Application Report
ZHCA587 – November 2013

KeyStone 系列 DSP 的存储器测试

冯华亮/Brighton Feng 多核 DSP

摘要

存储器相关的问题是 DSP 应用中非常普遍的问题。本文介绍 KeyStone I 系列 DSP 上一些存储器测试的方法。

目录

图

表

1 KeyStone DSP 存储器系统简介

KeyStone DSP 存储器架构如图 1 所示。

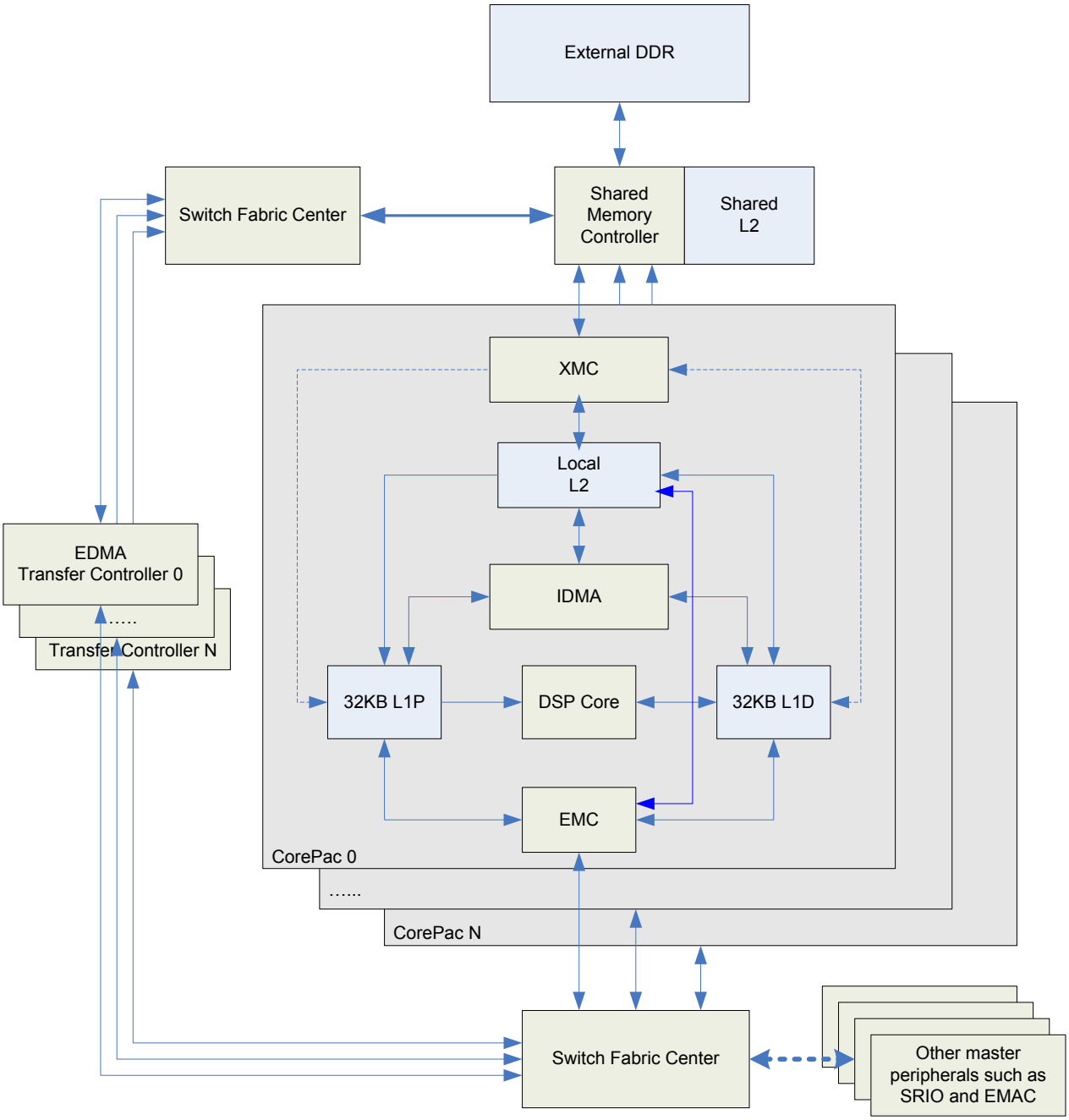

图 1 KeyStone DSP 存储器架构

对不同的 DSP，存储器的大小可能不同，DSP 核和 EDMA 传输控制器的个数也可能不同。表 1 比较了 KeyStone I 系列中常用的 3 颗 DSP。

表 1 KeyStone I 存储器系统比较

	C6678	C6670	TCI6614
L1D(KB/核)	32	32	32
L1P(KB/核)	32	32	32
Local L2 (KB/核)	512	1024	1024
Shared L2 (KB)	4096	2048	2048
DSP 核个数	8	4	4
EDMA 传输控制器个数	10	10	10

