

## **KeyStone DSP 上的 Multicore Navigator 的性能**

---

冯华亮/Brighton Feng

多核 DSP

### **摘要**

Multicore Navigator 是用于 KeyStone 系列 DSP 内部包交换的新型架构。本文讨论 Multicore Navigator 的性能，提供在各种条件下测试得到的性能数据，并讨论一些影响 Multicore Navigator 性能的因素。

## 目录

<b>1</b>	<b>Multicore Navigator 简介</b> .....	<b>3</b>
<b>2</b>	<b>QMSS 的性能</b> .....	<b>5</b>
2.1	PUSH 操作的开销.....	5
2.2	POP 操作的开销.....	6
2.3	通过不同区域访问队列.....	7
2.4	使用外部链接 RAM.....	8
2.5	队列挂起中断的时延.....	8
2.6	描述符累积的时延.....	9
2.7	描述符回收的时延.....	10
2.8	其它队列操作的性能考虑.....	11
<b>3</b>	<b>Packet DMA 的性能</b> .....	<b>11</b>
3.1	Packet DMA 传输的额外开销.....	11
3.2	Packet DMA 吞吐量.....	12
3.3	Packet DMA 和 EDMA 对比.....	13
	参考文献.....	14

## 图

图 1	Multicore Navigator 架构.....	3
图 2	QMSS Packet DMA 单通道吞吐量.....	12
图 3	QMSS Packet DMA 多通道总吞吐量.....	13
图 4	单通道 Packet DMA 与 EDMA 的比较.....	14

## 表

表 1	各种 PUSH 操作的开销.....	5
表 2	各种 POP 操作的开销.....	6
表 3	用外部链接 RAM 和内部链接 RAM 时 PUSH/POP 开销的比较.....	8
表 4	平均描述符累积时延.....	10

# 1 Multicore Navigator 简介

Multicore Navigator 包括 Queue Manager Subsystem (QMSS)和 Packet DMA (PKTDMA)，用它们可实现在器件内部高效的包交换。这大大降低了 DSP 核在内部通信方面的负担，从而提高了系统的整体性能。

