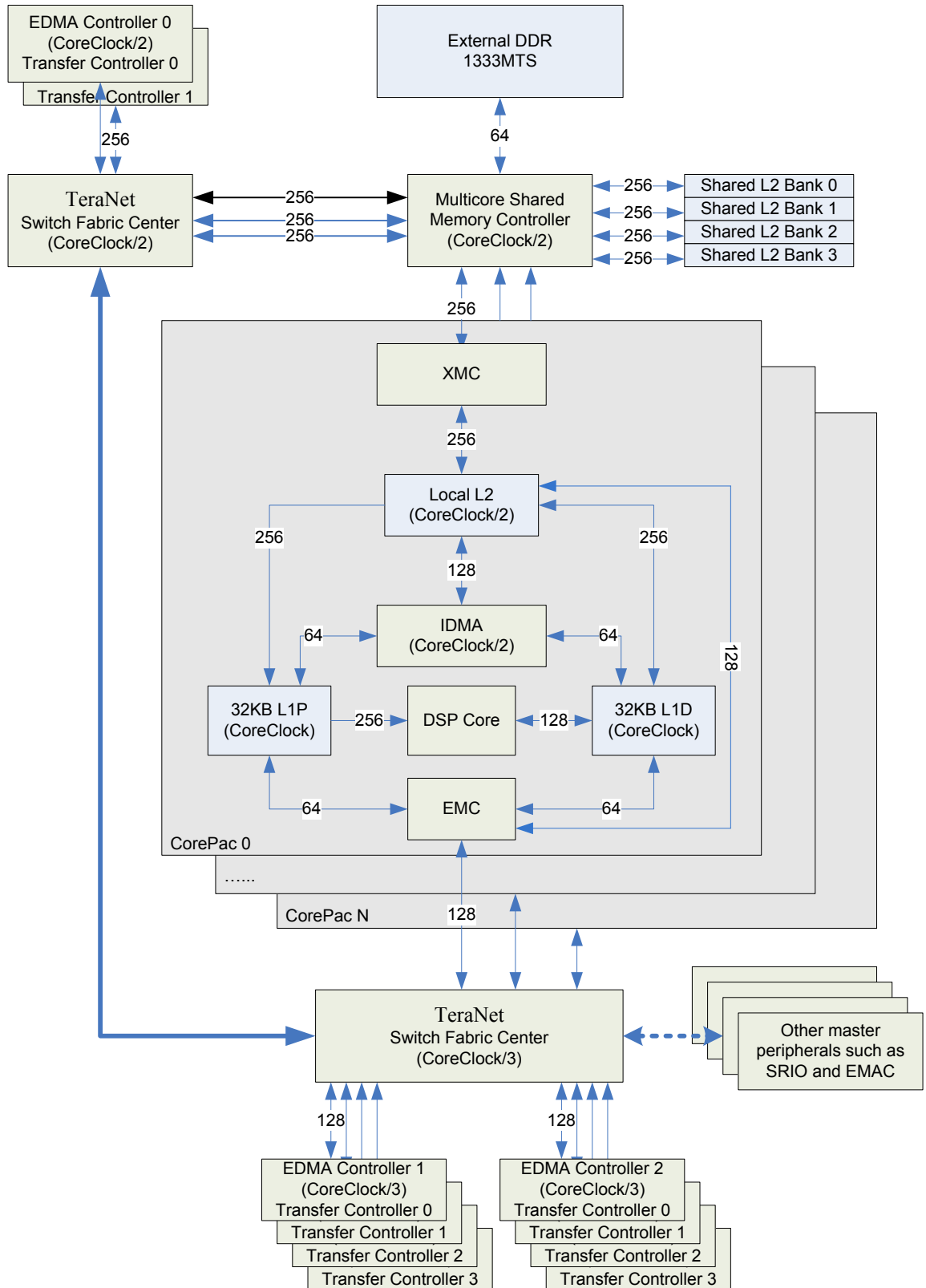


图 1 TMS320C6678 存储器系统

本文为设计人员提供存储器访问性能评估的基本信息；提供各种操作条件下的性能测试数据；还探讨了影响存储器访问性能的一些因素。

本文对分析以下常见问题会有所帮助：

1. 应该用 DSP 核还是 DMA 来拷贝数据？
2. 一个频繁访问存储器的函数会消耗多少时钟周期？
3. 当多个主模块共享存储器时，对某个模块的性能会有多大的影响？

本文中的大部分数据是在 C6678 EVM (EValuation Module)板上测试得到的，它上面有 64-bit 1333MTS 的 DDR 存储器。

## 2. DSP 核，EDMA3，IDMA 拷贝数据的性能比较

数据拷贝的带宽由下面三个因素中最差的一个决定：

1. 总线带宽
2. 源端吞吐量
3. 目的端吞吐量

表 1 总结了 C6678 上 C66x 核，IDMA 和 EDMA 的理论带宽。

**表 1 1GHz C6678 上 C66x 核，IDMA 和 EDMA 的理论带宽**

主模块	理论带宽 MB/s	说明
C66x 核	16, 000	$(128 \text{ bits}) / (8 \text{ bit/byte}) * 1000\text{M} = 16000\text{MB/s}$
IDMA	8, 000	$(64 \text{ bits}) / (8 \text{ bit/byte}) * 1000\text{M} = 8000\text{MB/s}$
EDMA CC0 (TC0, 1)	16, 000	$(256 \text{ bits}) / (8 \text{ bit/byte}) * (1000\text{M}/2) = 16000\text{MB/s}$
EDMA CC1, 2 (TC0~3)	5, 333	$(128 \text{ bits}) / (8 \text{ bit/byte}) * (1000\text{M}/3) = 5333\text{MB/s}$

表 2 总结了 C6678 EVM (64-bit 1333MTS DDR) 上各种存储器端口的理论带宽。

**表 2 1GHz C6678 上各种存储器端口的理论带宽**

存储器	理论带宽 MB/s	说明
-----	-----------	----

L1D	32, 000	$(256 \text{ bits}) / (8 \text{ bit/byte}) * (1000\text{M}) = 32000\text{MB/s}$
L1P	32, 000	$(256 \text{ bits}) / (8 \text{ bit/byte}) * (1000\text{M}) = 32000\text{MB/s}$
LL2	16, 000	$(256 \text{ bits}) / (8 \text{ bit/byte}) * (1000\text{M}/2) = 16000\text{MB/s}$
SL2	64, 000	$(4*256 \text{ bits}) / (8 \text{ bit/byte}) * (1000\text{M}/2) = 64000\text{MB/s}$
DDR	10, 666	$(64 \text{ bits}) / (8 \text{ bit/byte}) * (1333\text{MTS}) = 10666\text{MB/s}$

表 3 列出了在 1GHz C6678 EVM（64-bit 1333MTS DDR）上，在不同情况下用 EDMA，IDMA 和 DSP 核做大块连续数据拷贝测得的吞吐量。

在这些测试中，L1 上的测试数据块的大小是 8KB；IDMA LL2->LL2 拷贝的数据块的大小是 32KB；其它 DSP 核拷贝测试的数据块的大小是 64KB，其它 EDMA 拷贝测试的数据块大小是 128KB。

吞吐量由拷贝的数据量除以消耗的时间得到。

**表 3 DSP 核，EDMA 和 IDMA 数据拷贝的吞吐量比较**

吞吐量(MB/s) Src-> Dst	1GHz DSP		
	DSP核	EDMA	IDMA
LL2 -> LL2 (32KB L1D cache)	2557	4939	2356
LL2-> L1D (16KB L1D cache)	3927	4303	3343
L1D-> LL2 (16KB L1D cache)	7713	4303	4156
LL2-> SL2 (32KB L1D cache, prefetchable 缺省SL2空间)	3756	5266	N/A
LL2-> SL2 (noncacheable, nonprefetchable重映射的SL2空间)	2264		
LL2-> SL2 (32KB L1D cache, prefetchable重映射的SL2空间)	3362		
SL2-> LL2 (32KB L1D cache, prefetchable 缺省SL2空间)	3270	5266	N/A
SL2-> LL2 (noncacheable, nonprefetchable重映射的SL2空间)	591		
SL2-> LL2 (32KB L1D cache, prefetchable重映射的SL2空间)	2606		
SL2-> SL2 (32KB L1D cache, prefetchable 缺省SL2空间)	2660	7878	N/A
SL2-> SL2 (noncacheable, nonprefetchable重映射的SL2空间)	484		
SL2-> SL2 (32KB L1D cache, prefetchable重映射的SL2空间)	2454		

LL2-> 另一个核的LL2 (non-cacheable, nonprefetchable)	1642	5253	N/A
LL2-> 另一个核的LL2 (32KB L1D cache, prefetchable)	2675		
LL2-> 另一个核的LL2 (32KB L1D, 256KB L2 cache, prefetchable)	1842		
另一个核的LL2-> LL2 (non-cacheable, nonprefetchable)	183		
另一个核的LL2-> LL2 (32KB L1D cache, prefetchable)	1182		
另一个核的LL2-> LL2 (32KB L1D, 256KB L2 cache, prefetchable)	1838		
LL2 -> 64-bit DDR (non-cacheable, nonprefetchable)	1336	5267	N/A
LL2 -> 64-bit DDR (32KB L1D cache, prefetchable)	2677		
LL2 -> 64-bit DDR (32KB L1D, 256KB L2 cache, prefetchable)	2109		
64-bit DDR -> LL2 (non-cacheable, nonprefetchable)	188	5253	N/A
64-bit DDR -> LL2 (32KB L1D cache, prefetchable)	1321		
64-bit DDR -> LL2 (32KB L1D, 256KB L2 cache, prefetchable)	2025		
64-bit DDR -> 64-bit DDR (non-cacheable, nonprefetchable)	150	3878	N/A
64-bit DDR -> 64-bit DDR (32KB L1D cache, prefetchable)	976		
64-bit DDR -> 64-bit DDR (32KB L1D, 256KB L2 cache, prefetchable)	1847		

总的来说，DSP 核可以高效地访问内部存储器，而用 DSP 核访问外部存储器则不是有效利用资源的方式；IDMA 非常适用于 DSP 核本地存储器(L1D, L1P, LL2)内连续数据块的传输，但它不能访问共享存储器(SL2, DDR)；而外部存储器的访问则应尽量使用 EDMA。

Cache 配置显著地影响 DSP 核的访问性能，Prefetch buffer 也能提高读访问的效率，但它们不影响 EDMA 和 IDMA。这里所有 DSP 核的测试都是基于 cold cache（cache 和 Prefetch buffer 在测试前被清空）。

对 DSP 核，SL2 可以通过从 0x0C000000 开始的缺省地址空间被访问，通常这个地址空间被设置为 cacheable 而且 prefetchable。SL2 可以通过 XMC(eXtended Memory Controller)被重映射到其它存储器空间，通常重映射空间被用作 non-cacheable, nonprefetchable 访问（当然它也可以被设置为 cacheable 而且 prefetchable）。通过缺省地址空间访问比通过重映射空间访问稍微快一点。

前面列出的 EDMA 吞吐量数据是在 EDMA CC0 (Channel Controller 0) TC0 (Transfer Controller 0)上测得的，EDMA CC1 和 EDMA CC2 的吞吐量比 EDMA CC0 低一些，后面有专门的章节来比较 10 个 EDMA 传输控制器的差别。



### 3. DSP 核访问存储器的时延

L1 和 DSP 核的速度相同，所以 DSP 核每个时钟周期可以访问 L1 存储器一次。对一些特殊应用，需要非常快的访问小块数据，可以把 L1 的一部分配置成普通 RAM（而不是 cache）来存放数据。