2 存储器测试算法

本文介绍几种存储器测试算法，并讨论这几种算法的用途。

2.1 数据测试

下面是数据测试的伪代码：

```
for(memory range under test)
 fill the memory with a value;
for(memory range under test)
 read back the memory and compare the readback value to the written value
```

通常，这个测试会被执行几次，每次填充的值不一样。常用的填充值包括 0x55555555, 0xAAAAAAAA, 0x33333333, 0xCCCCCCCC, 0x0F0F0F0F, 0xF0F0F0F0, 0x00FF00FF, 0xFF00FF00FF00, 0xFFFFFFFF, 0。

这个测试可以用来检测数据比特粘连(bit-stuck)问题，例如，如果，

written value = 0, readback value = 0x8,

表示 bit 3 粘连到 1.

如果

written value = 0xFFFFFFFF, readback value = 0xFFFFFFFE,

表示 bit 0 粘连到 0.

如果能正确的写入并读出 0x55555555（或 0xAAAAAAAA），说明相邻的两个比特没有粘连；如果能正确写入并读出 0x33333333（或 0xCCCCCCCC），说明相邻的 4 个比特没有粘连；如果能正确写入并读出 0x0F0F0F0F（或 0xF0F0F0F0），说明相邻的 8 个比特没有粘连…

这个算法既可以用来测试数据总线连接，也可以用于测试存储器单元。当用于测试存储器单元时则每一个存储单元都需要写读所有的值，这将是比较耗时的测试；而用于测试数据总线连接时，只需要把所有的值都写读一遍就可以了（地址不限）。

2.2 地址测试

地址测试的伪代码如下:

```
for(memory range under test)
 fill each memory unit with its address value;
for(memory range under test)
 read back the memory and compare the readback value to the written value
```

这个测试可以用来检测地址比特粘连(bit-stuck)问题。例如，如果

written value = 0 at address 0

written value = 1 at address 1

written value = 2 at address 2

written value = 3 at address 3

……

readback value = 2 at address 0

readback value = 3 at address 1

readback value = 2 at address 2

readback value = 3 at address 3

……

则说明地址线的比特 1 粘连，因为地址 0，2 中的数据相同，地址 1，3 中的数据相同。

这个测试的主要目的是测试地址线和存储器中的地址译码单元，但它实际上对所有存储单元都做了数据写读，所以在一定程度上也测试了数据总线和存储单元。如果由于测试时间的限制，只允许对整个存储器空间进行一遍写读测试时，本节介绍的地址测试是首选。

2.3 走比特测试

走比特测试包括走“1”和走“0”测试。走比特测试即可测试数据，也可以测试地址。

走“1”指测试的数据或地址中只有一个比特为“1”，而所有其它比特为“0”，而且连续的访问中每一次“1”的位置都会移动一个比特，看起来好像是“1”在总线上走。而走“0”测试只是把测试的数据反了一下，看起来就像是一个“0”在总线上走。

2.3.1 数据走比特

数据走比特测试的伪代码如下：

```
bit_mask_value = 1;
for(number of valid data bits)
{
 write bit_mask_value to memory;
 readback the data and compared to written value;

 inversed_bit_mask_value = bit inverse of bit_mask_value;
 write inversed_bit_mask_value to memory;
 readback the data and compared to written value;

 bit_mask_value = bit_mask_value<<1;   //bit walking
}
```

这个测试可以检测数据比特粘连(bit-stuck)问题，更重要的是，它还可以检测比特间串扰。如果比特之间存在串扰，而这个测试让一个比特与所有其它比特都是相反的，那么其它比特对它的干扰会被最大化，从而让问题暴露出来。

例如，如果

written value = 0x00000010, readback value = 0

written value = 0xFFFFFFEF, readback value = 0xFFFFFFFF

这往往说明比特 4 被其它比特干扰。

这个算法既可以用来测试数据总线连接，也可以用于测试存储器单元。当用于测试存储器单元时则每一个存储单元都需要走“1”走“0”，这将是最耗时的测试；而用于测试数据总线连接时，只需要走一遍“1”，再走一遍“0”就可以了（地址不限）。