图 1 是 KeyStone 系列 DSP 的 Multicore Navigator 的架构框图。

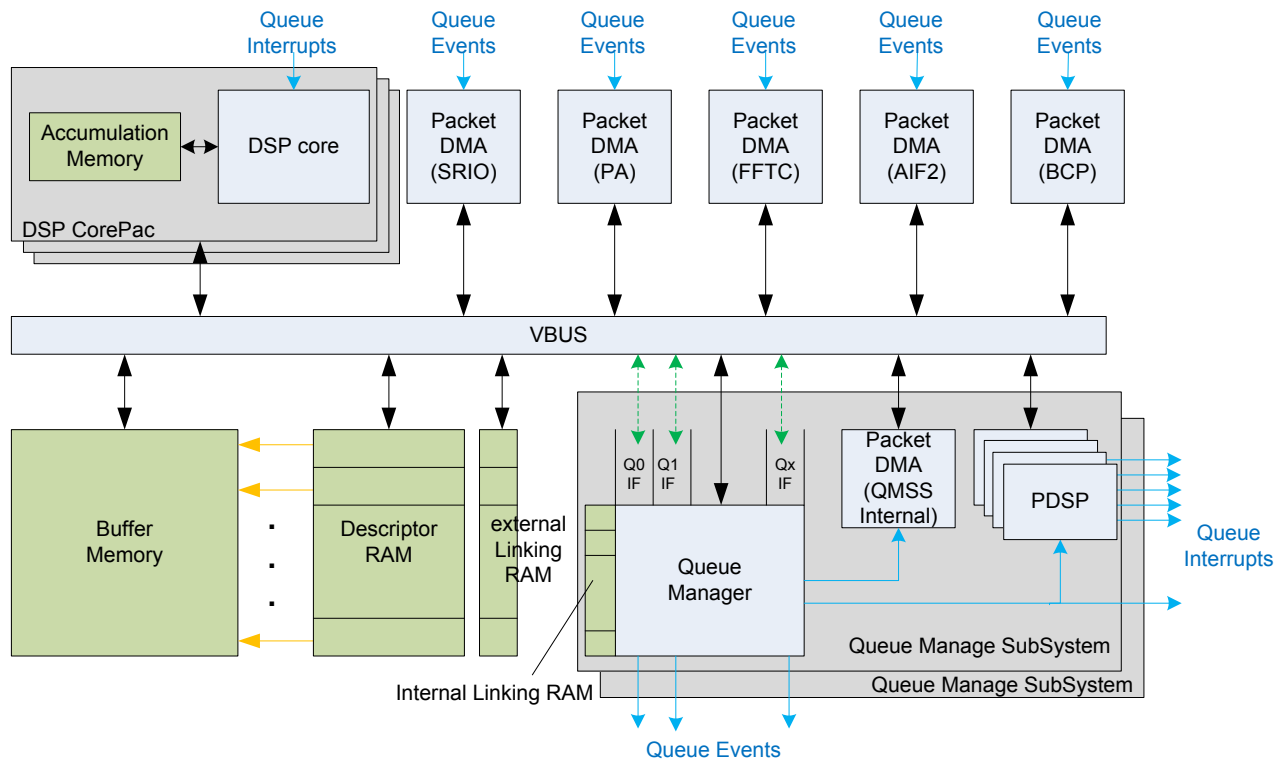


图 1 Multicore Navigator 架构

硬件队列是 Multicore Navigator 的基础，KeyStone 系列中不同器件支持的硬件队列的个数可能不一样，有的是 8192 个，有的是 16384 个。队列管理器管理这些队列，提供基本的操作包括 PUSH，POP 等。有的器件包含一个队列管理器，有的器件包含 2 个队列管理器。

队列管理器维护的关键数据结构是一个链接表。每个链接表项占用 64 比特，它主要用来表示队列中一个包的链接信息，即当前包的下一个包的指针。每个链接表项和一个包描述符一一对应。

典型的队列 PUSH 操作过程如下：

1. 系统中一个主模块把一个包描述符的地址写到一个队列对应的操作寄存器，这实际上给队列管理器产生一个 PUSH 请求。
2. 队列管理器读取这个队列的尾指针找到队列中的最后一个包描述符的链接表项。

3. 队列管理器修改最后一个包描述符的链接表项，让它指向新的被 PUSH 进来的包描述符的链接表项。
4. 队列管理器修改尾指针，让它指向新 PUSH 进来的包描述符的链接表项。
5. 队列管理器修改新包描述符的链接表项为空。

典型的队列 POP 操作过程如下：

1. 系统中一个主模块读一个队列对应的操作寄存器，这实际上给队列管理器产生一个 POP 请求。
2. 队列管理器读取该队列的头指针，获取队列中的第一个包描述符，把第一个包描述符的地址返回给发起读操作的主模块。
3. 队列管理器读取第一个包描述符的链接表项，找到队列中的下一个描述符的链接表项。
4. 队列管理器修改队列的头指针，让它指向队列中的第二个描述符的链接表项，从而让它变成了第一个描述符。

QMSS 中包含一个内部的链接 RAM，不同器件的链接 RAM 大小可能不一样，有的支持 16K 个链接表项，有的支持 32K 个链接表项。如果用户系统中需要的包描述符超过 16K 或 32K 个，则可以用其它的 RAM，包括 LL2 (Local Level 2 memory), SL2 (Shared Level 2 memory), DDR (Double Data Rate external memory)来存放更多的链接表项，这被叫做 QMSS 的外部链接 RAM。

对主模块而言 PUSH 是一个写操作，仅需要几个周期，通常不会让主模块停等；而 POP 操作对主模块而言是读操作，通常需要等待队列管理器的返回值。

为了解决 DSP 核在 POP 操作时停等时间长的问题，QMSS 内集成了若干个微控制器（PDSP，不同器件内集成的个数可能不一样）。用户可配置 PDSP，让它监测某些队列，当队列中有包描述符时，PDSP 把它 POP 出来，把包描述符的指针写到一个累积缓冲区中。累积缓冲区的位置和大小可配，通常，累积缓冲区在 DSP 核的 LL2 中。当 PDSP 填满累积缓冲区时可以给对应的 DSP 核产生一个中断事件，DSP 核在中断服务程序中读取累积缓冲区中的包描述符并处理对应的包。由于包描述符被累积到了 DSP 核的 LL2 中，DSP 核读取它们的时间大大的减少了。

KeyStone 系列的器件中，可访问硬件队列的主模块包括：

- 所有 DSP 核
- 所有包含 Packet DMA 的主模块
  - QMSS (Queue Manager Subsystem)
  - SRIO (Serial RapidIO)
  - PA (Ethernet Packet Accelerator)
  - FFTC (FFT Coprocessor, 仅部分器件支持)
  - AIF2 (Antenna Interface 2, 仅部分器件支持)

■ BCP (Bit CoProcessor, 仅部分器件支持)

Packet DMA 是专门用来做基于包的数据传输的 DMA (Direct Memory Access)引擎。传统的 EDMA 的传输请求通过参数表(Parameter RAM)来定义；而 Packet DMA 传输请求由包描述符定义，而包描述符可以挂到某个硬件队列上。另外，EDMA 支持最大 3 维的数据块传输，而且数据块之间的偏移可配；而 Packet DMA 仅支持 1 维线性数据块传输。