通常，L1 被全部配置成 cache，如果 cache 访问命中(hit)，DSP 核可在一个周期完成访问；如果 cache 访问没有命中(miss)，DSP 核需要等待数据从下一级存储器中被读到 cache 中。

本节讨论 DSP 核访问内部存储器和外部 DDR 存储器的时延。下面是时延测试的伪代码：

```
flushCache();
startCycle= getTimeStampCount();
for(i=0; i< AccessTimes; i++)
{
    Access memory at address;
    address+= stride;
}
totalCycles = getTimeStampCount()-startCycle;
Cycles/Access= totalCycles /AccessTimes;
```

#### 3.1 DSP 核访问 LL2 的时延

图 2 是在 1GHz C6678 EVM 上测得的 DSP 核访问 LL2 的时延。DSP 核执行 512 个连续的 LDDW (Load Double Word)或 STDW (Store Double Word)指令所花的时间被测量，平均下来每个操作所花的时间被画在图中。这个测试使用了 32KB L1D cache。

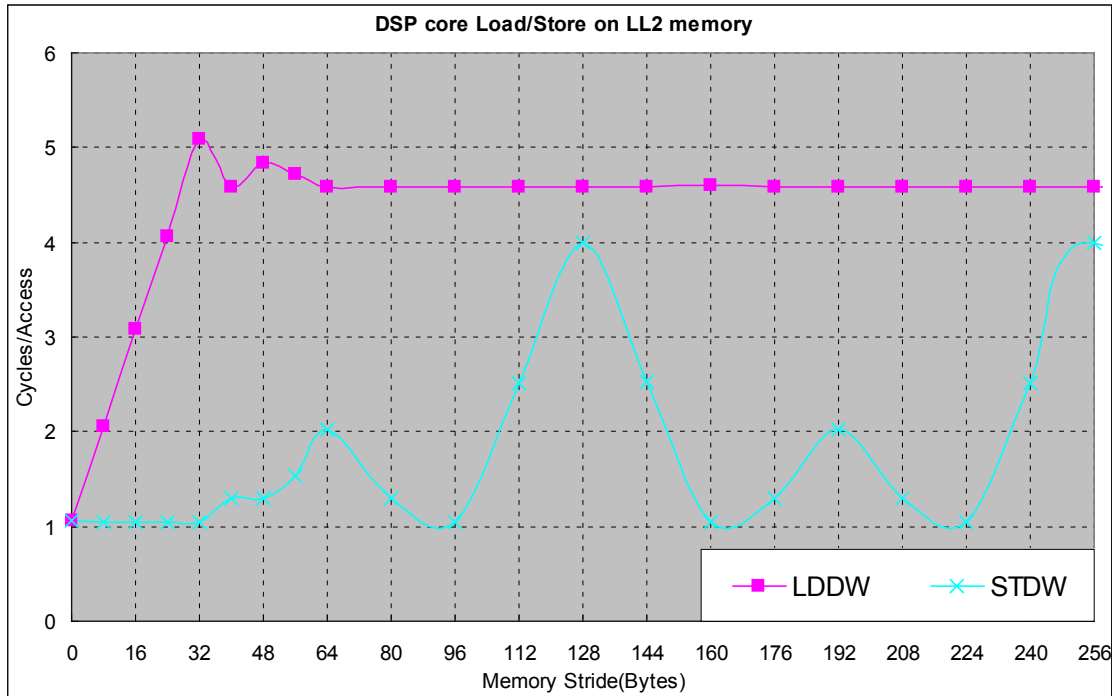


图 2 DSP 核访问 LL2

对 LDB/STB 和 LDW/STW 的测试表明，它们的时延与 LDDW/STDW 相同。

由于 L1D cache 只有在读操作时才会被分配，DSP 核读 LL2 总是通过 L1D cache。所以，DSP 核访问 LL2 的性能高度依赖 cache。多个访问之间的地址偏移(stride)显著地影响访问效率，地址连续的访问可以充分地利用 cache；大于或等于 64 字节的地址偏移导致每次访问都 miss L1 cache 因为 L1D cache 行大小是 64 bytes。

由于 L1D cache 不会在写操作时被分配，并且这里的测试之前 cache 都被清空了，所以任何对 LL2 的写操作都通过 L1D write buffer (4x16bytes)。对多个写操作，如果地址偏移小于 16 bytes，这些操作可能在 write buffer 中被合并成一个对 LL2 的写操作，从而获得接近平均每个写操作作用 1 个时钟周期的效率。

当多个写操作之间的偏移是 128 bytes 整数倍时，每个写操作都访问 LL2 的相同 sub-bank (LL2 包含两个 banks，每个 bank 包含 4 个总线宽度为 16-byte 的 sub-bank)，对相同 sub-bank 的连续访问的时延是 4 个时钟周期。对其它的访问偏移量，连续的写操作会访问 LL2 不同的 bank，这样的多个访问的在流水线上可以被重叠起来，从而使平均的访问时延比较小。

C66x 核在 C64x+核的基础上有很多改进，C66x 核的 L2 存储器控制器和 DSP 核速度相同，而 C64x+的 L2 存储器控制器的运行速度是 DSP 核速度的 1/2。图 3 比较了 C66x 和 C64x+ Load/Store LL2 存储器的性能。

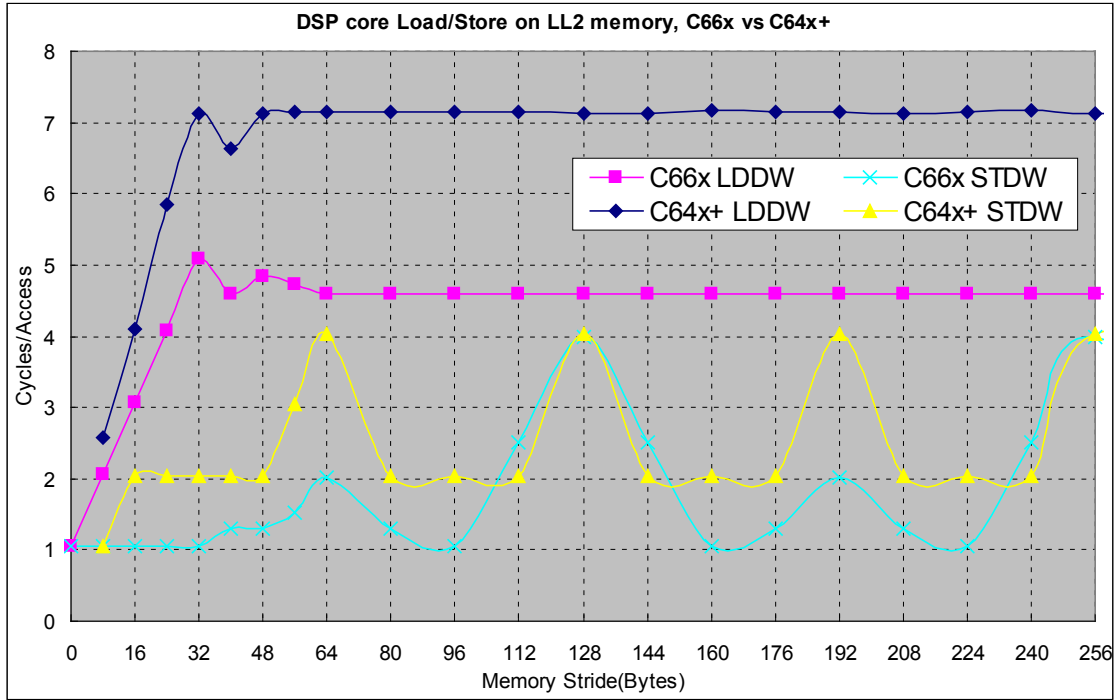
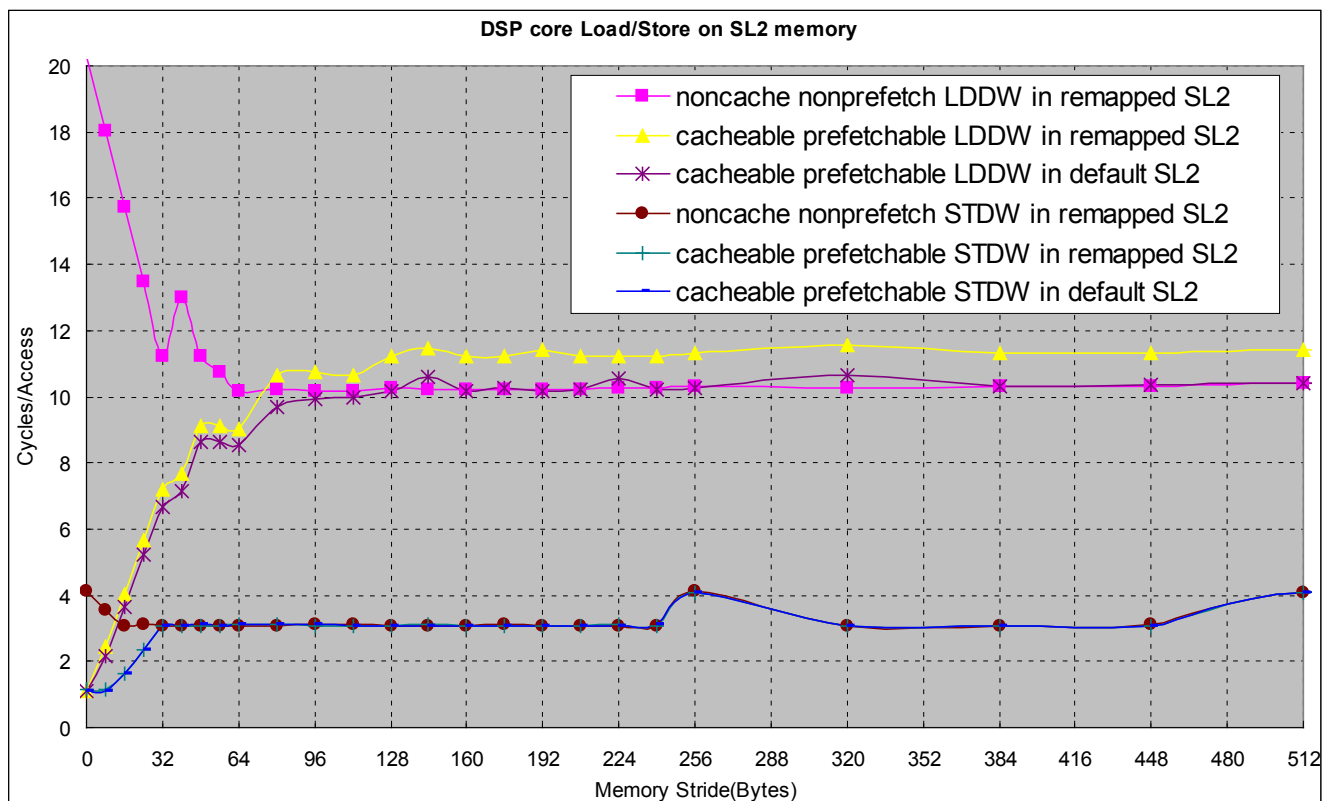


图 3 C66x 和 C64x+核在 LL2 上 Load/Store 的时延比较

### 3.2 DSP 核访问 SL2 的时延

图 4 是在 1GHz C6678 EVM 上测得的 DSP 核访问 SL2 的时延。DSP 核执行 512 个连续的 LDDW (Load Double Word)或 STDW (Store Double Word)指令所花的时间被测量，平均下来每个操作所花的时间被画在图中。测试中，L1D 被配置成 32KB cache。



