

TMS320C6000 指令集仿真器

技术参考

文献编号: ZHCU002

2005 年 4 月

目录	3
序言	7
1 TMS320C6000 仿真器介绍	9
1.1 功能	10
1.2 支持的处理器和仿真器配置	10
1.3 选择仿真器的注意事项	11
1.4 支持的硬件资源	11
1.4.1 CPU	11
1.4.2 存储器	11
1.4.3 外设	11
2 支持的模拟功能	15
2.1 外部事件和数据模拟	16
2.1.1 引脚连接	16
2.1.2 端口连接	19
2.2 保留的存储器存取检测	21
2.2.1 支持的配置	21
2.2.2 配置选项	21
2.2.3 错误报告格式	21
2.2.4 限制	21
2.3 CPU 资源冲突检测	21
2.3.1 支持的配置	22
2.3.2 检测到的冲突类型	22
2.3.3 不能检测到的冲突类型	22
2.3.4 配置选项	23
2.3.5 错误报告格式	23
2.4 仿真器分析	23
2.5 RTDX	23
2.6 DSP/BIOS	24
2.7 启动加载	24
2.8 应用存储器使用量检测	24
2.9 EMIF 时钟配置	24
2.10 回卷	24
3 各种配置の詳細功能	25
3.1 C62x/C64x/C67x CPU 精确到周期的仿真器	26
3.2 C6416/C6713/C6412/DM642 设备功能仿真器	26
3.2.1 支持的功能	26
3.2.2 已知限制	27
3.3 C6201/C6202/C6203/C6204/C6205/C6701 设备仿真器	27
3.3.1 支持的功能	27
3.3.2 已知限制	27
3.4 C6211/C6711/C6712/C6713 设备精确到周期的仿真器	28
3.4.1 支持的功能	28
3.4.2 已知限制	28

3.5	C6411/C6412/C6414/C6415/C6416/DM642 设备精确到周期的仿真器	28
3.5.1	支持的功能	28
3.5.2	已知限制	29
4	配置仿真器	31
4.1	设置资源冲突检测模式	32
4.2	设置保留存储器存取检测模式	32
4.3	设置启动加载	33
4.3.1	C6x0x 设备仿真器中的启动加载	33
4.3.2	C64x 设备精确到周期的仿真器/设备功能仿真器中的启动加载	33
4.4	设置 EMIF 和 CPU 时钟	33
4.5	启用回卷功能	34
4.6	设置 McBSP XBAR	34
4.6.1	如何写 XBAR 文件	34
4.6.2	要选择的配置文件格式	35
4.7	设置 McASP XBAR	35
4.7.1	要选择的配置文件格式	36
4.8	设置最大存储器使用量限制	36
4.9	引脚连接的文件格式	36
4.9.1	设置输入文件	36
4.9.2	绝对时钟周期	37
4.9.3	相对时钟周期	37
4.9.4	模式重复固定次数	37
4.9.5	重复至模拟结束 (EOS)	37
4.10	端口连接的文件格式	37
4.11	基本配置文件	38
5	性能数据	39
6	周期精确性	41
6.1	C6000 仿真器基准	42
6.2	关于周期精确度的注意事项	43

附表目录

1-1	C6000 仿真器支持的处理器	10
1-2	TMS320C6000 CPU 精确到周期的仿真器配置	11
1-3	TMS320C6201, C6202, C6203, C6204, C6205 和 C6701 仿真器	11
1-4	TMS320C6411、C6412、C6414、C6415、C6416 和 DM642 设备精确到周期的仿真器；C6412、C6416 和 DM642 功能仿真器	12
1-5	TMS320C6211、C6711、C6712 和 C6713 设备精确到周期的仿真器和 C6713 设备功能仿真器	13
2-1	TMS320C6000 设备的端口连接可用的存储器范围。	16
2-2	TMS320C6713 设备配置可用的引脚	16
2-3	TMS320C6202、C6203、C6412、C6414、C6415、C6416 和 DM642 设备配置可用的引脚	18
2-4	TMS320C64x、C62x 和 C67x CPU 精确到周期的仿真器配置可用的引脚	18
2-5	TMS320C6412、C6416、C6713 和 DM642 设备功能仿真器配置可用的引脚	18
2-6	TMS320C6000 设备的端口连接可用的存储器范围。	19
2-7	可能出现在资源冲突错误信息中的资源	23
4-1	XBAR 文件引脚名	34
5-1	C6000 仿真器的性能数据	39
6-1	C6000 仿真器的基准数据	42

请先阅读

关于本手册

此手册包含以下信息：

- 命名 TMS320C6000 指令集仿真器的配置所支持的 TMS320C6000™ (C6000™) 数字信号处理器 (DSP)
- 列出每种设备上已经建模的模块和引脚
- 说明仿真器的功能和限制
- 解释了如何配置仿真器
- 提供了一些关于周期精度的基准数据

命名惯例

本文档使用以下惯例：

- 程序示例以特殊字体显示。
- 在语法说明中，关键字或符号以**粗体**显示，变量以*斜体*显示。语法中**黑体**显示的部分应按所示方式输入；语法中*斜体*显示的部分描述了应输入的信息的类型。
- 方括号（[和]）用于标识可选参数。如果使用可选参数，需要在括号内指定信息。除非方括号是**粗体**，否则不要在方括号中再输入方括号。
- 大括号（{ 和 }）表明必须选择大括号内的参数之一，不要在大括号内输入大括号。
- 竖线符号（|）代表逻辑运算符 OR。

德州仪器（TI）提供的相关文档

TMS320C6000 CPU 及指令集参考指南 (SPRU189) 描述了 TMS320C6000 DSP 的架构、管道、指令集和中断。

TMS320C6000 外设参考指南 (SPRU190) 描述了 TMS320C6000 DSP 上可用的一般外设。本书包含与以下方面相关的信息：数据和程序存储器、外部存储器接口 (EMIF)、主机端口、串行端口、直接存储器存取 (DMA)、增强型直接存储器存取 (EDMA)、扩展总线 (XBUS)、时钟和锁相环路 (PLL) 以及断电模式。

TMS320C6000 Code Composer Studio 联机帮助 可通过 Code Composer Studio™ 集成开发环境 (IDE) 查阅。

商标

Intel, Pentium 都是 Intel Corporation or its subsidiaries in the United States and other countries 的注册商标

TMS320C6000, C6000, Code Composer Studio, TMS320C62x, TMS320C64x, TMS320C67x, C62x, C64x, C67x, RTDX, DSP/BIOS 是 Texas Instruments 的商标

TMS320C6000 仿真器介绍

TMS320C6000™ 指令集仿真器，可与 TMS320C6000 的 Code Composer Studio™ 集成开发环境 (IDE) 一起使用，它能够模拟 TMS320C62x™、TMS320C64x™ 和 TMS320C67x™ 这几代设备。本章列出了仿真器支持的 TMS320C6000 设备，并提供了帮助用户选择最佳仿真器配置的相关信息。

主题	页
1.1 功能	10
1.2 支持的处理器和仿真器配置	10
1.3 选择仿真器的注意事项	11
1.4 支持的硬件资源	11

1.1 功能

C6000™ 仿真器支持以下功能：

- TMS320C6000 CPU 全指令集架构执行
- 支持 C62x™, C64x™ 和 C67x™。支持所有基于这些内核的设备。
- 端口连接，同时支持外围设备模拟
- 引脚连接，同时支持外部中断模拟
- 分析事件支持调试和分析
- RTDX™ 支持
- DSP/BIOS™ 实时分析支持
- CPU 资源冲突检测
- 回卷功能支持 C6x CPU 仿真器

1.2 支持的处理器和仿真器配置

表 1-1 列出了 C6000 指令集仿真器支持的设备，以及在 Code Composer Studio Setup 程序的 Factory Board（测试板）菜单中需要选择的相应配置。