本文讨论 QMSS 和 Packet DMA 的性能，提供在各种条件下测试得到的性能数据，并讨论一些影响 Multicore Navigator 性能的因素。

如果没有特殊说明，本文中的性能数据是在 1GHz 的 C6678 评估板上的实测结果。评估板上的 DDR 是 1333MTS 64-bit 位宽。同系列的其它器件性能可能会稍有差别，但应该基本类似。

## 2 QMSS 的性能

QMSS 的主要性能指标包括 PUSH, POP 操作开销，队列挂起中断的时延，描述符累积的时延，描述符回收的时延。

### 2.1 PUSH 操作的开销

下面是 PUSH 性能测试的伪代码：

```
startTSC= TimeStampCount;
for(i=0; i< Number_of_Descriptors; i++)
{
    queueRegs->REG_D_Descriptor= uiDescriptor[i]; //PUSH
}
AverageCycles= (TimeStampCount - startTSC)/ Number_of_Descriptors;
```

表 1 是在 C6678 上的测试结果。

表 1 各种 PUSH 操作的开销

number of descriptors	通过不同区域 PUSH 消耗的 DSP 核时钟周期数	
	队列管理寄存器空间	队列管理 VBUSM 空间
512	15	14
256	15	13
128	14	11
64	14	7
32	12	1
16	10	1
8	6	1
4	7	1

2	7	1
1	6	1

根据以上测试结果，一个 PUSH 操作在不同情况下可能消耗 1 到 15 个 DSP 核时钟周期。

PUSH 操作对主模块来说实际上是写操作，DSP 核可以在把写的数据丢给写缓冲区后就立即返回。在写缓冲区满之前，DSP 核不会停等，这就是只 PUSH 几个描述符的情况；

而当很多描述符被一次性 PUSH 时，在写缓冲区满了之后，DSP 核需要停等，直到之前写的数据被队列管理器处理而腾出了缓冲区空间为止。在这种情况下，DSP 核每次 PUSH 停等的时间实际上就是队列管理器处理一个 PUSH 操作的时间，根据以上测试，它大概是 15 个 DSP 核时钟周期。

由此我们可以得出一个结论，如果一个主模块连续以大于 15 个 DSP 核时钟周期的速度做 PUSH 操作，它不会为 PUSH 操作停等；否则，它可能要停等 1 到 15 个 DSP 核时钟周期。

## 2.2 POP 操作的开销

下面是 POP 性能测试的伪代码：

```

startTSC= TimeStampCount;
for(i=0; i< Number_of_Descriptors; i++)
{
    uiDescriptor[i]= queueRegs->REG_D_Descriptor; //POP
}
AverageCycles= (TimeStampCount - startTSC)/ Number_of_Descriptors;
    
```

表 2 是在 C6678 上的测试结果。

表 2 各种 POP 操作的开销

number of descriptors	通过不同区域 POP 消耗的 DSP 核时钟周期数	
	队列管理寄存器空间	队列管理 VBUSM 空间
512	45	87
256	45	87
128	45	87
64	45	87
32	45	87
16	45	87
8	45	87
4	46	87
2	47	88
1	47	88

根据以上测试结果，一个 POP 操作至少需要 45 个 DSP 核时钟周期。POP 消耗的周期数比 PUSH 大很多，因为 POP 实际上是读操作，主模块必须等队列管理器返回数据。

如果有多个包描述符需要 POP，累积器可以被用来把包描述符从硬件队列中“累积”到 DSP 核的本地存储器，而 DSP 核再从它的本地存储器中读取包描述符，这样读一个描述符只需要消耗 DSP 核大概 5 个时钟周期，节约了至少 40 个时钟周期。

## 2.3 通过不同区域访问队列

队列管理器提供了多个区域（或者说是窗口）供主模块访问硬件队列。它们包括：

1. 通过 VBUSP 配置总线的寄存器类型的访问 (仅 DSP 核支持)
2. 通过 VBUSM 数据总线的数据类型的访问区域

它们被映射到不同地址，从主模块的角度说，访问不同的区域的区别仅仅是用不同的地址访问。

VBUSP 配置总线和 VBUSM 数据总线不同，只有 DSP 核可以用 VBUSP 总线，所有的主模块都可以用 VBUSM 总线。

根据以上测试结果，通过 VBUSM 总线 PUSH 比通过 VBUSP 停等的时间少，因为通过 VBUSM 的写缓冲区更深；但通过 VBUSP 总线 POP 比较快，因为 VBUSP 的读时延较小。