**图 4 DSP 核访问 SL2**

对 LDB/STB 和 LDW/STW 的测试表明，它们的时延与 LDDW/STDW 相同。

DSP 核读 SL2 通常会通过 L1D cache，所以，和访问 LL2 一样，DSP 核访问 SL2 的性能高度依赖 cache。

XMC 中还有一个 prefetch buffer (8x128bytes)，它可以被看作是一个额外的只对读操作可用的 cache。DSP 核之外的每 16-MB 存储器块都可以通过 MAR (Memory Attribute Register) 的 PFX (PreFetchable eXternally) bit 被配置为是否通过 prefetch buffer 读，使能它会对多个主模块共享存储器的效率有很大帮助；它也能显著地改善对 SL2 连续读的性能。不过，prefetch buffer 对写操作没有任何作用。

SL2 可以通过从 0x0C000000 开始的缺省的地址空间访问，这个空间总是 cacheable，通常它也被配置为 prefetchable。SL2 可以通过 XMC 的配置被重映射到其它地址空间，通常重映射空间被用作 non-cacheable, nonprefetchable 访问（当然它也可以被设置为 cacheable 而且 prefetchable）。通过缺省地址空间访问比通过重映射空间访问稍微快一点，因为地址重映射需要一个额外的时钟周期。

由于 L1D cache 不会在写操作时被分配，并且这里的测试之前 cache 都被清空了，所以任何对 SL2 的写操作都通过 L1D write buffer (4x16bytes)。对多个写操作，如果地址偏移小于 16 bytes，这些操作可能在 write buffer 中被合并成一个对 SL2 的写操作，从而获得比较高的效率。XMC 也有类似的写合并 buffer，它可以合并两个在 32 bytes 内的写操作，所以，对偏移小于 32 bytes 的写操作，XMC 的写 buffer 改善了写操作的性能。

当写偏移是  $N \times 256$  bytes 时，每个写操作总是访问 SL2 相同的 bank (SL2 存储器组织结构是 4 bank x 2 sub-bank x 32 bytes)，对相同 bank 的连续访问间隔是 4 个时钟周期。对其它的访问偏移量，连续的写操作会访问 SL2 不同的 bank，这样的多个访问的在流水线上可以被重叠起来，从而使平均的访问时延比较小。

图 5 比较了 DSP 核访问 SL2 和 LL2 的访问时延。对地址偏移小于 16 bytes 的连续访问，访问 SL2 的性能和 LL2 几乎相同。而对地址偏移比较大的连续访问，访问 SL2 的性能比 LL2 差。因此，SL2 最适合于存放代码。

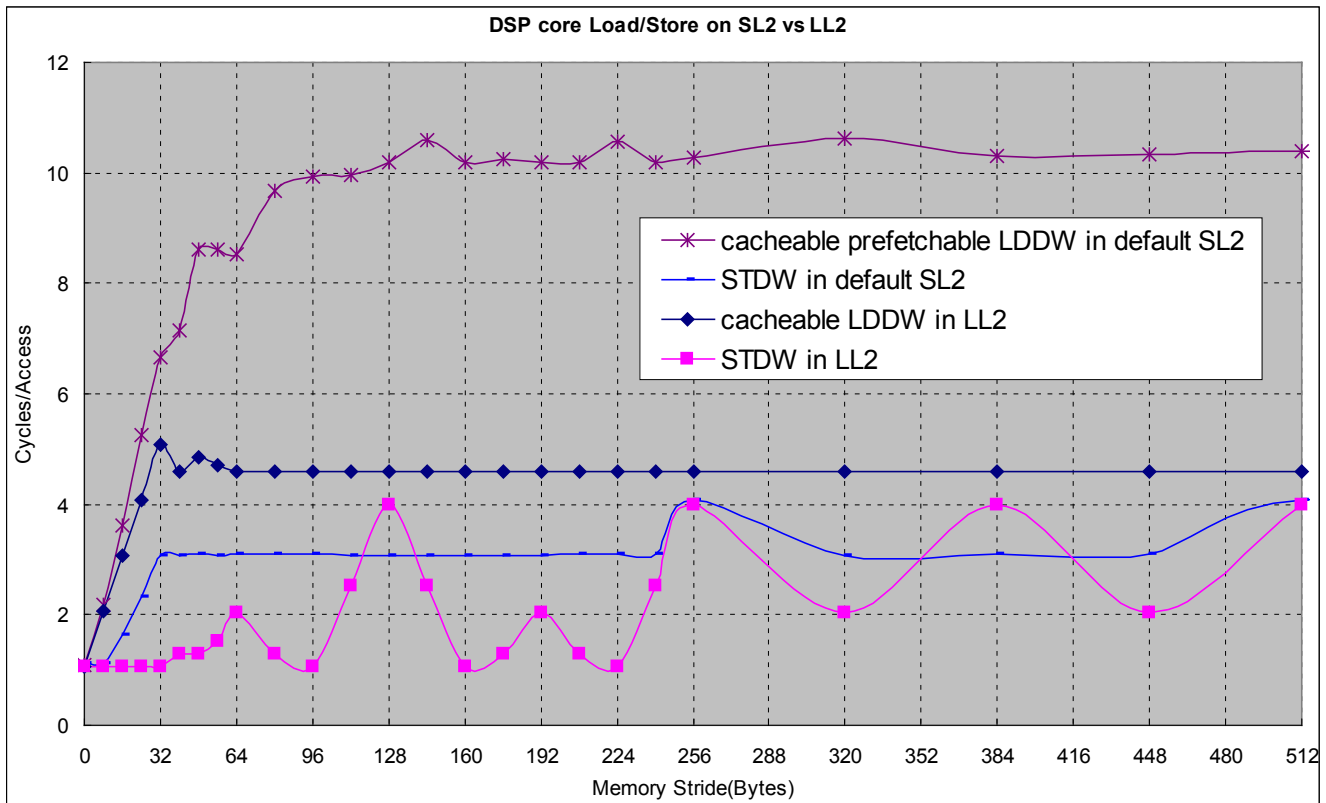


图 5 DSP 核访问 SL2 和 LL2 的性能比较

### 3.3 DSP 核访问外部 DDR 存储器的时延

DSP 核访问外部 DDR 存储器高度依赖 cache。当 DSP 核访问外部存储器时，一个传输请求会被发给 XMC。根据 cacheable 和 prefetchable 的设置，传输请求可能是下列情况中的一种：

- ◆ 一个数据单元 – 如果存储器空间是 non-cacheable, nonprefetchable

- ◆ 一个 L1 cache line - 如果存储器空间是 cacheable 而没有 L2 cache,
- ◆ 一个 L2 cache line - 如果存储器空间是 cacheable 并且设置了 L2 cache。

如果要访问的数据在 L1/L2 cache 或 prefetch buffer 中, 则不会有传输请求发出。

如果被访问的空间是 prefetchable 的, 可能还会产生额外的 prefetch 请求。

外部存储器的内容可以被缓存在 L1 cache 或/和 L2 cache, 或者都不用。DSP 核之外的每 16-MB 存储器块都可以通过 MAR (Memory Attribute Register) 的 PC (Permit Copy) bit 被配置为是否通过 cache 访问。如果 PC 比特为 0, 这段空间就不是 cacheable 的。如果 PC 比特是 1 而 L2 cache 大小为 0 (所有 LL2 都被用作普通 SRAM), 那外部存储器的内容只会被 L1 cache 缓存。如果 PC 比特是 1 并且 L2 cache 大于 0, 则外部存储器的内容可以被 L1 和 L2 cache 同时缓存。

像访问 SL2 一样, 对外部存储器的读操作也可以利用 XMC 里的 prefetch buffer。它可以通过 MAR (Memory Attribute Register) 的 PFX (PreFetchable eXternally) bit 来配置。

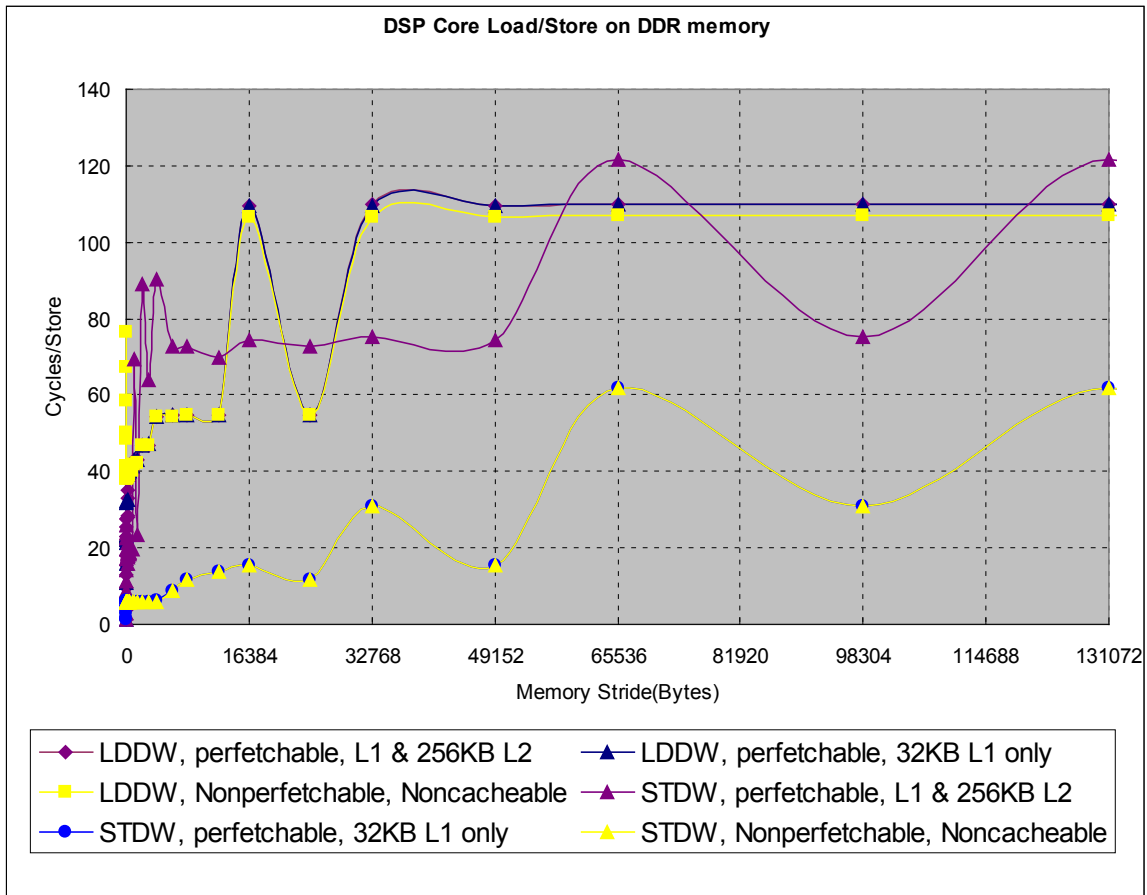
多个访问之间的地址偏移(stride)显著地影响访问效率, 地址连续的访问可以充分地利用 cache 和 prefetch buffer; 大于或等于 64 字节的地址偏移导致每次访问都 miss L1 cache 因为 L1D cache 行大小是 64 bytes; 大于或等于 128 字节的地址偏移导致每次访问都 miss L2 cache 因为 L2 cache 行大小是 128 bytes。

如果发生 cache miss, DSP 需要等待外部数据传输完成。等待的时间是请求发出时间, 数据传输时间或数据返回时间的总和。

图 6 是在 1GHz C6678 EVM (64-bit 1333MTS DDR) 上测得的 DSP 核访问 DDR 的时延。DSP 核执行 512 个连续的 LDDW (LoaD Double Word)或 STDW (STore Double Word)指令所花的时间被测量, 平均下来每个操作所花的时间被画在图中。测试中, L1D 被配置成 32KB cache, LL2 的 256KB 被设置为 cache。