2.3.2 地址线走比特

地址走比特用于测试地址总线的连接。地址线走比特测试的伪代码如下：

```
Write 0 to first address;
Write 0xFFFFFFFF to last address;
```

```
bit_mask_value = 1;
for(number of valid address bits)
{
 write bit_mask_value to address of bit_mask_value;
 readback the data at first address and compare to 0;

 inversed_bit_mask_value = bit inverse of bit_mask_value;
 write inversed_bit_mask_value to address of inversed_bit_mask_value;
 readback the data at last address and compare to 0xFFFFFFFF;

 bit_mask_value = bit_mask_value<<1;   //bit walking
}
```

和数据走比特类似，这个测试既可以检测地址比特粘连(bit-stuck)问题，还可以检测地址比特间串扰。

例如，如果

written value = 16 at address 16

readback value = 16 at address 0

written value = 0xFFFFFFEF at address 0xFFFFFFEF

readback value = 0xFFFFFFEF at address 0xFFFFFFFF

测试中，想要写往地址 16 的数据实际被写到地址 0；而想要写往地址 0xFFFFFFEF 的数据实际被写到最后一个地址单元。这往往说明地址比特 4 被其它比特干扰。

2.4 浮动总线问题

如果测试软件写入并很快从相同地址读出一个值的时候，如果数据线上存在电容特性，写操作会给数据线上的电容充电，总线会短暂的保持它的状态。当测试软件读操作时，总线会返回刚写入的值，即使实际上该数据并没有正确地被写入存储单元。这就是浮动总线（floating buses）问题。

浮动总线可能会“欺骗”简单的测试程序，为了规避浮动总线问题，需要在紧邻的对相同地址的写和读操作之间对其它地址写入一个和原来写的数据相反的数据。例如，

```
write A to address X;
write inversion of A to address Y;
read value from address X;
```

这样，浮动总线的问题就可以规避了。

2.5 测试结果的深入分析

从存储器测试结果的初步分析中，如果我们发现比特粘连或干扰，还需要进一步深入分析原因。通常原因可能来自于三方面：

1. 对于外接存储器，PCB 出问题的可能性比较大。最常见的包括焊接问题或设计问题。例如，某个比特被短接到电源或地。通常我们可以用万用表测量信号线之间或信号线和电源或地之间的阻抗来定位这种问题。串扰问题的定位则比较复杂，可能需要用示波器来测试所有相关的信号来确定串扰源。

2. 存储单元失效。如果是外接存储器，我们可以用示波器或逻辑分析仪在总线上监测写入和读出的数据，如果总线上监测到的写数据是对的，而读出的数据是错的，则往往是存储单元失效。

3. 存储控制器失效。如果我们排除了以上问题而怀疑存储控制器时，可以把好的板子和坏的板子上的控制器互换，如果问题跟着控制器走，则往往说明是控制器失效。

3 存储器测试覆盖率

完善的存储器测试应该涵盖：

- 系统中所有的存储器
- 所有可以访问存储器的模块，包括 DSP 核和 DMA 传输控制器
- 所有不同的访问路径

3.1 用不同的主模块测试

如图 1 所示，主要有三个主模块可以访问存储器：

- DSP 核可以访问所有存储器
- EDMA 可以访问所有存储器
- IDMA 只可以访问每个核本地的 L1 和 L2 存储器

理论上，我们应该用所有的主模块测试所有可访问的存储器，但这样有太多组合，很多重复的测试可能浪费很多时间。所以，测试的组合需要被优化，既保证所有主模块和存储器都被覆盖，又要避免无用的重复测试。

通常 DMA 访问存储器的效率比 DSP 核高很多，所以，对外部大存储器的测试优先采用 EDMA。

用 DMA 测试存储器是，我们需要一个额外的缓冲区 (buffer)，测试流程如下:

DSP core wirtes test data to the DMA buffer
DMA copy the contents of the DMA buffer to the memory under test
DMA copy the contents of the memory under test back to the DMA buffer
DSP core reads the contents in DMA buffer and compare to expected value.