如果没有特殊说明，在本文提到的所有测试中，DSP 核都是通过 VBUSM 区域 PUSH，而通过 VBUSP 区域 POP。

如果 DSP 核往一个队列里 PUSH 包描述符，而另一个 Packet DMA 从这个硬件队列里 POP 包描述符，DSP 核应该用 VBUSM 区域来 PUSH 包描述符，这样可以避免潜在的竞争风险，例如，对以下顺序的操作：

1. DSP 核通过 VBUSM 往位于 DDR 的包缓冲区写数据，
2. DSP 核把相应的包描述符通过 VBUSP PUSH 给 Packet DMA，
3. Packet DMA POP 包描述符，
4. Packet DMA 从位于 DDR 的包缓冲区中读取数据。

由于 DSP 核通过不同的总线写数据和包描述符，有可能包描述符比数据先到达，而导致 Packet DMA 读到包描述符时数据还没有更新。如果 DSP 核通过相同的 VBUSM 总线写数据和包描述符，就可以避免这种问题。

另外一个避免这种竞争风险的方法是在 PUSH 包描述符之前使用 MFENCE 指令来确保写的已经到达目的地。关于 MFENCE 指令的详细信息，请参阅“TMS320CC66x CPU and Instruction Set Reference Guide (sprugh7)”。

## 2.4 使用外部链接 RAM

QMSS 中包含一个内部的链接 RAM，如果用户系统中需要的包描述符超过内部链接 RAM 支持的数目，则可以用其它的 RAM，包括 LL2 (Local Level 2 memory), SL2 (Shared Level 2 memory), DDR (Double Data Rate external memory) 来存放更多的链接表项。

队列管理器访问外部链接 RAM 的开销比访问它的内部链接 RAM 的开销大。通常，用 DDR 做外部链接 RAM 的效率很低，因为 DDR 适合大块连续的访问，而链接表项的访问比较离散；用 LL2 的性能比较好，但 LL2 相对较小；所以一般用 SL2 比较合适。在本文提到的测试中，都是用 SL2 做为外部链接 RAM。

表 3 比较了用外部链接 RAM 和用内部链接 RAM 时 DSP 核做 PUSH/POP 操作消耗的时钟周期数。

**表 3 用外部链接 RAM 和内部链接 RAM 时 PUSH/POP 开销的比较**

number of descriptors	PUSH		POP	
	内部链接 RAM	外部链接 RAM	内部链接 RAM	外部链接 RAM
512	14	14	45	100
256	13	13	45	100
128	11	11	45	99
64	7	7	45	99
32	1	1	45	98
16	1	1	45	96
8	1	1	45	92
4	1	1	46	86
2	1	1	47	74
1	1	1	50	48

根据以上测试结果，对 PUSH 操作而言，我们看不出用外部链接 RAM 的明显区别；而对 POP 操作，用外部链接 RAM 会增加大概 50 个时钟周期的开销。和前面章节介绍的一样，累积器可以用来帮 DSP 核节省 POP 操作的开销。

在本文中，如无特殊说明，测试都是用的内部链接 RAM。

## 2.5 队列挂起中断的时延

队列管理器可以监测一些硬件队列，如果它们非空，则可以给其它主模块产生一个队列挂起的中断。下面是队列挂起中断的时延测试的伪代码：

```

.....
startTSC= TimeStampCount;
queueRegs->REG_D_Descriptor= uiDescriptor; //push to an empty queue
asm(" IDLE"); //wait for the queue pending interrupt
delay= intTSC - startTSC;
.....

```



```

interrupt void QueuePendISR()    //queue pending Interrupt Service Routine
{
    intTSC= TimeStampCount;    //save the Time Stamp Count when the interrupt happens
    .....
}

```

在 C6678 评估板上测得的时延大约是 **130** 个 DSP 核时钟周期。

## 2.6 描述符累积的时延

为了解决 DSP 核在 POP 操作时停等时间长的问题，QMSS 内集成了若干个微控制器（PDSP，不同器件内集成的个数可能不一样）。用户可配置 PDSP，让它监测某些队列，当队列中有包描述符时，PDSP 把它 POP 出来，把包描述符的指针写到一个累积缓冲区中。累积缓冲区的位置和大小可配，通常把累积缓冲区放在 DSP 核的 LL2 中。当 PDSP 填满累积缓冲区时可以给对应的 DSP 核产生一个中断事件，DSP 核在中断服务程序中读取累积缓冲区中的包描述符并处理对应的包。由于包描述符被累积到了 DSP 核的 LL2 中，根据前面的测试结果，DSP 核读取它们的时间大概会节省 40 个时钟周期。

PDSP 中加载不同的固件时，它支持的功能不一样。Acc48 固件监测最多 32 个高优先级队列，并且监测最多 512 (16x32) 个低优先级队列；Acc32 固件监测最多 32 个队列；Acc16 固件最多监测 512 (16x32) 个队列。