对 LDB/STB 和 LDW/STW 的测试表明, 它们的时延与 LDDW/STDW 相同。

注意, 下面第二和第三个图实际上是第一个图左边的放大。



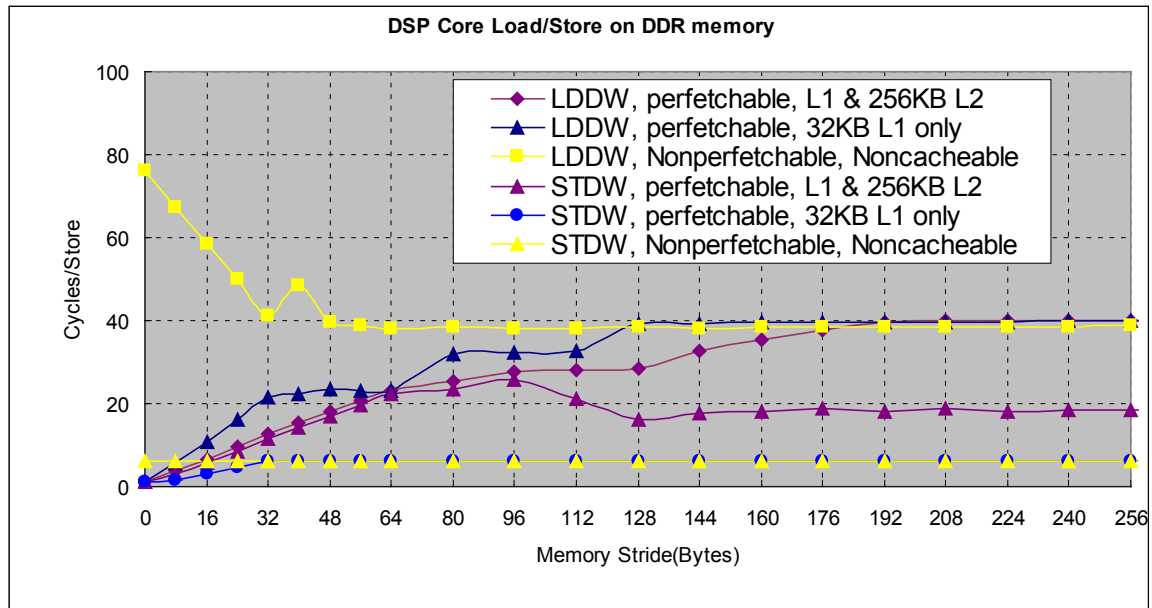
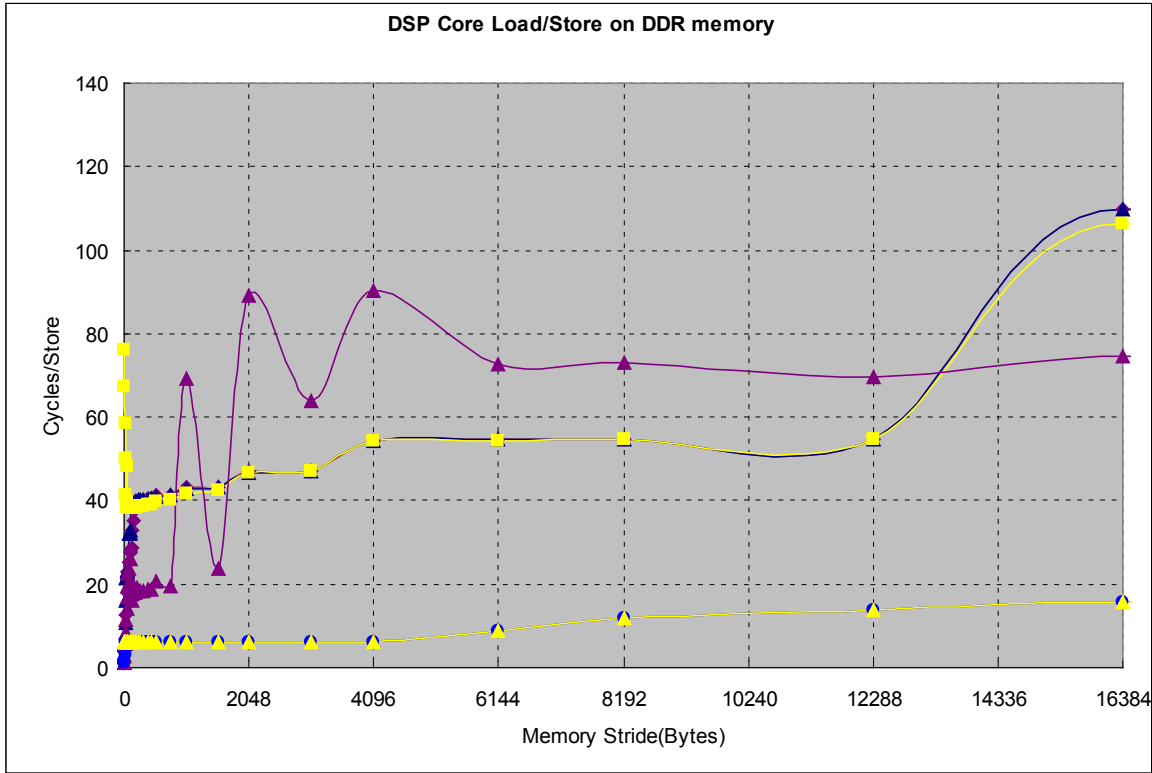


图 6 DSP 核对于 DDR Load/Store 的时延

对地址偏移小于 128 bytes 的访问，性能主要受 cache 的影响。



L2 cache 会在写操作时被分配，对任何写操作，cache 控制器总是先把被访问的数据所在的 cache 行（128 bytes）读进 L2 cache，然后在 cache 中改写数据。被改写的数据会在发生 cache 冲突或手工 cache 回写操作时被最终写到外部存储器里。当写操作的地址偏移是 1024 bytes 的整数倍时，多个访问在 L2 cache 中发生冲突的概率很大，所以 L2 cacheable 写操作的时延会显著地增加。最坏的情况下，每个写操作都会导致一个 cache 行的回写 (之前的数据因为冲突而被替换/回写) 和一个 cache 行的读入 (新的数据被分配到 cache 中)。

当地址偏移大于 512 bytes 时，DDR 页（行）切换开销成为性能下降的主要因素。C6678 EVM 上的 DDR 页（行）大小或 bank 宽度是 8KB，而 DDR3 存储器包含 8 个 banks。最坏的情况是，当访问地址偏移量是 64KB 时，每个读或写操作都会访问相同 bank 中一个新的行，而这种行切换会增加大约 40 个时钟周期的时延。请注意，不同的 DDR 存储器的时延可能会不一样。

## 4. DMA 访问存储器的性能

EDMA3 架构支持很多功能，可以实现高效的并行数据传输。本节讨论影响它性能的很多因素，如存储器类型，地址偏移等。

### 4.1 DMA 传输的额外开销

一般的传输时延被定义为 EDMA 被触发到真正的数据传输开始的时间。由于数据传输开始的时间无法用简单的方法测量，所以我们用最小数据单元的传输完成时间来代表 DMA 传输的时延或额外开销。根据不同源/目的地址的组合，这个值会有所不同。表 4 列出了在 1GHz C6678 EVM（64-bit 1333MTS DDR）上测得的从 EDMA 触发(写 ESR)最小数据传输(1 word)到 EDMA 传输结束(读到 IPR=1) 的平均的时钟周期数。

表 4 EDMA CC0 传输的额外开销

源 \ 目的	L1D	LL2	SL2	DDR
L1D	331	333	300	333
LL2	333	331	300	331
SL2	300	267	267	268
DDR	334	331	334	399

表 5 EDMA CC1 和 EDMA CC2 传输的额外开销

源 \ 目的	L1D	LL2	SL2	DDR
L1D	271	271	322	322
LL2	271	271	322	322
SL2	322	322	322	373
DDR	373	373	373	475

由于 EDMA CC0 连接到离 SL2 和 DDR 比较近的内部总线交换网络，因此它访问 SL2 和 DDR 额外开销比较小。而 EDMA CC1 和 CC2 连接到离 DSP 核的 L1 和 LL2 比较近的内部总线交换网络，所以它们访问 L1 和 LL2 的额外开销比较小。

IDMA 一般用来在 LL2 内部拷贝数据，测试得到的 IDMA 的平均额外开销是 61 个时钟周期。

传输的额外开销是小数据量传输要考虑的重要因素。单个数据单元的传输时间完全由 DMA 的额外开销决定。所以，对于小数据量的拷贝，我们需要在用 DMA 和用 DSP 核之间做权衡选择。

## 4.2 EDMA 10 个传输引擎的区别

C6678 上包含 10 个 EDMA3 TC (Transfer Controller)。这 10 个传输引擎并不完全相同。Table 6 总结了它们之间的区别。

表 6 EDMA 传输引擎之间的区别

参数	TC0_0	TC0_1	TC1_0	TC1_1	TC1_2	TC1_3	TC2_0	TC2_1	TC2_2	TC2_3
总线宽度(bits)	256	256	128	128	128	128	128	128	128	128
相对DSP核的速度	1/2	1/2	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3
FIFO大小(bytes)	1024	1024	1024	512	1024	512	1024	512	512	1024
缺省突发数据块大小(bytes)	128	128	128	64	128	64	128	64	64	128

表 7 比较了在 1GHz C6678 EVM (64-bit 1333MTS DDR) 上测得的各个 EDMA TC 的最大吞吐量。测试的参数是 ACNT=1024, BCNT=128, AB\_Sync (一次传完 ACNTxBCNT)。

表 7 1GHz C6678 上 EDMA TC 吞吐量比较

MB/s	TC0_0	TC0_1	TC1_0	TC1_1	TC1_2	TC1_3	TC2_0	TC2_1	TC2_2	TC2_3
LL2 -> LL2	5252	5252	5036	5036	5036	5036	5036	5036	5036	5036
LL2 -> SL2	5267	5267	5036	5036	5036	5036	5036	5036	5036	5036
SL2 -> LL2	5267	5267	5036	5036	5036	5036	5036	5036	5036	5036
SL2 -> SL2	7880	7880	5026	5026	5026	5026	5026	5026	5026	5026
LL2 -> DDR	5267	5267	5026	4987	5026	4987	5026	4997	4987	5026
DDR -> LL2	5252	5211	4948	3597	4968	3602	4977	3592	3602	4968
DDR -> DDR	3870	3878	3091	2976	3091	3043	3091	2980	2976	3091
SL2->DDR	10396	10396	5026	5026	5026	5026	5026	5026	5026	5026
DDR->SL2	9780	9926	5016	3519	5016	3553	5016	3523	3562	5016

对 SL2 和 DDR 之间的传输，TC0\_0 和 TC0\_1 能达到的吞吐量是其它 TC 的两倍。在本文其它部分，如果没有特殊说明，EDMA 性能数据都是在 TC0\_0 上测得的。

### 4.3 EDMA 带宽和传输灵活性的权衡

EDMA3 支持很多灵活的传输参数配置。多数配置情况下 EDMA 可以充分利用存储器带宽；但在某些情况下，传输性能可能会有所下降。为了设计高效的系统，我们需要理解什么样的配置能够实现高效的传输，而某些情况下必须在灵活性和效率之间做权衡。

#### 4.3.1 第一维大小 (ACNT) 的考虑 (突发数据块大小)

为了充分利用传输引擎的带宽，传输尽量大的数据块是非常重要的。

为了充分利用 128-bit 或 256-bit 的总线，ACNT 应该是 16 bytes 的整数倍；为了充分利用 EDMA 的突发数据块，ACNT 需要是 64 bytes 的整数倍；为了充分利用 EDMA FIFO，ACNT 应该至少是 512 bytes。