通常，用于 DMA 测试的缓冲区是一个位于 L2 中的比较小的缓冲区，而被测试的存储器往往要大得多，所以上述测试流程需要被重复很多遍才能完成对整个存储器的测试。

EDMA 有多个传输控制器，它们可以被交替使用来保证它们都被用到。例如，我们可以用 EDMA TC1 来填充并回读第一组数据，然后用 EDMA TC2 来填充并回读第二组数据…

请注意，在多核 DSP 中，EDMA 必须用全局地址来访问 L1 和 LL2 存储器。

3.2 测试系统中所有的存储器

如图 1 所示，Keystone 系统中主要有 4 种类型的存储器，它们都应该被测试到。

3.2.1 内部 L1 存储器

通常，我们建议用 IDMA 测试内部 L1 存储器。测试前，L1 存储器应该被配置成普通存储器而不是 cache。

DSP 核可以测试本核的 L1D 存储器，但 DSP 核不能测试本核的 L1P 存储器。

在多核 DSP 上，如果我们想在一个核上运行程序来测试另一个核的 L1 存储器，我们应该用 EDMA 而不能用 IDMA，因为 IDMA 只能访问本核的存储器。在对其它核的存储器测试时，其它核不能运行程序。

3.2.2 内部 LL2 存储器

LL2 可以被系统中所有的主模块测试。根据通常的使用情况，我们可以用 DSP 核和 EDMA 来测试本地 LL2 存储器。

通常，存储器测试代码和数据缓冲区都在本地 LL2 存储器。所以，测试程序应该检测被测试代码和测试缓冲区占用的空间大小，而仅对剩下的空闲空间进行测试。

尽管被测试代码和测试缓冲区占用的存储区没有被严格的测试算法测试，通常，如果这个区域有问题，测试程序往往会失败。所以，从一定程度上说，这块存储器也是被覆盖了的。如果真的需要用严格的测试算法来测试所有的空间的话，我们需要两个测试程序，一个测试程序占用存储器前半部分，而测试后半部分；另一个测试程序占用存储器后半部分，而测试前半部分。

在多核 DSP 上，如果我们想在一个核上运行程序来测试另一个核的 LL2 存储器，EDMA 和 DSP 核都可以用来做测试。在对其它核的存储器测试时，其它核不能运行程序。

3.2.3 内部共享 SL2 存储器

每个 DSP 核和 DMA 都有单独的总线访问共享 SL2 存储器，测试应该涵盖每条总线，通过每个 DSP 核和 DMA 来测试 SL2 存储器。

DSP 核通过缺省地址空间（从 0x0C000000 开始）访问 SL2 会经过 cache 和 prefetch buffer。要绕过 cache 和 prefetch buffer 访问 SL2，必须把 SL2 重新映射到其它地址空间，并把这个地址空间设置（通过 MAR 寄存器）成不可 cache，不可 prefetch。测试应该涵盖这两种情况。

3.2.4 外部 DDR 存储器

外部 DDR 在测试之前需要被正确配置。如果 DDR 测试在某些板子上通过，而在另一些板子上失败，可以尝试放松一些 DDR 的时序参数，如果这样能让测试通过则往往说明 DDR 接口存在时序问题。

DSP 核和 EDMA 都应该被用来测试 DDR 存储器。而对 DSP 核的测试，既需要测试通过 cache 访问的情况，也需要测试没有 cache 的情况。

DDR 最大可以有 8GB，对所有 DDR 存储单元的测试会消耗大量测试时间。因为 EDMA 访问效率最高，所以我们可以用 EDMA 来做 DDR 全空间测试。对其它主模块，如 DSP 核，我们不需要用它来访问整个 DDR 空间，只需要测试比较小的空间就可以了，目的只是要覆盖从这个主模块到 DDR 的数据路径，而 DDR 存储单元已经被 EDMA 完整的测过了。例如，测试 DSP 核通过 L2 cache 访问 DDR，我们只需要测试比 L2 cache 大的 DDR 空间就可以了，这样就覆盖了数据路径和 L2 cache。

3.3 通过不同数据路径测试

由于不同的主模块有自己的总线，同一存储器可以被不同的主模块从不同的数据路径访问。例如，DSP 核和 EDMA 通过不同的数据路径访问 LL2。

另外，同一主模块也可以通过不同的数据路径访问同一存储器。例如，DSP 核可以通过 cache 或不通过 cache 访问 DDR 存储器。

所有这些不同的数据访问路径都应该被测试程序覆盖。

4 KeyStone DSP 上的示例测试工程

本节介绍上述测试方法在 KeyStone DSP 上实现的例子。相应的测试代码可以从以下网址获得：

http://www.deyisupport.com/cfs-file.ashx/__key/telligent-evolution-components-attachments/00-53-01-00-00-10-90-46/Memory_5F00_Test.zip