表 1-1. C6000 仿真器支持的处理器

处理器	Code Composer Studio IDE 导入配置 ⁽¹⁾	建模的 CPU
TMS320C62x	C62xx CPU 精确到周期的仿真器	TMS320C62x
TMS320C64x	C64xx CPU 精确到周期的仿真器	TMS320C64x
TMS320C67x	C67xx CPU 精确到周期的仿真器	TMS320C67x
TMS320C6211	C6211 设备精确到周期的仿真器	TMS320C62x
TMS320C6411	C6411 设备精确到周期的仿真器	TMS320C64x
TMS320C6412 ⁽²⁾	C6412 设备精确到周期的仿真器	TMS320C64x
TMS320C6414	C6414 设备精确到周期的仿真器	TMS320C64x
TMS320C6415	C6415 设备精确到周期的仿真器	TMS320C64x
TMS320C6416 ⁽²⁾	C6416 设备精确到周期的仿真器	TMS320C64x
TMS320C6711	C6711 设备精确到周期的仿真器	TMS320C67x
TMS320C6712	C6712 设备精确到周期的仿真器	TMS320C67x
TMS320C6713 ⁽²⁾	C6713 设备精确到周期的仿真器	TMS320C67x
TMS320DM642 ⁽²⁾	DM642 设备精确到周期的仿真器	TMS320C64x
TMS320C6201	C6201 设备仿真器 ⁽³⁾	TMS320C62x
TMS320C6202	C6202 设备仿真器 ⁽³⁾	TMS320C62x
TMS320C6203	C6203 设备仿真器 ⁽³⁾	TMS320C62x
TMS320C6204	C6204 设备仿真器 ⁽³⁾	TMS320C62x
TMS320C6205	C6205 设备仿真器 ⁽³⁾	TMS320C62x
TMS320C6701	C6701 设备仿真器 ⁽³⁾	TMS320C67x

注：

Multiple C6000 仿真器配置不能在 Code Composer Studio Setup 程序 中导入。

(1) 该产品提供大端和小端配置（默认）。

(2) 设备功能仿真器对于设备精确到周期的仿真器同样可用，只需在 Code Composer Studio Setup 程序的处理器属性窗口中将 Simulator Type（仿真器类型）项从 Cycle Accurate（精确到周期）更改为 Functional（功能）便可使用该仿真器。

(3) 提供 Map 0 和 Map 1 配置（默认）。

1.3 选择仿真器的注意事项

不同的仿真器在功能建模、模拟的周期精确度和模拟性能方面所具有的优势也不相同。您可以根据应用的需要，待执行的功能和要求的周期精确度级别来选择不同的配置。您可以在 Code Composer Studio Setup 程序的 Factory Boards (测试板) 菜单中选择 CPU Cycle Accurate Simulators (CPU 精确到周期的仿真器) 或 Device Cycle Accurate Simulators (设备精确到周期的仿真器)。对于 C6x1x 仿真器，您可以通过在 Code Composer Studio Setup 程序的处理器属性窗口中将 Simulator Type (仿真器类型) 项从 Cycle Accurate (精确到周期) 更改为 Functional (功能) 便可使用该仿真器。

如果您的主要目的是优化核心算法，则可以选择 CPU 精确到周期仿真器配置。这样便只涉及内核精确度，而不需要完全的设备模拟。所有内核外的存取都是由平面存储器处理的；测量到的周期不能说明任何存储器存取延迟。

设备功能仿真器配置可以为一些关键外设的功能建模，而对仿真器性能仅会产生微小的影响。在这些仿真器中，被建模的只是外设的功能，以便于编程人员查看；它们并未精确到周期。当周期精确度不是很重要时，用户如果需要使用这些在功能仿真器中支持的，但在核心仿真器中不存在的设备功能，就可以使用这些配置。

例如，C6713 设备功能仿真器可以用于不需要周期精确度的应用开发。C6713 设备功能仿真器的运行速度快于 C6713 设备精确到周期的仿真器。

设备精确到周期的仿真器配置可以为设备的大部分外设建模。外设是精确到周期的，则仿真器也会精确到周期。这些仿真器可用于了解应用中周期的行为状况。有关特定仿真器配置的精确度的详细信息，请参阅第 3 章。

1.4 支持的硬件资源

以下部分对每种仿真器配置所支持的硬件资源进行了简要概述。有关仿真器配置的详细信息，请参阅第 3 章。

1.4.1 CPU

CPU 模型是精确到周期的。周期效应均可建模。

1.4.2 存储器

对于每种设备仿真器和设备精确到周期的仿真器，高速缓存和内存模型均与 TMS320C6000 外设参考指南 (SPRU190) 中描述的高速缓存和内存架构规格相匹配。它们支持标准高速缓存行为，例如：LRU 行替换、直接映射、指令集相关性、命中/缺失服务的高速缓存协议，以及监听和攻击等。为了匹配设备规格，同时建模的还有分配政策（读缺失时分配、写缺失时写回等）。平面存储器模型被钩至 EMIF 上以允许外部存储器存取。

对于 CPU 精确到周期的仿真器和设备功能仿真器，整个存储器范围将被建成平面存储器模型。

注：

不支持使用 GUI 或 GEL 脚本设置存储器映射。

1.4.3 外设

下表列出了每种仿真器配置所支持的外设。有关这些外设的详细信息，请参阅 TMS320C6000 外设参考指南 (SPRU190)。

表 1-2. TMS320C6000 CPU 精确到周期的仿真器配置

	C62x	C64x	C67x
CPU	支持	支持	支持
平面存储器	支持	支持	支持
定时器	支持	支持	支持

表 1-3. TMS320C6201, C6202, C6203, C6204, C6205 和 C6701 仿真器

	C6201	C6202	C6203	C6204	C6205	C6701
CPU	支持	支持	支持	支持	支持	支持

表 1-3. TMS320C6201, C6202, C6203, C6204, C6205 和 C6701 仿真器(接上表)

	C6201	C6202	C6203	C6204	C6205	C6701
内存/高速缓存模型	支持	支持	支持	支持	支持	支持
DMA	支持	支持	支持 ⁽¹⁾	支持 ⁽¹⁾	支持 ⁽¹⁾	支持
EMIF	支持	支持	支持	支持	支持	支持
中断选择器	支持	支持	支持	支持	支持	支持
McBSP	支持	支持	支持	支持	支持	支持
定时器	支持	支持	支持	支持	支持	支持
GPIO	不支持	不支持	不支持	不支持	不支持	不支持
HPI/XBUS/PCI	不支持	不支持	不支持	不支持	不支持	不支持

⁽¹⁾ 在此设备上所建的 DMA 功能模型与 C6202 仿真器中的功能是一样的。尚未对增强功能建模。

表 1-4. TMS320C6411、C6412、C6414、C6415、C6416 和 DM642 设备精确到周期的仿真器；C6412、C6416 和 DM642 功能仿真器

	C6411	C6412	C6414	C6415	C6416	DM642	C6412 FUNC	C6416 FUNC	DM642 FUNC
CPU	支持	支持	支持	支持	支持	支持	支持	支持	支持
内存/ 高速缓存模型	支持	支持	支持	支持	支持	支持	支持	支持	支持
QDMA	支持	支持	支持	支持	支持	支持	支持	支持	支持
EDMA	支持	支持	支持	支持	支持	支持	支持	支持	支持
EMIF	支持	支持	支持	支持	支持	支持	支持	支持	支持
中断选择器	支持	支持	支持	支持	支持	支持	支持	支持	支持
McBSP	支持	支持	支持	支持	支持	支持	支持	支持	支持
定时器	支持	支持	支持	支持	支持	支持	支持	支持	支持
GPIO	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持
HPI	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持
TCP/VCP	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	支持	无 ⁽¹⁾	无 ⁽¹⁾	不支持	无 ⁽¹⁾
Utopia	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	不支持	不支持	无 ⁽¹⁾	无 ⁽¹⁾	不支持	无 ⁽¹⁾
PCI	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持	不支持
?? McASP	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	支持	无 ⁽¹⁾	无 ⁽¹⁾	不支持
视频端口	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	无 ⁽¹⁾	不支持	无 ⁽¹⁾	无 ⁽¹⁾	不支持

⁽¹⁾ 无 - 硬件中不存在

表 1-5. TMS320C6211、C6711、C6712 和 C6713 设备精确到周期的仿真器和 C6713 设备功能仿真器

	C6211	C6711	C6712	C6713	C6713 FUNC
CPU	支持	支持	支持	支持	支持
内存/高速缓存模型	支持	支持	支持	支持	支持
QDMA	支持	支持	支持	支持	支持
EDMA	支持	支持	支持	支持	支持
EMIF	支持	支持	支持	支持	支持
中断选择器	支持	支持	支持	支持	支持
McBSP	支持	支持	支持	支持	支持
?? McASP	不支持	不支持	不支持	支持	不支持
定时器	支持	支持	支持	支持	支持
GPIO	不支持	不支持	不支持	不支持	不支持
HPI	不支持	不支持	不支持	不支持	不支持