图 7 画出了在 1GHz C6678 EVM (64-bit 1333MTS DDR) 上，从 SL2 到 DDR 传输 1~24K bytes 数据时测得的吞吐量。

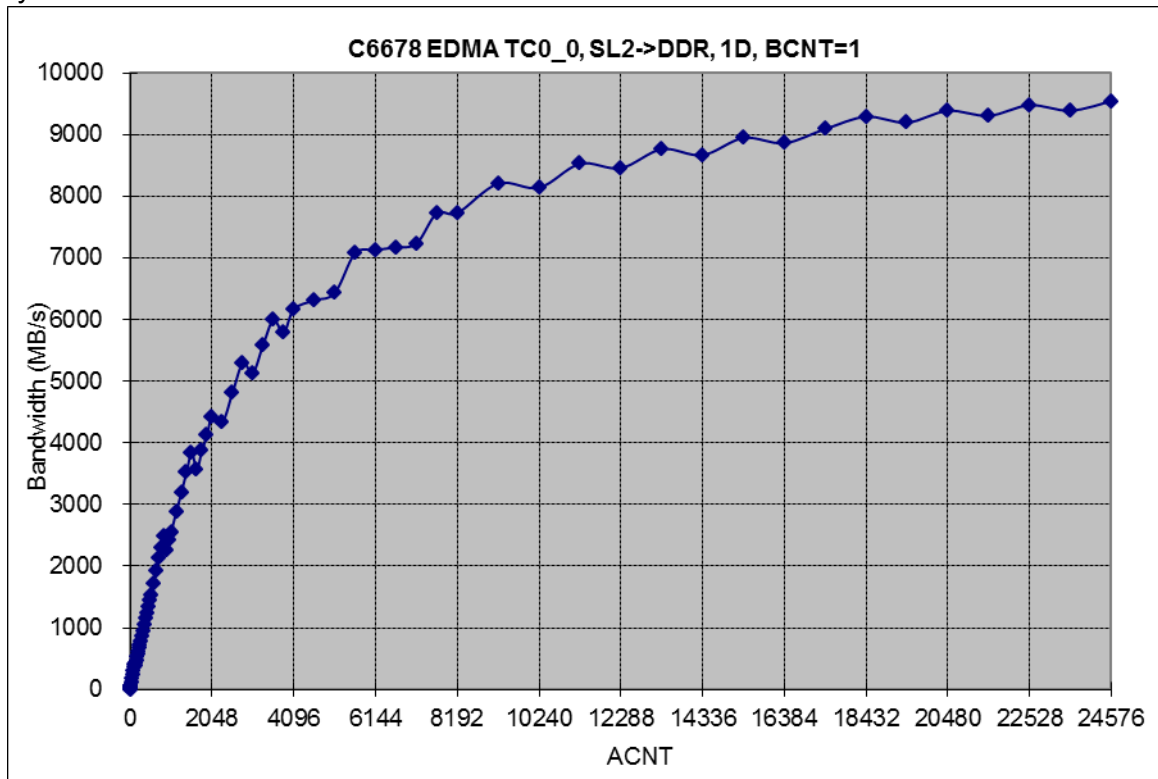


图 7 ACNT 对 EDMA 效率的影响

从测试结果可以看出，ACNT 越大，带宽的利用率越高。

### 4.3.2 二维传输的考虑 (传输优化)

如果 2D 传输 (AB\_Sync) 是线性的 (BIDX=ACNT), 并且 ACNT 是 2 的幂次方, EDMA 会把这个 2D 传输优化为 1D 传输。我们在 1GHz C6678 EVM (64-bit 1333MTS DDR) 上测试了不同的 ACNT 和 BCNT 的组合; 图 8 是线性 2D 传输的测试结果, 它说明, 不管 BCNT 是多少, 带宽由 ACNT $\times$ BCNT 的乘积决定。

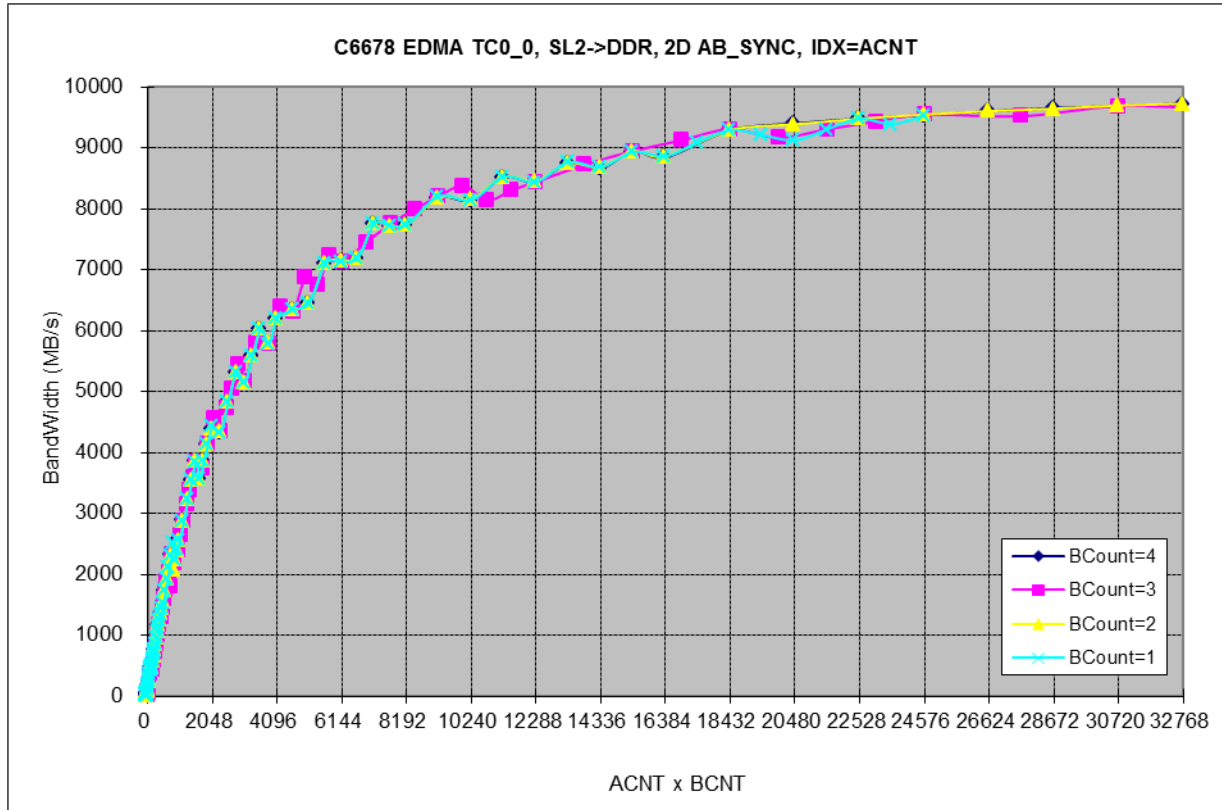


图 8 线性 2D 传输

如果 2D 传输不是线性的, 带宽的利用率完全有 ACNT 决定 (参见图 7)。

### 4.3.3 地址偏移的考虑

地址偏移会显著地影响 EDMA 的吞吐量。

图 9 说明了地址偏移对 EDMA 吞吐量的影响, 它是在 1GHz C6678 EVM (64-bit 1333MTS DDR) 上从 SL2 到 DDR 传输 1024 行 (BCNT=1024) 2D 数据时测得的。

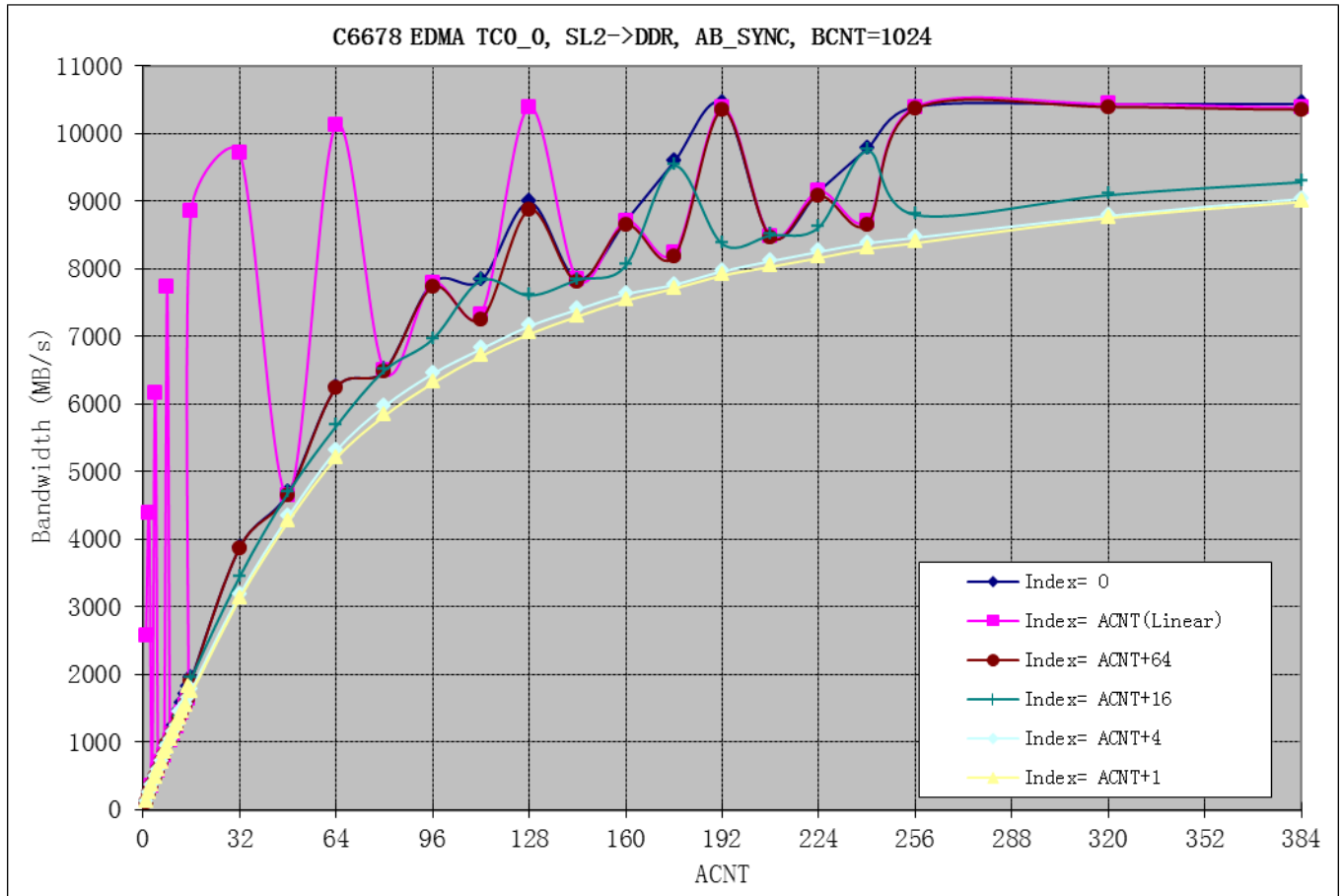


图 9 偏移量对 EDMA 带宽的影响

从测试结果可以看出，线性传输 (Index= ACNT)能充分利用带宽；其它 Index 设置会降低 EDMA 性能。最坏的情况是地址偏移为奇数时。如果地址偏移大于 8 并且是 2 的幂次方，性能的下落则比较小。

请注意，Index= ACNT，并且 ACNT 是 2 的幂次方时，2D 传输被优化为 1D 传输，因此性能比其它情况好很多。

除非特殊说明，本文列出的所有性能数据都是在 Index= ACNT 的情况下测得的。

#### 4.3.4 地址对齐

地址对齐对 EDMA 效率稍有影响。EDMA3 缺省突发数据块大小是 64 bytes 或 128 bytes，如果传输跨越 64 或 128 bytes 边界，EDMA3 TC 会把大小为 ACNT 数据块分割成 64 或 128 bytes 的突发数据块。这对 1~256 bytes 的数据传输影响会比较明显，而对更大块数据的传输的影响则不明显。

除非特殊说明，本文所有性能数据都是在地址对齐的情况下测得的。

## 5. 多个主模块共享存储器的性能

由于 C6678 有 8 个核和很多 DMA 主模块，它们可能会同时访问存储器。本节讨论多个主模块共享存储器的性能。

### 5.1 多个主模块共享 SL2 的性能

图 10 列出了数据在 SL2 中的组织结构。

Bank 0			Bank 1			Bank 2			Bank 3		
byte 0	...	byte 31	byte 64	...	byte 95	byte 128	...	byte 159	byte 192	...	byte 223
byte 32	...	byte 63	byte 96	...	byte 127	byte 160	...	byte 191	byte 224	...	byte 255
byte 256	...	byte 287	byte 320	...	byte 351						
byte 288	...	byte 319	byte 352	...	byte 383						
...	...	...	...	...	...	...	...	...	...	...	...