从包描述符 PUSH 到被监测队列到累积中断完成之间的时延的大小取决于 PDSP 监测的队列个数，以及这些队列的繁忙程度。本文介绍的测试是最简单的情况，即，PDSP 仅监测一个队列，而且仅测量累积一个包描述符的时延。

下面是描述符累积的时延测试的伪代码：

```

Setup ISR(Interrupt Service Routine) for Accumulation interrupt
Setup accumulation function of PDSP
.....
startTSC= TimeStampCount;
queueRegs->REG_D_Descriptor= uiDescriptor; //push to an empty queue
asm(" IDLE"); //wait for the queue pending interrupt
delay= intTSC - startTSC;
.....
interrupt void QueueAccumulationISR ()    //accumulation Interrupt Service Routine
{
    intTSC= TimeStampCount;    //save the Time Stamp Count when the interrupt happens
    .....
}

```

表 4 是在 C6678 上多次测量的平均值。

表 4 平均描述符累积时延

Firmware		Cycles
Acc 48	High priority channel	2953
	Low priority channel	7875
Acc 32		2841
Acc 16		1862

这个时延看起来比较大，但在 PDSP 累积包描述符时，DSP 核可以做其它工作。所以，这个方法比较适合于大量的对时延不敏感的包的处理。而对时延要求很紧的包，我们最好用查询方式或队列触发中断方式来监测它，而不要用累积器。

## 2.7 描述符回收的时延

通常，用 DSP 核软件回收一个包描述符的过程是：

- 1, 解析包描述符中的“return queue number”，“return policy”和“return push policy”域，
- 2, 把包描述符 PUSH 到解析出来的“return queue”。

为了节省 DSP 核软件的开销，PDSP 提供了描述符回收的功能，可以省掉上述 DSP 核的第 1 步操作。

使用 PDSP 的描述符回收功能时，DSP 核软件只需要把包描述符 PUSH 到 PDSP 监测的一个队列就可以了，PDSP 监测的回收队列是可选的，当有任何包描述符进入这个回收队列时，PDSP 会根据包描述符里的“return queue number”，“return policy”和“return push policy”域的配置把这个包描述符 PUSH 到相应的队列。

由于这个回收功能是由 PDSP 固件实现的，一个包描述符从被 PUSH 到回收队列到 PDSP 把它返回到最终的空闲队列的时延主要由 PDSP 的繁忙程度决定。

我们仅测试最简单的情况，即 PDSP 仅做描述符回收一件事。下面是描述符回收的时延测试的伪代码：

```

Setup reclamation function of PDSP
.....
startTSC= TimeStampCount;
queueRegs->REG_D_Descriptor= uiDescriptor;    //push used descriptor to reclamation queue
wait/poll the descriptor in the FDQ (destination queue)
delay= TimeStampCount - startTSC;

```

这种情况下在 C6678 上测得的时延大概是 **900** 个时钟周期。

这个方法适合于回收大量包描述符，而且对回收时间没有严格要求。如果包描述符需要很快被回收再用，那还是用常规的方法比较保险。

## 2.8 其它队列操作的性能考虑

DSP 核写读全部包描述符内容需要消耗比较大的时钟周期。对大部分应用，DSP 核可以在初始化阶段给描述符所有域赋初值，而运行时仅写读很少的域（如包大小）。

由于 host 类型的包的描述符和包缓冲区是可以分开的，通常它们在存储器中不是连续存放的。分别访问描述符和包缓冲区引入了额外的开销，尤其是当它们在可 cache 空间时（需要做 cache 一致性维护）。而 monolithic 类型的包的描述符和包缓冲区是在一起的。总的来说，host 类型的包提供了比较好的灵活性，但 monolithic 类型的包在存储器访问性能方面更好。用户需要综合考虑这些因素来选择合适的包类型。

如果包描述符是在可 cache 的 SL2 或 DDR 存储器空间，软件往往需要维护它的 cache 一致性。Host 类型的包描述符通常只有 32 到 64 字节，往往不能充分利用 64 字节的 L1D cache 或 128 字节的 L2 cache 行，为了比较小的包描述符来做 cache 一致性维护往往不划算。因此，对比较小的包描述符，可以尽量把它们放到 LL2 存储器，这样就不用做 cache 一致性维护了。

## 3 Packet DMA 的性能

Packet DMA 的性能是在环回模式下测得的，也就是说，发送的包被环回到接收端。下面是 Packet DMA 性能测试的伪代码：

```

Setup Packet DMA in loopback mode
.....
for(different_packet_size)
{
    Prepare packets for transfer;
    startTSC= TimeStampCount;
    for(number_of_channels)
        push descriptor of a packet to the TX queue of the channel;
    wait/poll the packets in the RX queue;
    delay= TimeStampCount - startTSC;
    throughput=total_data_size/delay;
}
    
```