支持的模拟功能

本章简要概述了每种仿真器配置所支持的模拟功能。有关每种配置的详细信息，请参阅第 3 章。

主题	页
2.1 外部事件和数据模拟	16
2.2 保留的存储器存取检测	21
2.3 CPU 资源冲突检测	21
2.4 仿真器分析	23
2.5 RTDX	23
2.6 DSP/BIOS	24
2.7 启动加载	24
2.8 应用存储器使用量检测	24
2.9 EMIF 时钟配置	24
2.10 回卷	24

2.1 外部事件和数据模拟

在真实的硬件中，DSP 可以与许多外部实体进行交互。仿真器具备模拟这些交互的功能。仿真器与这些外部实体之间的交互可分为以下两类：

- 控制信号触发活动到仿真器（例如，中断、串行端口锁定和串行端口同步事件）。
- 数据值是仿真器和外部实体之间交互的一部分（例如，作为 I/O 存储器一部分的对外设寄存器的读写，以及串行端口数据）。

例如，音频设备中的 DSP 的串行端口连接到 A/D 和 D/A 转换器或编解码器。DSP 与音频设备之间的交互方法是通过传输用于启动采样的同步信号及采样数据本身来进行的。其中，同步信号属于控制信号类别，而采样数据属于数据值类别。

仿真器提供了引脚连接和端口连接功能，分别用于模拟这两种类型的交互。

2.1.1 引脚连接

引脚连接工具允许您模拟并监控来自外部中断的信号。为了包括外部中断/触发器，TMS320C6000 设备中模仿了一些引脚。任何具有指定格式的文件都可以连接到这些引脚。

被模拟的引脚包括两种类型：脉冲和波形。脉冲引脚受上升沿的影响，而波形引脚受上升沿和下降沿的影响。例如，CPU 中断引脚受上升沿的影响，TMS320C6000 设备的串行端口 (CLKX, CLKR) 的时钟输入引脚受上升沿和下降沿的影响。请参阅以下表格，以了解不同的仿真器配置所支持的各种引脚的详细信息。

表 2-1. TMS320C6000 设备的端口连接可用的存储器范围。

引脚	说明	类型
NMI	不可屏蔽中断	脉冲
INT4	通用外部中断引脚	脉冲
INT5	通用外部中断引脚	脉冲
INT6	通用外部中断引脚	脉冲
INT7	通用外部中断引脚	脉冲
TINP0	Timer0 输入引脚	波形
TINP1	Timer1 输入引脚	波形
FSX0	McBSP0 的发送帧同步引脚	波形
FSR0	McBSP0 的接收帧同步引脚	波形
CLKX0	McBSP0 的发送时钟引脚	波形
CLKR0	McBSP0 的接收时钟引脚	波形
CLKS0	McBSP0 的外部时钟输入	波形
FSX1	McBSP1 的发送帧同步引脚	波形
FSR1	McBSP1 的接收帧同步引脚	波形
CLKX1	McBSP1 的发送时钟引脚	波形
CLKR1	McBSP1 的接收时钟引脚	波形
CLKS1	McBSP1 的外部时钟输入	波形

表 2-2. TMS320C6713 设备配置可用的引脚

引脚	说明	类型
NMI	不可屏蔽中断	脉冲
INT4	通用外部中断引脚	脉冲
INT5	通用外部中断引脚	脉冲
INT6	通用外部中断引脚	脉冲
INT7	通用外部中断引脚	脉冲
TINP0	Timer0 输入引脚	波形
TINP1	Timer1 输入引脚	波形
FSX0	McBSP0 的发送帧同步引脚	波形
FSR0	McBSP0 的接收帧同步引脚	波形

表 2-2. TMS320C6713 设备配置可用的引脚(接上表)

引脚	说明	类型
CLKX0	McBSP0 的发送时钟引脚	波形
CLKR0	McBSP0 的接收时钟引脚	波形
CLKS0	McBSP0 的外部时钟输入	波形
FSX1	McBSP1 的发送帧同步引脚	波形
FSR1	McBSP1 的接收帧同步引脚	波形
CLKX1	McBSP1 的发送时钟引脚	波形
CLKR1	McBSP1 的接收时钟引脚	波形
CLKS1	McBSP1 的外部时钟输入	波形
ACLKX0	McASP0 发送元件的时钟引脚	脉冲
ACLKR0	McASP0 接收元件的时钟引脚	脉冲
AHCLKX0	McASP0 发送元件的较高频率时钟引脚	脉冲
AHCLKR0	McASP0 接收元件的较高频率时钟引脚	脉冲
AFSX0	McASP0 发送元件的帧同步引脚	脉冲
AFSR0	McASP0 接收元件的帧同步引脚	脉冲
ACLKX1	McASP1 发送元件的时钟引脚	脉冲
ACLKR1	McASP1 接收元件的时钟引脚	脉冲
AHCLKX1	McASP1 发送元件的较高频率时钟引脚	脉冲
AHCLKR1	McASP1 接收元件的较高频率时钟引脚	脉冲
AFSX1	McASP1 发送元件的帧同步引脚	脉冲
AFSR1	McASP1 接收元件的帧同步引脚	脉冲

表 2-3. TMS320C6202、C6203、C6412、C6414、C6415、C6416 和 DM642 设备配置可用的引脚

引脚	说明	类型
NMI	不可屏蔽中断	脉冲
INT4	通用外部中断引脚	脉冲
INT5	通用外部中断引脚	脉冲
INT6	通用外部中断引脚	脉冲
INT7	通用外部中断引脚	脉冲
TINP0 ⁽¹⁾	Timer0 输入引脚	波形
TINP1 ⁽¹⁾	Timer1 输入引脚	波形
FSX0	McBSP0 的发送帧同步引脚	波形
FSR0	McBSP0 的接收帧同步引脚	波形
CLKX0	McBSP0 的发送时钟引脚	波形
CLKR0	McBSP0 的接收时钟引脚	波形
CLKS0	McBSP0 的外部时钟输入	波形
FSX1	McBSP1 的发送帧同步引脚	波形
FSR1	McBSP1 的接收帧同步引脚	波形
CLKX1	McBSP1 的发送时钟引脚	波形
CLKR1	McBSP1 的接收时钟引脚	波形
CLKS1	McBSP1 的外部时钟输入	波形
FSX2 ⁽²⁾	McBSP2 的发送帧同步引脚	波形
FSR2 ⁽²⁾	McBSP2 的接收帧同步引脚	波形
CLKX2 ⁽²⁾	McBSP2 的发送时钟引脚	波形
CLKR2 ⁽²⁾	McBSP2 的接收时钟引脚	波形
CLKS2 ⁽²⁾	McBSP2 的外部时钟输入	波形
ACLKRO ⁽³⁾	McASPO 接收元件的时钟引脚	脉冲
ACLKXO ⁽³⁾	McASPO 发送元件的时钟引脚	脉冲
AHCLKRO ⁽³⁾	McASPO 接收元件的较高频率时钟引脚	脉冲
AHCLKXO ⁽³⁾	McASPO 发送元件的较高频率时钟引脚	脉冲
AFSRO ⁽³⁾	McASPO 接收元件的帧同步引脚	脉冲
AFSXO ⁽³⁾	McASPO 发送元件的帧同步引脚	脉冲

(1) 此引脚仅适用于 C6202 和 C6203 设备仿真器。

(2) 此引脚不适用于 DM642 设备精确到周期的仿真器，因为它仅具有两个 McBSP。

(3) 此引脚仅适用于 DM642 设备精确到周期的仿真器。

表 2-4. TMS320C64x、C62x 和 C67x CPU 精确到周期的仿真器配置可用的引脚

引脚	说明	类型
NMI	不可屏蔽中断	脉冲
INT4	通用外部中断引脚	脉冲
INT5	通用外部中断引脚	脉冲
INT6	通用外部中断引脚	脉冲
INT7	通用外部中断引脚	脉冲

表 2-5. TMS320C6412、C6416、C6713 和 DM642 设备功能仿真器配置可用的引脚

引脚	说明	类型
NMI	不可屏蔽中断	脉冲
FSX0	McBSP0 的发送帧同步引脚	脉冲
FSR0	McBSP0 的接收帧同步引脚	脉冲
CLKX0	McBSP0 的发送时钟引脚	脉冲

表 2-5. TMS320C6412、C6416、C6713 和 DM642 设备功能仿真器配置可用的引脚(接上表)