图 10 SL2 bank 组织结构

所有主模块都可以通过 MSMC (Multicore Shared Memory Controller) 独立地访问 4 个 SL2 bank 中的任一个。多个主模块可以并行地访问不同的 bank；如果多个主模块要同时访问相同的 bank，那就需要根据优先级仲裁。

表 8 列出了在 1GHz C6678 上测得的多个主模块同时访问 SL2 的性能数据。每个主模块反复访问自己在 SL2 中的数据 buffer，在相同的时间内（大概 2 秒），每个主模块传输的数据量被统计；而每个主模块获得的带宽则由数据量除以时间计算出来。

在这个测试中，每个核的 L1D cache 大小是 32KB，没有使用 L2 cached，prefetch buffer 被使用。

在下面的表中，每列是一个测试场景的结果，不同测试场景的主要区别是同时访问存储器的主模块的个数，格子中的数据代表相应的主模块在这个测试场景下获得的带宽，一行中的空格代表对应的主模块在这个测试场景下未被使用。最后一行中的数据是在这个测试场景下所有主模块获得的带宽的总和。

**表 8 多个 DSP 核共享 SL2 的性能**

<b>SL2-&gt;LL2 带宽(MB/s), 相同优先级</b>											
核 0	2249	2249	2249	2249	2249	2249					
核 1		2243	2243	2243	2242	2242					
核 2			2249	2249	2249	2249	2249				
核 3				2242	2242	2241	2242				
核 4					2249	2248	2249	2250			
核 5					2241	2241	2242	2243	2243		
核 6						2248	2249	2249	2249	2250	
核 7						2240	2241	2242	2243	2243	2243
<b>总和</b>	<b>2265</b>	<b>4508</b>	<b>6757</b>	<b>8999</b>	<b>13488</b>	<b>17974</b>	<b>13472</b>	<b>8984</b>	<b>6735</b>	<b>4493</b>	<b>2243</b>
<b>LL2-&gt;SL2 带宽(MB/s), 相同优先级</b>											
核 0	3636	3636	3636	3636	3636	3636					
核 1		3635	3635	3635	3633	3615					
核 2			3635	3635	3633	3615	3634				
核 3				3635	3633	3615	3634				
核 4					3633	3615	3634	3635			
核 5					3633	3615	3634	3635	3635		
核 6						3615	3634	3635	3635	3636	
核 7						3615	3634	3635	3635	3636	3636
<b>总和</b>	<b>3677</b>	<b>7312</b>	<b>10947</b>	<b>14582</b>	<b>21839</b>	<b>28964</b>	<b>21804</b>	<b>14540</b>	<b>10905</b>	<b>7272</b>	<b>3636</b>

以上测试结果证明 SL2 不会成为多个 DSP 核同时访问的瓶颈。SL2 有足够的带宽 (500M x 32 x 4 = 64000MB/s)来支持所有 DSP 核的同时访问。每个 DSP 核的吞吐量受限于它自己。

由于 SL2 的带宽足够支持所有核同时访问，所以核的优先级在这种情况下基本不起作用。

表 9 多个 EDMA 共享 SL2 的性能

带宽(MB/s), DDR->SL2 by DMA CC0, LL2->SL2 by other DMA, 相同优先级												
DMA0 TC0		9942	5245	4011	2670	2003	1996					
DMA0 TC1			5245	4011	2670	2003	1996					
DMA1 TC0				3963	2645	1989	1336	1788				
DMA1 TC1				3963	2645	1989	692	938				
DMA1 TC2					2645	1989	1983	2642	2672			
DMA1 TC3					2645	1989	1982	2639	2672			
DMA2 TC0						1983	1977	2629	2644	3973		
DMA2 TC1						1983	1976	2628	2644	3973		
DMA2 TC2							673	904	2644	3973	5064	
DMA2 TC3							1317	1755	2644	3973	5064	
总和		<b>9942</b>	<b>10490</b>	<b>15948</b>	<b>15920</b>	<b>15928</b>	<b>15928</b>	<b>15923</b>	<b>15920</b>	<b>15892</b>	<b>10128</b>	<b>5064</b>
带宽(MB/s), SL2->DDR by DMA CC0, SL2->LL2 by other DMA, 相同优先级												
DMA0 TC0		10388	5247	4526	3184	2436	2431					
DMA0 TC1			5247	4526	3184	2436	2431					
DMA1 TC0				4526	3184	2455	1245	1804				
DMA1 TC1				2263	1592	1231	628	916				
DMA1 TC2					3184	2454	2451	3513	3554			
DMA1 TC3					1592	1229	1230	1770	1787			
DMA2 TC0						2436	2436	3489	3502	5086		
DMA2 TC1						1226	1229	1763	1791	2719		
DMA2 TC2							623	903	1789	2714	4885	
DMA2 TC3							1214	1749	3440	5027	4885	5066
总和		<b>10388</b>	<b>10494</b>	<b>15841</b>	<b>15920</b>	<b>15903</b>	<b>15918</b>	<b>15907</b>	<b>15863</b>	<b>15546</b>	<b>9770</b>	<b>5066</b>
带宽(MB/s), DDR->SL2 by DMA CC0, LL2->SL2 by other DMA, 不同优先级												
DMA0 TC0	0	9942	9175	7628	7701	7701	6893					
DMA0 TC1	1		1320	2793	2728	2728	1910					
DMA1 TC0	2			4023	4013	4012	3449	3527				
DMA1 TC1	3			1412	1378	1379	626	1881				
DMA1 TC2	4				102	102	2	5069	5098			
DMA1 TC3	5				5	5	0	224	664			
DMA2 TC0	6					0	0	4	71	5045		
DMA2 TC1	7					0	0	0	4	720		
DMA2 TC2	0						1789	1776	5044	5045	5064	
DMA2 TC3	1						1231	3419	5044	5045	5064	5064
总和		<b>9942</b>	<b>10495</b>	<b>15856</b>	<b>15927</b>	<b>15927</b>	<b>15900</b>	<b>15900</b>	<b>15925</b>	<b>15855</b>	<b>10128</b>	<b>5064</b>



带宽(MB/s), SL2->DDR by DMA CC0, SL2->LL2 by other DMA, 不同优先级												
DMA0 TC0	0	10388	5497	5496	5495	5495	7164					
DMA0 TC1	1		4997	4996	4996	4996	109					
DMA1 TC0	2			5075	5074	5074	3322	3367				
DMA1 TC1	3			170	150	150	30	951				
DMA1 TC2	4				45	45	0	21	2654			
DMA1 TC3	5				0	0	0	0	123			
DMA2 TC0	6					0	0	0	9	2644		
DMA2 TC1	7					0	0	0	0	137		
DMA2 TC2	0						1921	1901	3403	3407	4921	
DMA2 TC3	1						59	1681	4707	4707	4921	5066
总和		<b>10388</b>	<b>10494</b>	<b>15737</b>	<b>15760</b>	<b>15760</b>	<b>12605</b>	<b>7921</b>	<b>10896</b>	<b>10895</b>	<b>9842</b>	<b>5066</b>

由于总共有 10 个 EDMA TC，而只有 8 个 DSP 核，在这些测试里，8 个 TC 在 SL2 和 8 个核的 LL2 之间传输数据，另两个 TC 则在 SL2 和 DDR 之间传输数据。

尽管 SL2 本身的带宽非常高，但所有 EDMA 访问 SL2 时都需要经过内部总线交换网络的一个相同的端口，这个端口就成了这种情况下的瓶颈。这个端口的理论带宽是  $500\text{MHz} \times 32 \text{ bytes} = 16000\text{MB/s}$ 。如果所有 EDMA 传输的优先级相同，带宽会在这些传输之间平均分配。而当优先级不同时(上表中第二列是每个主模块的优先级)，优先级低的 EDMA 传输得到的带宽比较小。对负载很高的情况下，一些低优先级的传输可能会被“饿死”，也就是说，得到的带宽为 0。

根据Table 6中列出的不同 EDMA TC 的区别，EDMACC1 TC1 和 TC3，EDMACC2 TC1 和 TC2 在某些情况下获得的带宽会比其它 TC 小，即使优先级相同。

## 5.2 多个主模块共享 DDR 的性能

多个主模块同时访问 DDR 时，DDR 控制器会根据主模块的优先级进行仲裁。

C6678 的 DDR 控制器支持包含 1, 2, 4, 或 8 个 banks 的 DDR 存储器。现在大多数 DDR3 存储器都包含 8 个 bank，以 C6678 EVM 上的 DDR 存储器为例，它的组织结构如图 11 所示。请注意，对不同的 DDR 存储器，每一行的大小可能会不同。

	Bank0	Bank1	...	Bank7
Row 0	byte 0~8191	byte 8192~8192*2-1	...	byte 8192*7~8192*8-1
Row 1	byte 8192*8~8192*9-1	byte 8192*9~8192*10-1	...	byte 8192*15~8192*16-1
...	...	...	...	...

图 11 DDR bank 数据组织结构

尽管 DDR 存储器有多个 bank，但与 SL2 的 bank 不同的是，它们并没有独立的总线与控制器相连，而是共享相同的总线接口。所以，DDR 存储器 bank 的个数并不直接的影响带宽，但它却显著的影响多个主模块共享 DDR 的效率。

DDR SDRAM 的访问以行或页为基础。一个主模块在访问某个页中的数据之前，这一页必须首先被“打开”，然后这一页中的内容才可以被随意访问。如果主模块想访问同一个 bank 中另一页中的数据，必须先关闭原先的页，然后打开新的一页。而页切换(关闭/打开)操作需要消耗额外的时钟周期。

每个 bank 只可以有一页被打开，DDR 存储器的 bank 数越多，页切换的概率就越小。例如，一个主模块在访问了 bank0 的 0 页后，它可以访问 bank1 的 1 页而不用关闭之前访问的 bank0 的 0 页，然后这个主模块可以自由的在这两页之间访问而不会产生页切换。包含 8 个 bank 的 DDR 存储器可以有 8 页同时被打开。

为了测试页切换的影响，我们定义了两种测试的数据结构。

	Bank 0	Bank 1	Bank2	...	Bank n	...
Row 0	主模块 0 访问范围					
Row 1	主模块 1 访问范围					
Row 2	主模块 2 访问范围					
...	...					
Row n	主模块 n 访问范围					
...	...					