### 3.1 Packet DMA 传输的额外开销

本文中，Packet DMA 传输的额外开销被定义为传输最小单元（1 个字）所需的时间，即从包被 PUSH 到发送队列到从接收队列读出包之间的时间。

在 C6678 上测得的 Packet DMA 传输的额外开销大概是 600 个时钟周期。

传输额外开销对小包传输来说是个大问题。对于小包传输，用户需要综合考虑来决定使用 DMA 还是 DSP 核直接拷贝。

### 3.2 Packet DMA 吞吐量

图 2 是在 1GHz C6678 的 QMSS Packet DMA 上测得的单通道传输不同大小的包的吞吐量。

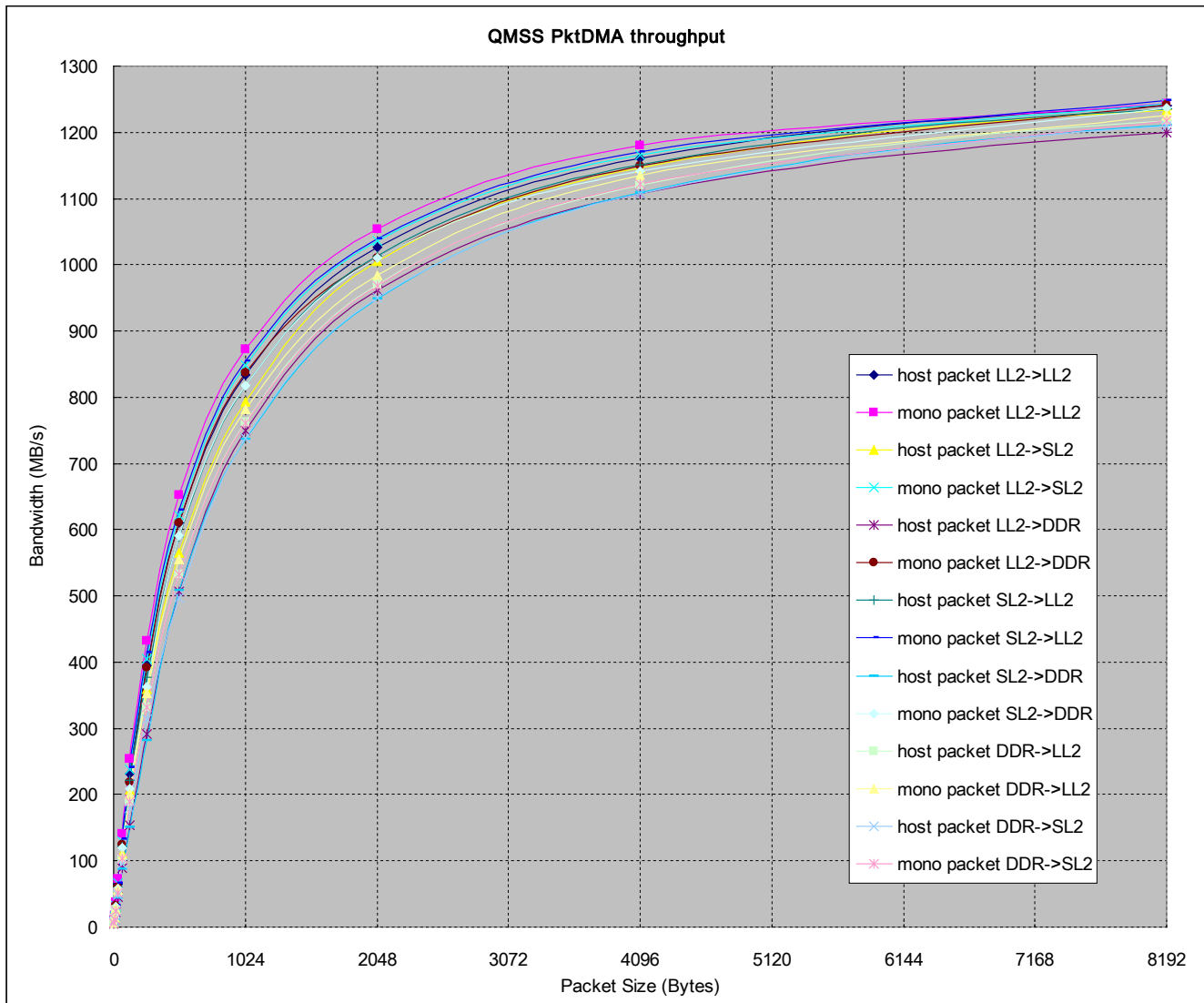


图 2 QMSS Packet DMA 单通道吞吐量

由于额外开销固定，包越大，带宽利用率就越高。

包缓冲区的位置对吞吐量的影响不大，Packet DMA 访问 LL2 比 SL2 稍快一点，访问 SL2 比 DDR 稍快一点。

包类型对吞吐量的影响也不大，monolithic 类型的包稍好一点。

Packet DMA 有全双工的 128 比特的总线，速率是 DSP 核的 1/3，在 1GHz DSP 上，Packet DMA 的理论带宽是  $128/8 \times 1000/3 = 5333\text{MB/s}$ 。以上测试得到的吞吐量远小于理论带宽，这是因为这个测试中只用了一个通道。

Packet DMA 支持多通道，通道之间基于优先级进行调度。由于 Packet DMA 需要周期性的轮询每一个通道，如果只有一个通道传输数据是不能充分利用带宽的，因为 Packet DMA 在轮询没有使用的通道时总线可能会空闲。使用的通道数越多，则总线带宽的利用率越高。