引脚	说明	类型
CLKR0	McBSP0 的接收时钟引脚	脉冲
CLKS0	McBSP0 的外部时钟输入	脉冲
FSX1	McBSP1 的发送帧同步引脚	脉冲
FSR1	McBSP1 的接收帧同步引脚	脉冲
CLKX1	McBSP1 的发送时钟引脚	脉冲
CLKR1	McBSP1 的接收时钟引脚	脉冲
CLKS1	McBSP1 的外部时钟输入	脉冲
FSX2 ⁽¹⁾	McBSP2 的发送帧同步引脚	脉冲
FSR2 ⁽¹⁾	McBSP2 的接收帧同步引脚	脉冲
CLKX2 ⁽¹⁾	McBSP2 的发送时钟引脚	脉冲
CLKR2 ⁽¹⁾	McBSP2 的接收时钟引脚	脉冲
CLKS2 ⁽¹⁾	McBSP2 的外部时钟输入	脉冲

(1) 此引脚仅在 C6416 设备功能仿真器中可用。

引脚连接功能可以通过 Code Composer Studio 命令窗口、GEL 命令或引脚连接插件来启用。请参阅第 4.9 部分以获得有关引脚文件格式的详细信息。请参阅 Code Composer Studio 联机帮助以获得有关如何连接和断开特定引脚的详细信息。

2.1.2 端口连接

端口连接功能可以将外部数据送入仿真器，并将仿真器数据发送到外部实体。外部数据是通过端口连接文件（参见第 4.10 部分中介绍的文件格式）送入和发出仿真器的）的图像所示。此功能可用于在支持的地址设置到仿真器的输入或输出数据流。文件连接到存储器（端口）地址以进行读取时，只要读取设备中的存储器，则文件中的数据将被存取。同样，当文件连接到存储器（端口）地址进行写入时，所有写入该地址的数据将被写入文件。仿真器为所有处理器配置提供端口连接功能。

如果是串行端口，则可以在写模式下由串行端口发送寄存器的存储器映射地址连接一些文件，以达到发送数据的目的。同样，您可以在读取模式下由串行端口接收寄存器的存储器映射地址连接一些文件，以达到接收数据的目的。TMS320C6000 设备上的可用存储器范围，文件可以连接到这些地址进行读/写，具体如表 2-6。

表 2-6. TMS320C6000 设备的端口连接可用的存储器范围。

配置文件 ⁽¹⁾	可用存储器范围
C6201 设备仿真器, Map 1	0x0040 0000 - 0x017F FFFF DXR0, DRR0, DXR1, DRR1 ⁽²⁾
C6201 设备仿真器, Map 0	0x0000 0000 - 0x013F FFFF DXR0, DRR0, DXR1, DRR1 ⁽²⁾
C6202 设备仿真器, Map 1	0x0040 0000 - 0x017F FFFF DXR0, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6202 设备仿真器, Map 0	0x0000 0000 - 0x013F FFFF DXR0, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6203 设备仿真器, Map 1	0x0040 0000 - 0x017F FFFF DXR0, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6203 设备仿真器, Map 0	0x0000 0000 - 0x013F FFFF DXR0, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6204 设备仿真器, Map 1	0x0040 0000 - 0x017F FFFF DXR0, DRR0, DXR1, DRR1 ⁽²⁾
C6204 设备仿真器, Map 0	0x0000 0000 - 0x013F FFFF DXR0, DRR0, DXR1, DRR1 ⁽²⁾

(1) 对于列出的每种配置，它们提供的小端（默认）和大端版本都具有相同的可用存储器范围，以用于串行端口连接。

(2) 要连接这些外设寄存器，使用存储器映射地址。只能通过以下地址实现端口到 McBSP 的连接：0x3000 0000（对应 DX0 和 DR0）、0x3400 0000（对应 DX1 和 DR1）以及 0x3800 0000（对应 DX2 和 DR2）。

表 2-6. TMS320C6000 设备的端口连接可用的存储器范围。(接上表)

配置文件 ⁽¹⁾	可用存储器范围
C6205 设备仿真器, Map 1	0x0040 0000 - 0x017F FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6205 设备仿真器, Map 0	0x0000 0000 - 0x013F FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6411 设备精确到周期的仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6412 设备精确到周期的仿真器	0x0800 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6412 设备功能仿真器	0x8000 0000 - 0xBFFF FFFF
C6414 设备精确到周期的仿真器	0x6000 0000 - 0x6FFF FFFF 0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6415 设备精确到周期的仿真器	0x6000 0000 - 0x6FFF FFFF 0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6416 设备精确到周期的仿真器	0x6000 0000 - 0x6FFF FFFF 0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6416 设备功能仿真器	0x6000 0000 - 0x6FFF FFFF 0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1, DXR2, DRR2 ⁽²⁾
C6701 设备仿真器, Map 1	0x0040 0000 - 0x017F FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6701 设备仿真器, Map 0	0x0000 0000 - 0x013F FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6211 设备精确到周期的仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6711 设备精确到周期的仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C6713 设备精确到周期的仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1, McASPO TXBUFO-15, McASPO RXBUFO-15, McASP1 TXBUFO-15, McASP1 RXBUFO-15 ⁽²⁾
C6713 设备功能仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
DM642 设备精确到周期的仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1, McASPO TXBUFO-15, McASPO RXBUFO-15 ⁽²⁾
DM642 设备功能仿真器	0x8000 0000 - 0xBFFF FFFF DXRO, DRR0, DXR1, DRR1 ⁽²⁾
C62xx CPU 精确到周期的仿真器	0x0000 0000 - 0xFFFF FFFF
C64xx CPU 精确到周期的仿真器	0x0000 0000 - 0xFFFF FFFF
C67xx CPU 精确到周期的仿真器	0x0000 0000 - 0xFFFF FFFF

2.1.2.1 读模式

这两种模式用于连接端口连接文件:

- **回卷模式**是默认连接模式。在此模式中,在彻底消耗完读取文件的内容后,仿真器会将文件回卷,从文件头开始读取,以执行进一步的读存取。例如,在读取模式中将下面的示例文件用于地址 0x2000,在对地址进行第六次读存取时,0x12346666 被读取。
- 在**非回卷模式**中,在文件尾进行的读存取不会导致读文件。读取的数据是存储器存在的任何内容。例如,在写模式中将下面的示例文件用于地址 0x2000,在对地址进行第六次读存取时,读取的值即为存储器中的值 - 0x89897f7f (第五次存取读取的值)。

C6000 目标的示例文件(字长,4 字节):

```
12346666
33449999
```

cb56aaaa
5656cccc
89897f7f

2.1.2.2 复位

在复位时，所有读文件指针都会被回卷，写文件被关闭并在写模式中重新打开。

端口连接可通过 Code Composer Studio 命令窗口、GEL 命令，或通过端口连接插件进行。请参阅 Code Composer Studio 联机帮助以获得如何连接和断开特定端口的详细信息。

2.2 保留的存储器存取检测

DSP 应用对设备的保留存储器位置进行存取将造成不确定的程序行为。当应用存取该设备的保留存储器位置时，仿真器可以通过标记错误消息来检测此类违例。

如果已在仿真器中为外设建模，则任何对此外设的存取也将被视为对保留存储器的存取。仿真器对此类地址的任何写存取不会有效，且读取返回零；例如，C6146 设备功能仿真器上的 TCP/VCP 存储器映射地址。

2.2.1 支持的配置

所有 C6x1x 仿真器都具备此功能。

所有 C6x0x 仿真器都不支持此功能。

CPU 仿真器没有对任何保留存储器建模。

2.2.2 配置选项

在以下模式中，通过 Code Composer Studio Setup 程序配置仿真器以进行保留存储器存取检测：

- **YES (检测)**。所有保留存储器存取将被检测出来，并标记为错误，同时还将标出存取的地址。模拟暂停
- **NO (不检测)**。未使用保留存储器存取检测。写入到保留存储器无效，同时，读取返回 0。
- **Create Log (创建日志)**。所有保留存储器存取将被检测出来，并记录在文本文件中，同时还将记录存取的地址。模拟不会暂停。

请参阅第 4.2 部分 以获得详细信息。

2.2.3 错误报告格式

如果打开了保留存储器检测，则系统会按照下面的格式报告/记录错误：

存取保留存储器的范围：

存储器映射错误：{READ | WRITE} 存取了被硬件保留的地址 AAAAAAAAAA。

例如：

存储器映射错误：WRITE 存取了被硬件保留的地址 0xc7fffc。

存取不支持的存储器范围：

存储器映射错误：{READ | WRITE} 存取了仿真器不支持的地址 AAAAAAAAAA。

例如：

存储器映射错误：READ 存取了仿真器不支持的地址 0x188000。

2.2.4 限制

除了硬件中保留的存储区范围外，此功能不支持（通过使用 Code Composer Studio 的内存映射功能）增加任何新的保留的存储区范围。同时也不允许选择性地删除任何此类保留的存储区。