图 12 多个主模块访问相同 DDR bank 的不同页

上面的情况是最坏的情况，页切换的额外开销被最大化。每换一个主模块访问就会导致一次页切换。

下面的情况则是最佳情况，每个主模块始终访问打开的页，而不会有任何页切换发生。

	Bank 0	Bank 1	Bank2	...	Bank n	...
Row 0	主模块 0 访问范围					
Row 1		主模块 1 访问范围				
Row 2			主模块 2 访问范围			
...				...		

Row n					主模块 n 访问范围	
...	...					...

图 13 多个主模块访问不同 DDR bank 的不同页

### 5.2.1 多个 DSP 核共享 DDR 的性能

下面的表格列出了不同场景下多个 DSP 核共享 1GHz C6678 EVM 上 64-bit 1333MTS DDR 的性能。每个主模块反复访问自己在 DDR 中的数据 buffer，在相同的时间内（大概 2 秒），每个主模块传输的数据量被统计；而每个主模块获得的带宽则由数据量除以时间计算出来。

在这些测试中，DDR 是 cacheable 且 prefetchable 的，L1D cache 是 32KB，L2 cache 是 256KB，prefetch buffer 被使能。我们没有测试 Non-cacheable 的情况是因为 Non-cacheable 情况下每个核需要的带宽比 cacheable 的情况下要少很多。

在下面的表中，每列是一个测试场景的结果，不同测试场景的主要区别是同时访问存储器的主模块的个数，格子中的数据代表相应的主模块在这个测试场景下获得的带宽，一列中的空格代表对应的主模块在这个测试场景下未被使用。最后一行中的数据是在这个测试场景下所有主模块获得的带宽的总和。

表 10 多个 DSP 核共享 DDR 的性能

DDR->LL2 带宽(MB/s), 不同 bank 不同页, 相同优先级										
核 0		1631	1626	1606	1536	1288				
核 1			1602	1584	1524	1283				
核 2				1584	1524	1283	1529			
核 3				1584	1524	1283	1525			
核 4					1524	1283	1525	1597		
核 5					1524	1283	1525	1585		
核 6						1283	1525	1585	1619	
核 7						1283	1525	1585	1604	1626
总和		1631	3228	6358	9156	10269	9154	6352	3223	1626
DDR->LL2 带宽(MB/s), 相同 bank 不同页, 相同优先级										
核 0		1632	1419	1060	720	539				
核 1			1412	1057	719	539				
核 2				1057	719	539	719			
核 3				1057	719	539	719			
核 4					719	539	719	1068		
核 5					719	539	719	1068		
核 6						539	719	1068	1357	

核 7						539	719	1068	1357	1626
总和		<b>1632</b>	<b>2831</b>	<b>4231</b>	<b>4315</b>	<b>4312</b>	<b>4314</b>	<b>4272</b>	<b>2714</b>	<b>1626</b>
<b>LL2-&gt;DDR 带宽(MB/s), 不同 bank 不同页, 相同优先级</b>										
核 0		1858	1844	1816	1643	1294				
核 1			1821	1796	1633	1290				
核 2				1796	1633	1290	1651			
核 3				1796	1633	1290	1650			
核 4					1633	1290	1650	1800		
核 5					1633	1290	1650	1791		
核 6						1290	1650	1791	1836	
核 7						1290	1650	1791	1828	1848
总和		<b>1858</b>	<b>3665</b>	<b>7204</b>	<b>9808</b>	<b>10324</b>	<b>9901</b>	<b>7173</b>	<b>3664</b>	<b>1848</b>
<b>LL2-&gt;DDR 带宽(MB/s), 相同 bank 不同页, 相同优先级</b>										
核 0		1858	1602	1079	719	539				
核 1			1593	1079	719	539				
核 2				1079	719	539	719			
核 3				1079	719	539	719			
核 4					719	539	719	1078		
核 5					719	539	719	1078		
核 6						539	719	1078	1516	
核 7						539	719	1078	1516	1848
总和		<b>1858</b>	<b>3195</b>	<b>4316</b>	<b>4314</b>	<b>4312</b>	<b>4314</b>	<b>4312</b>	<b>3032</b>	<b>1848</b>
<b>DDR-&gt;LL2, 不同 bank 不同页, 不同优先级</b>										
核 0	0	1631	1628	1623	1611	1571				
核 1	1		1600	1594	1580	1535				
核 2	2			1582	1562	1479	1602			
核 3	3			1551	1521	1394	1580			
核 4	4				1433	1247	1561	1614		
核 5	5				1272	1076	1518	1594		
核 6	6					943	1426	1582	1621	
核 7	7					827	1266	1550	1600	1626
总和		<b>1631</b>	<b>3228</b>	<b>6350</b>	<b>8979</b>	<b>10072</b>	<b>8953</b>	<b>6340</b>	<b>3221</b>	<b>1626</b>
<b>LL2-&gt;DDR, 不同 bank 不同页, 不同优先级</b>										
核 0	0	1858	1851	1844	1825	1731				
核 1	1		1811	1804	1778	1681				
核 2	2			1792	1741	1588	1813			
核 3	3			1707	1620	1397	1779			
核 4	4				1428	1159	1737	1833		

核 5	5				1180	983	1617	1806		
核 6	6					867	1427	1792	1841	
核 7	7					798	1181	1705	1813	1848
总和		<b>1858</b>	<b>3662</b>	<b>7147</b>	<b>9572</b>	<b>10204</b>	<b>9554</b>	<b>7136</b>	<b>3654</b>	<b>1848</b>

从上面的测试结果可以看出，多核同时访问**相同**DDR bank中不同页的性能比多核同时访问**不同**DDR bank中不同页的性能差很多，原因就是页切换的额外开销。

从上面的测试结果还可以看出，DDR的带宽(1333 x 8 = 10666MB/s)对所有DSP核同时访问来说是不够的，DSP核的优先级对它获得的带宽有明显影响。当优先级相同时，带宽在多个核之间平均分配；而当优先级不同时(上表中第二列是每个主模块的优先级)，优先级低的核得到的带宽比较小。

为了防止低优先级的主模块被“饿死”，DDR控制器提供了老请求优先级临时提高的功能。我们可以通过配置一个计数器，当一个老的请求的等待时间超过这个计数值时，它的优先级会被临时提高。如果没有特殊说明，本文所有测试中这个计数周期都被配置成4x16=64个DDR3CLKOUT时钟周期。在64个DDR3CLKOUT时钟周期内，可以传输64x2x8=1024 bytes。

表 11 是在 1GHz C6678 EVM (64-bit 1333MTS DDR) 上用不同的优先级提升计数值测得的数据。表中第二列是每个主模块的预设优先级。

表 11 DDR 优先级提升计数值的影响

LL2->DDR, 不同 bank, 不同优先级, 优先级提升计数值= 0									
核 0	0	1709	1614	1433	1253				
核 1	1	1694	1604	1428	1249	1434			
核 2	2	1694	1604	1428	1249	1434	1615		
核 3	3	1694	1604	1428	1249	1434	1615	1709	
核 4	4	1694	1604	1428	1249	1434	1615	1709	
核 5	5		1604	1428	1249	1434	1615	1709	
核 6	6			1428	1249	1434	1615	1709	
核 7	7				1249	1434	1615	1709	
总和		<b>8485</b>	<b>9634</b>	<b>10001</b>	<b>9996</b>	<b>10038</b>	<b>9690</b>	<b>8545</b>	
LL2->DDR, 不同 bank, 不同优先级, 优先级提升计数值= 64									
核 0	0	1838	1825	1802	1731				
核 1	1	1795	1778	1748	1681	1789			
核 2	2	1777	1741	1685	1588	1746	1813		
核 3	3	1684	1620	1529	1397	1668	1779	1827	
核 4	4	1512	1428	1298	1159	1516	1737	1797	
核 5	5		1180	1085	983	1294	1617	1775	
核 6	6			914	867	1089	1427	1675	
核 7	7				798	924	1181	1503	

总和		<b>8606</b>	<b>9572</b>	<b>10061</b>	<b>10204</b>	<b>10026</b>	<b>9554</b>	<b>8577</b>		
<b>LL2-&gt;DDR, different bank, 不同优先级, 优先级提升计数值= 4080</b>										
核 0	0	1843	1847	1848	1821					
核 1	1	1803	1806	1808	1792	1837				
核 2	2	1787	1788	1789	1773	1811	1836			
核 3	3	1706	1699	1699	1719	1786	1810	1833		
核 4	4	1456	1430	1418	1442	1699	1786	1805		
核 5	5		968	962	970	1420	1697	1786		
核 6	6			516	520	967	1424	1700		
核 7	7				214	518	969	1435		
总和		<b>8595</b>	<b>9538</b>	<b>10040</b>	<b>10251</b>	<b>10038</b>	<b>9522</b>	<b>8559</b>		

从上面的测试结果可以看出，优先级提升计数值=0 实际上使得预设的优先级不起作用。当这个计数值越大是，预设的优先级起的作用越大。所以，在实际应用中，设计者需要根据应用的需求来选择的一个合适的值。

### 5.2.2 多个 EDMA 共享 DDR 的性能

下面的表格列出了不同场景下多个 EDMA TC 共享 1GHz C6678 EVM 上 64-bit 1333MTS DDR 的性能。

表 12 多个 EDMA 共享 DDR 的性能

<b>SL2-&gt;DDR by DMA0, LL2-&gt;DDR by other DMA, 不同 bank 不同页, 相同优先级</b>												
DMA0 TC0		10383	5247	2631	1757	1317	1123					
DMA0 TC1			5247	2630	1757	1317	1123					
DMA1 TC0				2616	1745	1311	751	1174				
DMA1 TC1				2616	1745	1311	382	601				
DMA1 TC2					1745	1311	1119	1743	1757			
DMA1 TC3					1745	1311	1119	1742	1757			
DMA2 TC0						1308	1117	1739	1745	2623		
DMA2 TC1						1308	1117	1737	1745	2623		
DMA2 TC2							377	592	1745	2623	5019	
DMA2 TC3							746	1164	1745	2623	5019	
总和		<b>10383</b>	<b>10494</b>	<b>10493</b>	<b>10494</b>	<b>10494</b>	<b>8974</b>	<b>10492</b>	<b>10494</b>	<b>10492</b>	<b>10038</b>	<b>5060</b>
<b>SL2-&gt;DDR by DMA0, LL2-&gt;DDR by other DMA, 相同 bank 不同页, 相同优先级</b>												
DMA0 TC0		10388	3693	1280	652	398	356					
DMA0 TC1			3693	1280	652	397	356					
DMA1 TC0				1275	650	397	237	360				