4.1 CCS 工程

例子工程的目录结构如图 2 所示。

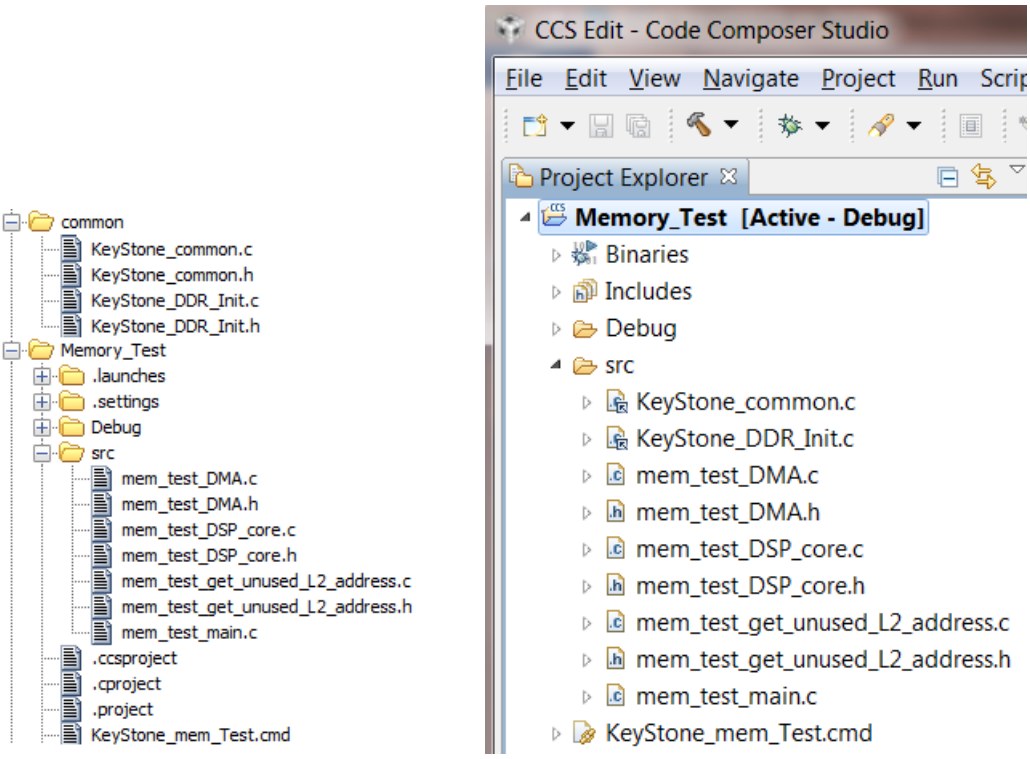

图 2 例子工程目录结构

工程文件在“Memory_Test”目录中。一些通用的初始化代码，如 PLL, EDMA, DDR 的初始化在“common”目录中。主要的测试代码在“Memory_Test\src”子目录中。表 2 描述了主要的源文件。

表 2 例子代码的源文件

源文件	描述
KeyStone_common	一些常用的公共代码，包括 PLL，EDMA 初始化等。 还包括存储器总线测试的代码，API 是： unsigned int Memory_Data_Bus_Test(unsigned int uiBaseAddress, unsigned int uiBusWidth) unsigned int Memory_Address_Bus_Test(unsigned int uiBaseAddress, unsigned int uiNumBytes, unsigned int uiMAU_bytes)
KeyStone_DDR_Init	DDR 初始化代码。
DMA_mem_test	基于 EDMA 和 IDMA 的测试实现，主要 API 包括： int EDMA_MEM_Test(unsigned int uiStartAddress, unsigned int uiStopAddress, unsigned int uiDmaBufAddress, unsigned int uiDmaBufSize); int IDMA_MEM_Test(unsigned int uiStartAddress, unsigned int uiStopAddress,

	unsigned int uiDmaBufAddress, unsigned int uiDmaBufSize);
DSP_core_mem_test	基于 DSP 核的测试实现，主要 API 包括： int DSP_core_MEM_Test(unsigned int uiStartAddress, unsigned int uiStopAddress, unsigned int uiStep);
get_unused_L2_address	解析 .map 文件以确定未用的（可测试的）的 LL2 的起始地址。

测试流程由“mem_test_main.c”中的代码控制。基本流程如下：

Disable all caches

Test LL2 bus

Test SL2 bus

Test DDR bus

Test Local L1 with IDMA

Test other core’s L1 with EDMA

Enable 32KB L1P cache

Test Local L2 with DSP core

Test other core’s L2 with DSP core

Test Shared L2 with DSP core

Test 1KB of external memory with DSP core (just cover the data path)