2.3 CPU 资源冲突检测

C6000 CPU 在其 A 和 B 两面各有四个功能单元。这些单元可以加强并行功能，支持最多八个并行指令。这些单元，以及数据存取路径和 CPU 寄存器便是构成 C6000 CPU 资源的主要部分。

在不同的指令下，同时使用这些资源有不同的限制。如果执行代码与这些限制发生冲突，则不能保证应用行为。

仿真器中的资源冲突检测非常重要，因为许多这类冲突仅在运行时才显现出来，所以无法在汇编时检测到它们。

2.3.1 支持的配置

所有的 C6000 仿真器都支持资源冲突检测。

2.3.2 检测到的冲突类型

仿真器可以检测到以下类型的冲突：

- 单元过度使用（S、M、L 和 D 单元）
- XPath 和 T 单元过度使用
- C67x 多周期单元使用限制
- 写多个寄存器
- 多个多周期 NOP 处于同一周期
- 多个分支处于同一周期

2.3.3 不能检测到的冲突类型

同时多次读取一个寄存器（多于四次）。

2.3.4 配置选项

在以下三种模式中，通过 Code Composer Studio Setup 程序配置仿真器以进行资源冲突检测：

- YES（检测）。所有冲突均被检测，并标记为错误。模拟暂停
- NO（不检测）。不使用资源冲突检测。
- Create Log（创建日志）。所有冲突均被检测，并且错误信息被记录到文件中。模拟不会暂停。

请参阅第 4.1 部分以获得详细信息。

2.3.5 错误报告格式

检测到资源冲突时，仿真器会生成一个错误字符串，显示在错误窗口中或记录到文件中，这取决于配置的模式。错误字符串会以下面的格式报告：

```

错误类型 *** 错误信息出现在 PC= XXXXXXXX
资源 YYY、YYY 和 YYY 的某类型冲突发生在 ZZZZ 阶段
参考 文献编号 Sec ABC
  
```

例如：

```

运行目标 CPU 时出错 ***运行时错误出现在 PC = 00000030
寄存器 B7 的多写冲突发生在 E1 阶段。
Ref SPRU189 Sec 3.7, Sec 5.6
  
```

- 指明的阶段 ZZZZ，与 PC = XXXXXXXX 上的指令相关。
- 显示的元件信息仅提供了概览参考，并取决于设备类型。

表 2-7 列出了有效的资源名称。

表 2-7. 可能出现在资源冲突错误信息中的资源

L 单元/读端口	T 存取路径	32 MSBPATH 加载路径	多周期 Nop
M 单元/读端口	32 LSBPATH 存储路径	L 单元长写端口	控制寄存器文件
D 单元/读端口	32 MSBPATH 存储路径	M 单元长写端口	共享 32 LSB 存储路径/L/S 单元长写端口
S 单元/读端口	L 单元写端口	S 单元长写端口	共享 32 LSB 存储路径/L/S 单元长写端口
L 单元长读端口	M 单元写端口	内部 M 单元资源 V	共享 L/S 单元长写端口寄存器 A# 和 B#
M 单元长读端口	S 单元写端口	内部 M 单元资源 U	无效资源
S 单元长读端口	32 LSBPATH 加载路径	XPath	

2.4 仿真器分析

TMS320C6000 仿真器分析允许您设置并监控特定事件的出现。部分模拟的事件包括程序缓存缺失、程序缓存命中、程序获取、程序存取块 0 和程序存取块 1。仿真器分析插件可以报告特殊系统事件的出现情况，以便您精确地监控和测量程序性能。您可以设置事件以增大计数器值或事件出现时停止执行。

暂停执行事件的功能可用于调试应用的执行。例如，通过设置 C6211 目标在 EDMA 的 Timer0 同步事件被触发时暂停，用户可以调试同步事件是否在正常运行。

计算一段执行期内的事件数的功能可以使用户了解执行期间内程序的行为情况。例如，计算高速缓存缺失数有助于在优化存储器布局时确定其热点。请参阅 Code Composer Studio IDE 联机帮助，以获得为每种配置提供的分析事件列表，并了解如何通过使用仿真器分析插件来启用它们。

2.5 RTDX

仿真器支持运行实时数据交换 (RTDX)。要使用仿真器运行 RTDX 应用，您必须将应用与 RTDX 仿真器目标库连接起来。您可以很容易地将仿真器上运行的应用切换到在真正的硬件上运行。有关 RTDX 的详细信息，请参阅 Code Composer Studio IDE 联机帮助。

2.6 DSP/BIOS

任何使用 DSP/BIOS 的应用都可以在所有 C6000 仿真器上运行。为了对这些应用启用实时分析，您需要确保配置中的 RTDX 模式被设置为仿真器。请参阅 Code Composer Studio IDE 联机帮助，以获得有关 DSP/BIOS 主题的详细信息。

2.7 启动加载

启动加载是一个过程，它将有限数量的字（具体数量因 C620x/C670x 和 C621x/C64x 设备而有所不同）从启动模式指定的地址复制到地址 0x0。只有通过仿真器基本配置文件指定有效的启动模式将启动加载启用后，启动加载才会在仿真器中执行。请参阅第 4.3 部分以获得详细信息。

2.8 应用存储器使用量检测

该功能可以检测应用是否占用了大量的存储器。如果应用的存储器使用量超过了指定的上限（默认值为 64 MB），仿真器就会报告错误。您可以通过基本配置文件中提供的选项改写此默认限制。请参阅第 4 章以获得详细信息。

注：

C6x0x 仿真器不支持应用的存储器使用量检测。

2.9 EMIF 时钟配置

通过使用 Code Composer Studio Setup 程序，以下仿真器支持 EMIF 的可编程性和 CPU 时钟速度：

- C6211 设备精确到周期的仿真器
- C6711/C6712/C6713 设备精确到周期的仿真器
- C6411/C6412/C6414/C6415/C6416/DM642 设备精确到周期的仿真器

有关详细信息，请参阅第 4.4 部分。

2.10 回卷

以下 CPU 仿真器在 Code Composer Studio 环境下支持称为“回卷”的功能。使用回卷功能，您便可以查看过去所执行的应用的记录。这样可以减少调试应用所花的时间。有关详细信息，请参阅‘回卷功能’用户指南 (SPRU713)。

- C62xx CPU 精确到周期的仿真器
- C64xx CPU 精确到周期的仿真器
- C67xx CPU 精确到周期的仿真器

请参阅第 4.5 部分以获得在 CPU 仿真器上启用此功能的详细信息。

各种配置の詳細功能

本章介绍了每种配置的功能和已知限制。

主题	页
3.1 C62x/C64x/C67x CPU 精确到周期的仿真器	26
3.2 C6416/C6713/C6412/DM642 设备功能仿真器	26
3.3 C6201/C6202/C6203/C6204/C6205/C6701 设备仿真器	27
3.4 C6211/C6711/C6712/C6713 设备精确到周期的仿真器	28
3.5 C6411/C6412/C6414/C6415/C6416/DM642 设备精确到周期的仿真器	28

3.1 C62x/C64x/C67x CPU 精确到周期的仿真器

在 CPU 精确到周期的仿真器配置中，建模的只有 CPU 核心，同时还提供了设备计时器支持和完全可寻址空间的平面存储器模型。如果不需要设备外设的功能，这些仿真器可用于算法验证。这些配置比起其它的仿真器配置要更快。这些仿真器同样支持回卷（参见第 4.5 部分），此功能有助于更快地调试应用。

支持所有在 *TMS320C6000 CPU 和 指令集参考指南* 中描述的指令 (SPRU189)。这包括支持所有的寄存器、寻址模式、分支条件、并行指令和浮点指令。CPU 核心模型是精确到周期的，已对管线效果进行了完全建模。每种仿真器配置支持所有用户可见的寄存器（数据寄存器集 A 和 B，以及所有控制寄存器）。支持所有中断。

CPU 精确到周期的仿真器具有精确到周期的计时器模型。它们同样支持模拟使用 RTDX 的应用。这些仿真器上可以运行基于 DSP/BIOS 的应用，并可以使用实时分析功能来调试和分析应用。另外，引脚连接和端口连接功能可用于设置应用所需的外部激励，以进行模拟。