图 3 是在 1GHz C6678 的 QMSS Packet DMA 上测得的同时使用多个通道时测得的总吞吐量。

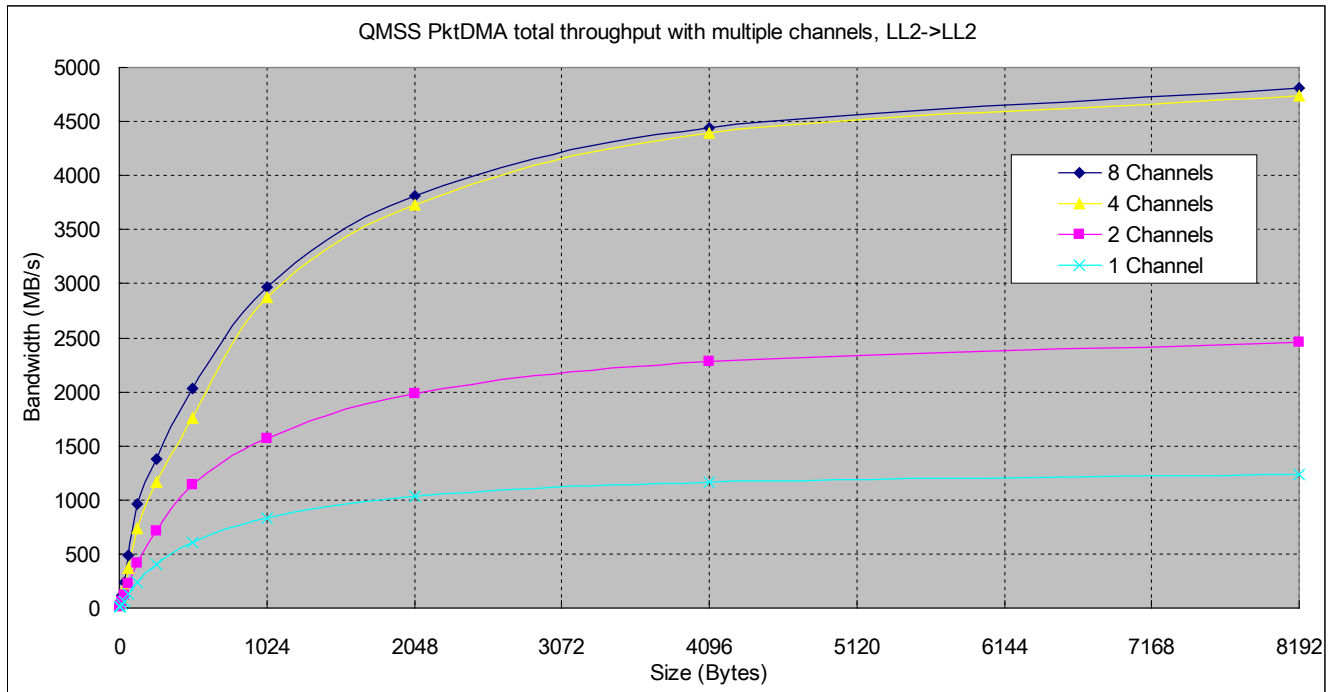


图 3 QMSS Packet DMA 多通道总吞吐量

以上测试结果说明 4 个通道就可以充分利用总线带宽。而通道数更多时，总吞吐量受总线带宽的限制。

### 3.3 Packet DMA 和 EDMA 对比

KeyStone 系列 DSP 中有多个 Packet DMA 传输引擎，这些 Packet DMA 引擎并不完全相同。

图 4 比较了 1GHz C6678 上不同 Packet DMA 通道和一个 EDMA 通道在两个 DSP 核的 LL2 之间传输数据的吞吐量。

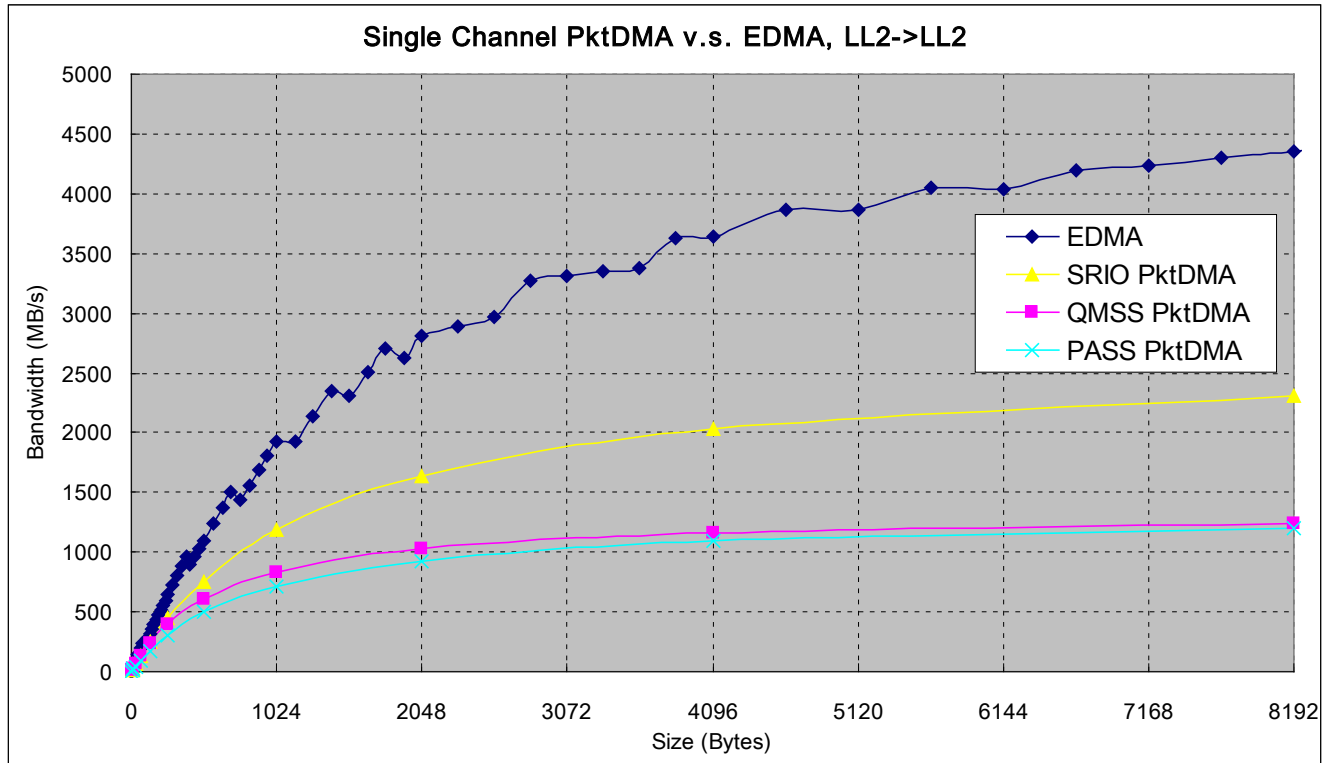


图 4 单通道 Packet DMA 与 EDMA 的比较

上图说明，一个 QMSS Packet DMA 的吞吐量和一个 PASS (Packet Accelerator SubSystem) Packet DMA 的吞吐量类似；它们大概是一个 SRIO Packet DMA 通道吞吐量的一半。而这些 Packet DMA 通道的吞吐量比一个 EDMA 通道的吞吐量小很多，一个 EDMA 通道就可以充分利用总线的带宽。

总的来说，Packet DMA 提供了灵活的包传输方式，它的吞吐量也应该能满足大部分应用的需求。Packet DMA 单通道的吞吐量不能充分利用总线和存储器系统的带宽。对于需要巨大吞吐量的应用，EDMA 可能是比较好的选择。

## 参考文献

1. *KeyStone Architecture Multicore Navigator User Guide (SPRUGR9)*
2. *TMS320C66x DSP CorePac User Guide (SPRUGW0)*
3. *KeyStone Architecture Enhanced Direct Memory Access (EDMA3) Controller User Guide (SPRUGS5)*
4. *TMS320CC66x CPU and Instruction Set Reference Guide (sprugh7)*

## 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