Enable 32KB L1D cache

Test Local L2 with EDMA

Test Local L2 with DSP core

Test other core’s L2 with EDMA

Test other core’s L2 with DSP core

Test Shared L2 with EDMA

Test Shared L2 with DSP core

Test external memory with EDMA (cover full external memory space)

Test 64KB of external memory with DSP core (just cover the data path and L1D cache)

Enable 256KB L2 cache

Test other core’s L2 with DSP core

Test 512KB external memory with DSP core (just cover the data path and L1D, L2 caches)
Test Shared L2 with DSP core through remmapped noncacheable nonprefetchable window

4.2 测试配置

测试代码中有多个宏参数可以用来对测试进行配置。

数据测试填充的数值在“DSP_core_mem_test.c”中定义如下。用户可以在这里添加，删除或修改填充的数值。

```
unsigned long long ulDataPatternTable[] = {
 0x0000000000000000,
 0xffffffffffffffff,
 0xaaaaaaaaaaaaaaaa,
 0x5555555555555555,
 0xcccccccccccccccc,
 0x3333333333333333,
 0xf0f0f0f0f0f0f0f0,
 0x0f0f0f0f0f0f0f0f,
 0xff00ff00ff00ff00,
 0x00ff00ff00ff00ff
};
```

本文介绍的三个主要测试算法可以通用以下在“DSP_core_mem_test.c”和“DMA_mem_test.c”中定义的宏开关使能或禁用。“1”表示使能，“0”表示禁用。

#define BIT_PATTERN_FILLING_TEST	1
#define ADDRESS_TEST	1
#define BIT_WALKING_TEST	1

每个存储器都可以通过“KeyStone_mem_test_main.c”中定义的宏开关控制是否被测试。

#define LL1_MEM_TEST	1
#define OTHER_L1_TEST	1
#define LL2_MEM_TEST	1
#define OTHER_L2_TEST	1
#define SL2_MEM_TEST	1

#define EXTERNAL_MEM_TEST	1

是否用 DSP 和或 DMA 测试存储器也可以通过“KeyStone_mem_test_main.c”中定义的宏开关控制。

#define TEST_BY_DSP_CORE	1
#define TEST_BY_DMA	1

这个测试工程是在 TI 的评估板上实现的。如果要在用户真实的板子上运行，需要根据板子的设计在“KeyStone_DDR_Init.c”中修改相应的 DDR 参数。PLL 倍频系数也可能需要在调用 KeyStone_main_PLL_init()的代码中修改。

要让这些修改后的配置生效，测试工程必须被重新编译。由于测试工程用到了 CSL (Chip Support Library)中的头文件，在重编之前可能还需要重新指定 CSL 的包含路径。

4.3 测试时间

测试花的时间主要取决于存储器大小，DSP 和存储器速度。表 3 列出了本文介绍的所有存储器测试在 TI 的评估板（EVM）上测试所花的时间。

表 3 在 EVM 板上的测试时间

	C6678 EVM	C6670 EVM	TCI6614 EVM
DSP 速度	1GHz x 8 核	1GHz x 4 核	1GHz x 4 核
DDR 速度	1333MTS	1333MTS	1333MTS
DDR 大小	512MB	1GB	1GB
测试时间 (秒)	155	230	293

总线测试花的时间只有几百微秒，而存储单元全空间测试花的时间很多。其中，超过 90%的时间都花在对 DDR 的测试上。

对存储单元的数据测试，地址测试，和走比特测试所花的时间的比例大概是 10:1:64，因为数据测试写读了 10 个不同的值，地址测试写读了 1 个值，而走比特测试写读了 64 个值。

用户可以根据测试时间的要求选用不同的测试用例。一般可能有三种组合：

1. 用毫秒级的时间做简单快速的总线测试。
2. 用秒级时间做全存储器空间的基本测试。测试用例包括总线测试和地址测试。
3. 用几分钟甚至几十分钟做全存储器空间的完善测试，执行本文介绍的所有测试用例。

参考文献

1. *TMS320C66x DSP CorePac User's Guide* (SPRUGW0A)
2. *KeyStone Multicore Shared Memory Controller (MSMC) User's Guide (*SRPUGW7*)*
3. *KeyStone DDR3 Memory Controller User's Guide (*SPRUGV8*)*
4. KeyStone Architecture Enhanced Direct Memory Access (EDMA3) Controller User Guide (SPRUGS5)

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

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