这些配置所支持的引脚列表和可连接到文件的存储器范围如第 2 章。通过使用仿真器分析插件，这些仿真器配置便可跟踪多个目标事件，并允许计算事件数量或中断这些事件。请参阅 Code Composer Studio IDE 联机文档，以获得有关事件列表以及如何使用这些事件的详细信息。

3.2 C6416/C6713/C6412/DM642 设备功能仿真器

设备功能仿真器将对外设的功能建模，但不会对这些设备在整个周期中的行为建模。所以，这些仿真器比起其对应的精确到周期的模型在速度上会更快，但仍然可以运行使用这些外设的应用。这些仿真器可用于验证应用的功能性以及测量应用可能需要的时钟周期。

3.2.1 支持的功能

DM642、C6416 和 C6713 设备功能仿真器具有精确到周期的 CPU；精确到周期的计时器模块；L1P、L1D 和 L2 高速缓存的功能模型；中断选择器的功能模型；McBSP 的功能模型；EMIF 的功能模型；以及 EDMA 的功能模型。所有超出高速缓存的存储器将被建为平面存储器模型。

C6416 设备功能仿真器可用于在功能上模拟 C6414/15 设备。C6713 设备功能仿真器可用于在功能上模拟 C6711、C6712 和 C6211 的配置。C6412 设备功能仿真器可用于在功能上模拟 C6411 的配置。

使用 RTDX 和 DSP/BIOS 实时分析的应用可在这些仿真器上运行。另外，引脚连接和端口连接功能可用于设置应用所需的外部激励，以进行模拟。

这些配置所支持的引脚列表和可连接到文件的存储器范围如第 2 章。这些仿真器配置可跟踪多个目标事件，并允许计算事件数量或中断这些事件。请参阅 Code Composer Studio IDE 联机文档，以获得有关事件列表以及如何使用这些事件的详细信息。

3.2.2 已知限制

这些仿真器仅对计时器、L1 和 L2 高速缓存、中断选择器、EMIF、McBSP 和 EDMA 外设建模。

- **McBSP 限制**
 - 不支持 SPCR 寄存器中的 FREE 和 SOFT 字段，因为这是模拟的功能。
 - 不支持启用额外延迟 DX 打开时间。功能设备仿真器不支持 TRI STATE 指示功能。因此，SPCR 寄存器中的 DXENA 寄存器字段不受支持。
 - 功能设备仿真器中不存在上升/下降沿触发活动的概念。它仅支持脉冲类型的输入；因此，不支持以下寄存器字段的功能：FSXP、FSRP、CLKXP、CLKRP（全部在 PCR 寄存器中）和 CLKSP（在 SRGR 寄存器中）。
 - 不支持通用 I/O 模式。因此，不支持以下寄存器字段的功能：XIOEN、RIOEN、CLKS_STAT、DX_STAT 和 DR_STAT（全部在 PCR 寄存器中）。
 - 不支持数据延迟的概念。因此，它不具备 RCR 寄存器中的 RDATDLY 功能，以及 XCR 寄存器中的 XDATDLY 功能。接受到 FSX/FSR 后，数据传输/接收将立即开始。
- **EMIF 限制**
 - 在设备功能仿真器中，所有 DSP 中的数据传输都会建成一次性传输模型，也就是说，所有的数据传输均被视为在零周期延迟下进行的。因此，SDEXT 寄存器中的所有寄存器字段的功能都不可用。
 - 它不支持将 SDTIM 寄存器中的 COUNTER 字段用作通用计时器。
 - 设备功能仿真器不支持刷新 SDRAM。因此，与 SDRAM 刷新操作有关的所有寄存器字段的功能均不可用。
- **高速缓存 限制**
 - 设备功能仿真器中的高速缓存被建为 无数据高速缓存模型；也就是说，L1P、L1D 和 L2 高速缓存不存储数据，仅标记信息。任何对 L1D 和 L2 的修改将直接反映在实际存储器中。因此，对高速缓存的操作结果，例如高速缓存清理和高速缓存失效等可能会不同。我们始终推荐先使用重设命令，然后再使用重新加载命令，最后使用重新启动命令，以便实现正确的操作行为。
- **处理保留空间**
 - 在设备功能仿真器中，当您写入保留位置时，该操作会被视为将 0x0000 0000 写到该位置。同样，当您从保留位置读取数据值时，只会读取到 0x0000 0000。
- **ROM 存取行为**
 - 当您写到指定为 ROM (0x0186 0000 - 0x0187 FFF0) 区域时，6713 设备功能仿真器不会生成错误信息。

如前文所述，这些配置上的模拟并不是精确到周期的。

3.3 C6201/C6202/C6203/C6204/C6205/C6701 设备仿真器

这些模拟器将为 CPU 核心、内部程序（备用高速缓存）和数据存储器控制器、DMA、EMIF、中断选择器、McBSP 和计时器建模。CPU 与内部程序存储器（备用高速缓存）的 DMA 以及内部数据存储器存取之间的仲裁也将被建模。DMA 支持设备具有的所有功能，例如四通道、资源仲裁和优先级配置、分离通道操作和 DMA 存取外设置的范围。使用引脚连接功能或 McBSP XBAR 可以设置 McBSP 的外部激励。有关详细信息，请分别参阅第 4.9 部分与第 4.6 部分。

3.3.1 支持的功能

C6x0x 设备仿真器具有精确到周期的 CPU 和精确到周期的计时器模块。同时也对 L1P、内部程序和 EMIF 子系统进行了精确建模。DMA 子系统对于 C6201 设备仿真器是精确的。

支持引脚连接、端口连接和分析事件功能。

这些配置所支持的引脚列表和可连接到文件的存储器范围如第 2.1 部分。通过使用仿真器分析插件，这些仿真器配置便可跟踪多个目标事件，并允许计算事件数量或中断这些事件。请参阅 Code Composer Studio IDE 联机文档，以获得有关事件列表以及如何使用这些事件的详细信息。

3.3.2 已知限制

- **常规**
 - 未对 HPI 建模。
 - 未对 XBUS 建模。
 - 未对 GPIO 建模。
- **DMA**
 - 不支持用于为 HPI 请求提供服务的辅助通道（通道 5）。

- 在 C6203、C6204 和 C6205 中，用于保存来自高性能源（例如，内部存储器）的数据的 DMA FIFO 由全部四个通道共享（尽管应该是每个通道所专用的）。
- 不支持同步事件 DSPINT 和 SDRAM_INT。
- 只有启用了同步事件时才可以捕捉到它们。
- EMIF
 - 不会为这些配置建立 SDRAM 中断模型。
- 周期精确性
 - 不保证周期总数的精确性。

3.4 C6211/C6711/C6712/C6713 设备精确到周期的仿真器

这些仿真器将为 CPU 核心、L1P 和 L1D 高速缓存、L2 高速缓存/SRAM 存储器、EDMA、EMIF、中断选择器、McBSP 和计时器建模。另外，C6713 仿真器还将为 McASP 模块建模。EMIF 的外部存储器也将被建模，以便根据 EMIF 的设置为对应的 CE 空间模拟存储器的预期行为。除了以上部分，C6713 仿真器还将为 S2、ROM 和 ROM 修补建模。

3.4.1 支持的功能

该仿真器也支持启动加载功能。

C6x1x 设备精确到周期的仿真器具有精确到周期的 CPU 和精确到周期的计时器模块。同时也对 L1 和 L2 进行了精确建模。EDMA 子系统和 EMIF，虽然不是精确建模，但会按照设计规格按比例精确周期，这取决于传输类型、存储器参数等。

使用 RTDX 和 DSP/BIOS 实时分析的应用可在这些仿真器上运行。另外，引脚连接和端口连接功能可用于设置应用所需的外部激励，以进行模拟。

这些配置所支持的引脚列表和可连接到文件的存储器范围如第 2.1 部分。这些仿真器配置可跟踪多个目标事件，并允许计算事件数量或中断这些事件。请参阅 Code Composer Studio IDE 联机文档，以获得有关事件列表以及如何使用这些事件的详细信息。

3.4.2 已知限制

- 常规
 - 未对 HPI 建模。
 - 未对 GPIO 建模。
- DMA
 - 不支持同步事件 DSPINT 和 SDRAM_INT。
- EMIF
 - 不支持 SDRAM 中断。
- 高速缓存
 - 不支持 L2CLEAN 寄存器的功能。