DMA1 TC1				1275	650	397	119	181				
DMA1 TC2					650	397	355	538	653			
DMA1 TC3					650	397	355	538	653			
DMA2 TC0						397	355	538	651	1283		
DMA2 TC1						397	355	538	651	1283		
DMA2 TC2							119	181	651	1283	2728	
DMA2 TC3							237	359	651	1283	3895	5063
<b>总和</b>		<b>10388</b>	<b>7386</b>	<b>5110</b>	<b>3904</b>	<b>3177</b>	<b>2844</b>	<b>3233</b>	<b>3910</b>	<b>5132</b>	<b>6623</b>	<b>5063</b>
<b>DDR-&gt;SL2 by DMA0, DDR-&gt;LL2 by other DMA, 不同 bank 不同页, 相同优先级</b>												
DMA0 TC0		9942	5245	3161	2106	1621	1610					
DMA0 TC1			5245	3160	2105	1621	1610					
DMA1 TC0				2607	2097	1613	816	1183				
DMA1 TC1				1574	1048	809	410	596				
DMA1 TC2					2097	1612	1602	2312	2328			
DMA1 TC3					1048	808	803	1172	1188			
DMA2 TC0						1607	1597	2303	2312	3226		
DMA2 TC1						809	803	1170	1185	2016		
DMA2 TC2							402	591	1184	2016	3414	
DMA2 TC3							799	1172	2304	3225	4990	5005
<b>总和</b>		<b>9942</b>	<b>10490</b>	<b>10502</b>	<b>10501</b>	<b>10500</b>	<b>10452</b>	<b>10499</b>	<b>10501</b>	<b>10483</b>	<b>8404</b>	<b>5005</b>
<b>DDR-&gt;SL2 by DMA0, DDR-&gt;LL2 by other DMA, 相同 bank 不同页, 相同优先级</b>												
DMA0 TC0		9942	3926	2264	1032	640	475					
DMA0 TC1			3926	2264	1032	640	475					
DMA1 TC0				1422	1022	638	238	558				
DMA1 TC1				991	517	320	119	280				
DMA1 TC2					1020	638	474	1034	1129			
DMA1 TC3					516	320	237	563	648			
DMA2 TC0						637	473	1029	1116	1811		
DMA2 TC1						320	237	562	647	1451		
DMA2 TC2							119	279	647	1450	2710	
DMA2 TC3							237	555	1130	1803	3610	5005
<b>总和</b>		<b>9942</b>	<b>7852</b>	<b>6941</b>	<b>5139</b>	<b>4153</b>	<b>3084</b>	<b>4860</b>	<b>5317</b>	<b>6515</b>	<b>6320</b>	<b>5005</b>
<b>SL2-&gt;DDR by DMA0, LL2-&gt;DDR by other DMA, 不同 bank 不同页, 不同优先级</b>												
DMA0 TC0	0	10383	5496	4147	4140	4140	3551					
DMA0 TC1	1		4997	4147	4140	4140	1499					
DMA1 TC0	2			2122	2104	2104	2050	2572				
DMA1 TC1	3			78	90	90	435	1322				
DMA1 TC2	4				18	18	0	2630	2670			

DMA1 TC3	5				0	0	0	83	220			
DMA2 TC0	6					0	0	1	23	2697		
DMA2 TC1	7					0	0	0	0	261		
DMA2 TC2	0						1046	1316	3767	3769	5019	
DMA2 TC3	1						870	2506	3754	3708	5019	5060
总和		<b>10383</b>	<b>10493</b>	<b>10494</b>	<b>10492</b>	<b>10492</b>	<b>9451</b>	<b>10430</b>	<b>10434</b>	<b>10435</b>	<b>10038</b>	<b>5060</b>
<b>DDR-&gt;SL2 by DMA0, DDR-&gt;LL2 by other DMA, 不同 bank 不同页, 不同优先级</b>												
DMA0 TC0	0	9942	9175	5785	5726	5726	5051					
DMA0 TC1	1		1320	4338	4288	4288	1904					
DMA1 TC0	2			311	347	347	1368	2346				
DMA1 TC1	3			68	80	80	321	1012				
DMA1 TC2	4				57	57	26	1814	1879			
DMA1 TC3	5				2	2	0	1047	1195			
DMA2 TC0	6					0	0	565	1348	2175		
DMA2 TC1	7					0	0	9	87	1394		
DMA2 TC2	0						844	1310	2422	2903	3544	
DMA2 TC3	1						979	2359	3523	3993	4962	5005
总和		<b>9942</b>	<b>10495</b>	<b>10502</b>	<b>10500</b>	<b>10500</b>	<b>10493</b>	<b>10462</b>	<b>10454</b>	<b>10465</b>	<b>8506</b>	<b>5005</b>

由于总共有 10 个 EDMA TC，而只有 8 个 DSP 核，在这些测试里，8 个 TC 在 DDR 和 8 个核的 LL2 之间传输数据，另两个 TC 则在 DDR 和 SL2 之间传输数据。

从上面的测试结果可以看出，DDR 的带宽 (1333 x 8 = 10666MB/s) 对所有 EDMA 同时访问来说是不够的，EDMA TC 的优先级对它获得的带宽有明显影响。优先级低的 EDMA 传输得到的带宽比较小。对负载很高的情况下，一些低优先级的传输可能会被“饿死”，也就是说，得到的带宽为 0。

根据 Table 6 中列出的不同 EDMA TC 的区别，EDMACC1 TC1 和 TC3，EDMACC2 TC1 和 TC2 在某些情况下获得的带宽会比其它 TC 小，即使优先级相同。

从上面的测试结果还可以看出，多个 DMA 同时访问 **相同** DDR bank 中不同页的性能比多个 DMA 同时访问 **不同** DDR bank 中不同页的性能差很多，原因就是页切换的额外开销。当 DDR 负载加重时结果变得更差。最坏的情况下，页切换的额外开销会占用整体传输时间的绝大部分，从而使总吞吐量急剧下降。

页切换的概率，也就是，多个主模块访问 **相同** DDR bank 的概率取决于同时访问的主模块的个数和 DDR bank 个数。例如，4 个 DMA 随机访问 8 个 bank 的 DDR 存储器，至少两个 TC 访问相同的 bank 的概率是：

$$1 - C_8^4 P_4^4 / 8^4 = 59\%$$

表 13 列出了理论上，多个主模块访问相同 bank 的概率：



**表 13 多个主模块访问相同 bank 的概率**

	2 个主模块	4 个主模块	6 个主模块	8 个主模块	10 个主模块
8 banks	12.5%	59%	92.3%	99.7%	100%

为了减少 DDR 页切换次数，如果有多个传输，有的要访问已经打开的页，有的要访问没有打开的页，C6678 的 DDR 控制器一般情况下会让对已经打开的页的访问先执行。

## 6. 总结

总的来说，DSP 核可以高效地访问内部存储器，而用 DSP 核访问外部数据则不是有效利用资源的方式；IDMA 非常适用于 DSP 核本地存储器(L1D, L1P, LL2)内连续数据块的传输，但它不能访问共享存储器(SL2, DDR)；而外部存储器的访问则应尽量使用 EDMA。

为了充分利用 cache，DSP 核应尽量连续访问。

EDMA 的 ACNT 越大，效率越高。

SL2 有足够的带宽来支持所有核的同时访问。DDR 的带宽对所有核同时访问来说是不够的，DSP 核的优先级对它获得的带宽有明显影响。

DDR 性能受页切换的影响很大，为了减少 DDR 页切换应该尽量每次访问大块数据。

## 参考文献

1. *TMS320C66x DSP CorePac User Guide (SPRUGW0)*
2. *KeyStone Architecture Multicore Shared Memory Controller (MSMC) User Guide (SPRUGW7)*
3. *KeyStone Architecture DDR3 Memory Controller User Guide (SPRUGV8)*
4. *KeyStone Architecture Enhanced Direct Memory Access (EDMA3) Controller User Guide (SPRUGS5)*
5. *TMS320TCI6608 data manual (SPRS623)*

## 重要声明

德州仪器(TI) 及其下属子公司有权根据 JESD46 最新标准, 对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权根据 JESD48 最新标准中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的组件的性能符合产品销售时 TI 半导体产品销售条件与条款的适用规范。仅在 TI 保证的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非适用法律做出了硬性规定, 否则没有必要对每种组件的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用 TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何 TI 专利权、版权、屏蔽作品权或其它与使用了 TI 组件或服务的组合设备、机器或流程相关的 TI 知识产权中授予的直接或隐含权作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从 TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是 TI 的专利权或其它知识产权方面的许可。

对于 TI 的产品手册或数据表中 TI 信息的重要部分, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。TI 对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

在转售 TI 组件或服务时, 如果对该组件或服务参数的陈述与 TI 标明的参数相比存在差异或虚假成分, 则会失去相关 TI 组件或服务的所有明示或暗示授权, 且这是不正当的、欺诈性商业行为。TI 对任何此类虚假陈述均不承担任何责任或义务。

客户认可并同意, 尽管任何应用相关信息或支持仍可能由 TI 提供, 但他们将独力负责满足与其产品及其应用中使用的 TI 产品相关的所有法律、法规和安全相关要求。客户声明并同意, 他们具备制定与实施安全措施所需的全部专业技术和知识, 可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类安全关键应用中使用任何 TI 组件而对 TI 及其代理造成的任何损失。

在某些场合中, 为了推进安全相关应用有可能对 TI 组件进行特别的促销。TI 的目标是利用此类组件帮助客户设计和创立其特有的可满足适用的功能安全性标准和要求的终端产品解决方案。尽管如此, 此类组件仍然服从这些条款。

TI 组件未获得用于 FDA Class III (或类似的生命攸关医疗设备) 的授权许可, 除非各方授权官员已经达成了专门管控此类使用的特别协议。

只有那些 TI 特别注明属于军用等级或“增强型塑料”的 TI 组件才是设计或专门用于军事/航空应用或环境的。购买者认可并同意, 对并非指定面向军事或航空航天用途的 TI 组件进行军事或航空航天方面的应用, 其风险由客户单独承担, 并且由客户独力负责满足与此类使用相关的所有法律和法规要求。

TI 已明确指定符合 ISO/TS16949 要求的产品, 这些产品主要用于汽车。在任何情况下, 因使用非指定产品而无法达到 ISO/TS16949 要求, TI 不承担任何责任。

	产品		应用
数字音频	<a href="http://www.ti.com.cn/audio">www.ti.com.cn/audio</a>	通信与电信	<a href="http://www.ti.com.cn/telecom">www.ti.com.cn/telecom</a>
放大器和线性器件	<a href="http://www.ti.com.cn/amplifiers">www.ti.com.cn/amplifiers</a>	计算机及周边	<a href="http://www.ti.com.cn/computer">www.ti.com.cn/computer</a>
数据转换器	<a href="http://www.ti.com.cn/dataconverters">www.ti.com.cn/dataconverters</a>	消费电子	<a href="http://www.ti.com.cn/consumer-apps">www.ti.com.cn/consumer-apps</a>
DLP® 产品	<a href="http://www.dlp.com">www.dlp.com</a>	能源	<a href="http://www.ti.com.cn/energy">www.ti.com.cn/energy</a>
DSP - 数字信号处理器	<a href="http://www.ti.com.cn/dsp">www.ti.com.cn/dsp</a>	工业应用	<a href="http://www.ti.com.cn/industrial">www.ti.com.cn/industrial</a>
时钟和计时器	<a href="http://www.ti.com.cn/clockandtimers">www.ti.com.cn/clockandtimers</a>	医疗电子	<a href="http://www.ti.com.cn/medical">www.ti.com.cn/medical</a>
接口	<a href="http://www.ti.com.cn/interface">www.ti.com.cn/interface</a>	安防应用	<a href="http://www.ti.com.cn/security">www.ti.com.cn/security</a>
逻辑	<a href="http://www.ti.com.cn/logic">www.ti.com.cn/logic</a>	汽车电子	<a href="http://www.ti.com.cn/automotive">www.ti.com.cn/automotive</a>
电源管理	<a href="http://www.ti.com.cn/power">www.ti.com.cn/power</a>	视频和影像	<a href="http://www.ti.com.cn/video">www.ti.com.cn/video</a>
微控制器 (MCU)	<a href="http://www.ti.com.cn/microcontrollers">www.ti.com.cn/microcontrollers</a>		
RFID 系统	<a href="http://www.ti.com.cn/rfidsys">www.ti.com.cn/rfidsys</a>		
OMAP应用处理器	<a href="http://www.ti.com.cn/omap">www.ti.com.cn/omap</a>		
无线连通性	<a href="http://www.ti.com.cn/wirelessconnectivity">www.ti.com.cn/wirelessconnectivity</a>	德州仪器在线技术支持社区	<a href="http://www.deyisupport.com">www.deyisupport.com</a>

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122  
Copyright © 2013 德州仪器 半导体技术 (上海) 有限公司