3.5 C6411/C6412/C6414/C6415/C6416/DM642 设备精确到周期的仿真器

这些仿真器将为 CPU 核心、L1P 和 L1D 高速缓存、L2 高速缓存/SRAM、EDMA、EMIFA、中断选择器、McBSP 和计时器建模。EMIF 的外部存储器也将被建模，以便根据 EMIF 的设置为对应的 CE 空间模拟存储器的预期行为。EMIFB 和 CP/VCP 不支持 C6411/DM642 设备精确到周期的仿真器。

3.5.1 支持的功能

C641x 设备精确到周期的仿真器具有精确到周期的 CPU 和精确到周期的计时器模块。L1 和 L2 高速缓存也将被精确建模。EDMA 子系统和 EMIF 将按照设计，根据不同的参数按比例地运行周期。

使用 RTDX 和 DSP/BIOS 实时分析的应用可在这些仿真器上运行。另外，引脚连接和端口连接功能可用于设置应用所需的外部激励，以进行模拟。

这些配置所支持的引脚列表和可连接到文件的存储器范围如第 2.1 部分。这些仿真器配置可跟踪多个目标事件，并允许计算事件数量或中断这些事件。请参阅 Code Composer Studio IDE 联机文档，以获得有关事件列表以及如何使用这些事件的详细信息。

3.5.2 已知限制

- 常规
 - 未对 HPI/PCI 建模。
 - 未对 Utopia 建模。
 - 不支持 GPIO。
- EDMA
 - 不支持 XDMA 中的事件极性。因此，仅获取处于高活动状态的事件。
 - 不支持“推数据传输”功能。
 - 未对 DSPINT、GPIOINT、SD_INTA、SD_INTB、PCI、UREVT 和 UXEVT 建模。
- 高速缓存
 - L2 不能提交所有优先级的传输请求 (TR)，它只能提交优先级为 0 的请求。
 - L1P 至 L2SRAM 的管理缺失的周期效果不准确。
- TCP/VCP 协处理器
 - 不支持暂停/恢复条件。

配置仿真器

您可以通过 Code Composer Studio Setup 程序来配置仿真器以提供不同的功能。但是，要修改高级选项则需要修改基本配置文件（请参阅第 4.11 部分）。

主题	页
4.1 设置资源冲突检测模式	32
4.2 设置保留存储器存取检测模式	32
4.3 设置启动加载	33
4.4 设置 EMIF 和 CPU 时钟	33
4.5 启用回卷功能	34
4.6 设置 McBSP XBAR	34
4.7 设置 McASP XBAR	35
4.8 设置最大存储器使用量限制	36
4.9 引脚连接的文件格式	36
4.10 端口连接的文件格式	37
4.11 基本配置文件	38

4.1 设置资源冲突检测模式

CPU 资源冲突检测功能允许用户查找在使用 CPU 寄存器和功能单元时出现的任何资源问题。如果您确信代码中不存在资源冲突问题，则可以关闭此功能以提高模拟速度。

此功能可以通过 Code Composer Studio Setup 程序配置，方法是：在处理器属性窗口中，从 Detect CPU Resource Conflicts（检测 CPU 资源冲突）项中设置以下选项之一：

- **Yes（检测）。** 仿真器会检测所有资源冲突，并将其标记为错误。出现错误时，仿真器会暂停并将控制权返还给您。用户可以选择更正错误、复位程序和再次运行；或是跳过错误，继续执行。如果跳过错误，则会出现预警错误信息，单击 Run（运行）选项以继续模拟。

注：

YES（是）为默认模式。如果在基本配置文件中未作任何指定（请参见第 4.11 部分），仿真器将以此种模式执行。

- **No（否）。** 仿真器未检测到任何资源冲突。
- **Create Log（创建日志）。** 仿真器会检测所有资源冲突，但不会将其标记为错误。相反，错误会被写入到文件中。要指定文件位置，请使用 Log File（日志文件）项。
- **Log File: 完全限定的文件名。** 如果未指定文件名，则在“文件”模式中生成的错误会被记录在名为 resource_errors.log 的文件中，它位于 CCStudio\dri vers 目录下。
当重新调用 Code Composer Studio IDE 时，错误日志文件将被覆盖。但是，如果在一个 CCStudio 调用的一次或多次运行期间出现了多个错误，则所有的错误会被写入到文件中。

4.2 设置保留存储器存取检测模式

默认情况下，所有仿真器上都启用了对保留存储器存取的检测。此功能可以通过 Code Composer Studio Setup 程序配置，方法是：在处理器属性窗口中，从 Detect Reserved Memory Access（检测保留存储器存取）项中设置以下选项之一：

- **Yes（检测）。** 当程序存取了保留存储器位置时，仿真器将暂停。它会显示一个错误，指明存取的性质和地址。这是默认选项。
- **No（不检测）。** 仿真器不报告任何对保留存储器的存取。在这种模式下，写入到保留存储器无效，同时，读取返回 0。

注：

任何对保留存储器位置的调试存取（通过存储器窗口）不会导致错误。调试写无效，调试读始终返回 0。

- **Create Log（创建日志）。** 仿真器会将所有的错误信息传送到一个文件，而不暂停模拟。要指定保存错误的文件，请使用 Log File（日志文件）项。
- **Log File: 完全限定的文件名。** 如果未指定文件名，则在“文件”模式中生成的错误会被记录在名为 res_mem_access_errors.log 的文件中，它位于 CCStudio\dri vers 目录下。
当重新调用 Code Composer Studio IDE 时，错误日志文件将被覆盖。但是，如果在一个 CCStudio 调用的一次或多次运行期间出现了多个错误，则所有的错误会被写入到文件中。

4.3 设置启动加载

4.3.1 C6x0x 设备仿真器中的启动加载。

以下是在 C6x0x 设备仿真器中支持的启动模式：

- ROM_8BIT
- ROM_16BIT
- ROM_32BIT

通过 Code Composer Studio Setup 程序界面来配置 C6x0x 设备仿真器的启动模式。您必须在所需的仿真器基本配置文件中手动添加一个 BOOTSRC 条目（请参阅第 4.11 部分）以使用启动加载功能。

例如，要在 C6202 设备仿真器中选择 ROM_8BIT 启动模式，基本配置文件必须包括以下内容以启用启动加载。

```
MODULE C6202;  
...  
    BOOTSRC ROM_8BIT;  
END C6202;
```

如果缺少 BOOTSRC 标志或设置为 NONE，则不会进行启动加载。如果启动模式被设置为上面列出的三个值中的任意一值，DMA 会将 64K 字节从 EMIF CE1（其地址在 MAPO 中为 0x01000000，在 MAP1 中为 0x01400000）复制到地址 0。除了启动模式，您可以通过仿真器基本配置文件来配置位于地址 0 的存储器类型。要进行配置，基本配置文件应将 MEMO 开关设置为有效的存储器类型，您可以从下列中选择：

- ONCHIP
- SDRAM_8BIT
- SDRAM_16BIT
- SDRAM_32BIT
- SBSRAM_CLK2
- SBSRAM_CLK1

如果没有指定值，则在默认情况下，位于地址 0 的存储器将被设置为 ONCHIP。

4.3.2 C64x 设备精确到周期的仿真器/设备功能仿真器中的启动加载。

在 C64x 设备仿真器中，启动加载过程（如果已从 Code Composer Studio Setup 程序启用）会将 1K 字节（256 字）复制到位于地址 0（即，此情况中的内部存储器）的存储器。

要在 C6411/DM642/C6414/C6415/C6416 上启用启动加载，则 Code Composer Studio Setup 程序中 BOOT Mode（启动模式）应设置为下面任一个值：

- NONE
- EMIFB

如果没有指定值，或设置为 NONE，或通过基本配置文件设置为其它任何无效值（请参阅第 4.11 部分），“启动模式”将默认设置为 NONE。这意味着不会进行启动加载。将 EMIFB 指定为“启动模式”将导致启动加载过程通过 EDMA 将 256 个字从 EMIFB CE1 空间 (0x64000000) 复制到内部存储器地址 0。

启动模式 QUICK 可以进行简单的存储器复制（不像启动模式 EMIFB 那样通过 EDMA），能够将 256 个字从 EMIFA CE1 空间（即，0x90000000）复制到内部存储器。QUICK 启动模式因而更快。

4.4 设置 EMIF 和 CPU 时钟

C6000 仿真器上的 EMIF 与 CPU 时钟的比率必须按照相应的 DSP 开发底板 (DSK/TEB/EVM) 的特点进行设置。为了确保开发底板上的周期与仿真器上的周期一致，此步骤必须执行。

您必须在 Code Composer Studio Setup 程序中指定正确的 EMIF 和 CPU。

例如，考虑使用 CPU 时钟频率为 600MHz，EMIF 时钟频率为 100MHz 的 C6416 DSK。为了使 C6416 设备精确到周期的仿真器的特点与 DSK 匹配，应通过 Code Composer Studio Setup 程序将 EMIF 时钟频率 (MHz) 和 CPU 时钟频率 (MHz) 分别设置为 100 和 600。

注：

如果未完成此步骤，则对于外部存储器存取，仿真器和开发底板之间的周期会有很大差异。

4.5 启用回卷功能

通过使用回卷功能，您可以查看过去在仿真器上执行的应用的历史记录。这样可以减少调试应用所花的时间。目前，仅 CPU 仿真器支持此选项。默认情况下，此功能是被禁用 (OFF) 的，但可以通过 Code Composer Studio Setup 程序从 CPU 仿真器中将其启用 (ON)。

通过 Code Composer Studio Setup 程序可以配置此功能，方法是：在处理器属性窗口中对 Rewind (回卷) 项的相应选项进行设置。

启用回卷功能后，Code Composer Studio Setup 程序会为您提供两个选项以选择回卷跟踪的位置：

- In Memory (存储器中，默认)
- On Disk (磁盘上)

请参阅《回卷功能 用户指南 (SPRU713)》，以获得有关为仿真器配置回卷功能的详细信息。

4.6 设置 McBSP XBAR

您可以使用仿真器提供的机制使两个 McBSP 互连，以测试和验证为串行传输所写的代码。XBAR (crossbar) 的用途是测量台，您可以对其编程以设置需要的连接。

通常，对一个 McBSP 编程用于传输，而对另一个 McBSP 编程用于接收。因此，对应的 McBSP 的 DX 和 DR 引脚必须互相挂钩。同样，时钟和帧同步信号也必须互连。仿真器中的 XBAR 组件支持这种连接的规范和实施。

XBAR 连接是在 XBAR 数据文件中指定的。第 4.6.2 部分详细描述了文件格式。XBAR 数据文件的路径和文件名必须包含在仿真器基本配置文件中（请参阅第 4.11 部分），该配置文件可以从 Code Composer Studio Setup 程序中选择。随后，当调用 Code Composer Studio IDE 时，XBAR 功能将生效。

4.6.1 如何写 XBAR 文件

表 4-1 中列出了 XBAR 文件支持的引脚名 所示：

表 4-1. XBAR 文件引脚名

引脚名	指示的引脚	寄存器
MCBSP0: DX	DX	McBSP0
MCBSP1: DX	DX	McBSP1
MCBSP2: DX	DX	McBSP2
MCBSP0: DR	DR	McBSP0
MCBSP1: DR	DR	McBSP1
MCBSP2: DR	DR	McBSP2
MCBSP0: CLKX	CLKX	McBSP0
MCBSP1: CLKX	CLKX	McBSP1
MCBSP2: CLKX	CLKX	McBSP2
MCBSP0: CLKR	CLKR	McBSP0
MCBSP1: CLKR	CLKR	McBSP1
MCBSP2: CLKR	CLKR	McBSP2
MCBSP0: FSX	FSX	McBSP0
MCBSP1: FSX	FSX	McBSP1
MCBSP2: FSX	FSX	McBSP2
MCBSP0: FSR	FSR	McBSP0
MCBSP1: FSR	FSR	McBSP1
MCBSP2: FSR	FSR	McBSP2

1. 连接是通过源引脚-目标引脚的配对规范来指定的。驱动程序驱动的实体关系是在与下列内容相同的行上指定的：

- MCBSP0: FSX > MCBSP1: FSR
- MCBSP0: DX > MCBSP1: DR
- MCBSP0: CLKX > MCBSP1: CLKR

2. 文件语法不支持注释。两个指定行之间不应有空行。

3. 您可以按任何顺序提及连接，但确保每对引脚都指定了驱动程序驱动的连接。例如，当提及 MCBSP1: DX → MCBSP0: DR、MCBSP1: CLKX → MCBSP0: CLKR、MCBSP1: FSX → MCBSP0: FSR 连接时，请遵守有效的 XBAR 连接：

有效 XBAR 连接

MCBSP1: DX > MCBSP0: DR

MCBSP1: CLKX > MCBSP0: CLKR

MCBSP1: FSX > MCBSP0: FSR

无效 XBAR 连接

MCBSP1: DX > MCBSP1: CLKX

MCBSP1: FSX > MCBSP0: DR

MCBSP0: CLKR > MCBSP0: FSR

4. 确保文件的结尾没有空行，因为这样可能会导致问题。

注：

向后兼容性支持旧的 McBSP XBAR 文件格式，但建议用户使用新的格式。

4.6.2 要选择的配置文件格式。

McBSP XBAR 不能通过 Code Composer Studio Setup 程序界面进行配置。您必须在所需的仿真器基本配置文件中手动添加一个 MCBSP_XBAR_FILE 条目（请参阅第 4.11 部分）。

对于 C6x0x 设备，如下添加 MCBSP_XBAR_FILE 条目：

```

MODULE TB;
  MCBSP_XBAR_FILE <path_of_xbar_file_name>;
END TB;
  
```

另一方面，对于 C64xx 设备，如下添加 MCBSP_XBAR_FILE 条目：

```

MODULE C64xx;
...
  MCBSP_XBAR_FILE <path_of_xbar_file_name>; // path of xbar data file
END C64xx;
  
```

值 <path_of_xbar_file_name>.dat 可以指示 XBAR 数据文件的路径和文件名。例如：

```

MCBSP_XBAR_FILE C:\ccstudio\drivers\mcbasp_xbar.dat;
  
```

4.7 设置 McASP XBAR

仿真器提供了可以连接两个 McASP 的机制，正如其所连接两个 McBSP 的方式。构成 XBAR 文件行的连接格式如下：

SRC_DEV : *SRC_PIN* > *DST_DEV* : *DST_PIN*

其中

SRC_DEV 是传出数据的设备。

SRC_PIN 是发送数据的引脚。

DST_DEV 是数据传入的设备。

DST_PIN 是接收数据的引脚。

以下是需要注意的一些限制：

- 支持的外设：McASPO
- 支持的引脚：AXR0-AXR15、AFSR、AFSX
- 不支持的引脚：AMUTEIN、AMUTE、ACLKR、ACLKX、AHCLKR 和 AHCLKX

例如，XBAR 文件可包含以下连接：

```

MCASPO: AXR8 > MCASPO: AXR0
MCASPO: AXR9 > MCASPO: AXR1
MCASPO: AXR10 > MCASPO: AXR2
MCASPO: AXR11 > MCASPO: AXR3
MCASPO: AXR12 > MCASPO: AXR4
MCASPO: AXR13 > MCASPO: AXR5
  
```

```
MCASPO: AXR14 > MCASPO: AXR6
MCASPO: AXR15 > MCASPO: AXR7
MCASPO: AFSX > MCASPO: AFSR
```

如果不需要互连，可将文件留空。

4.7.1 要选择的配置文件格式。

McASP XBAR 不能通过 Code Composer Studio Setup 程序界面进行配置。您必须在所需的仿真器基本配置文件中手动添加一个 MCASP_XBAR_FILE 条目（请参阅第 4.11 部分）。

对于 C6x0x 设备，如下添加 MCASP_XBAR_FILE 条目：

```
MODULE TB;
  MCBSP_XBAR_FILE <path_of_xbar_file_name>;
END TB;
```

另一方面，对于 C64x 设备，如下添加 MCASP_XBAR_FILE 条目：

```
MODULE C64xx;
...
  MCBSP_XBAR_FILE <path_of_xbar_file_name>; // path of xbar data file
END C64xx;
```

值 <path_of_xbar_file_name> 可以指示 XBAR 数据文件的路径和文件名。例如：

```
MCASP_XBAR_FILE C:\ccstudio\drivers\mcasep_xbar.dat;
```

4.8 设置最大存储器使用量限制

默认的最大存储器使用量为 64MB，您可以通过在基本配置文件中添加以下行以改写默认值（请参阅第 4.11 部分）：

```
MEM_USAGE_LIMIT <max_mem_usage_in_MB>;
```

例如，运行需要占用设备 50 至 100 MB 存储器空间的测试方案，默认的 64MB 肯定不够用，所以您需要改写此值。例如，对于 C6416 设备精确到周期的仿真器配置，要将最大存储器限制设置为 100 MB，请转到基本配置文件中的 MODULE C6416 部分，添加条目 MEM_USAGE_LIMIT 100，如下所示。

```
MODULE C6416;
...
  MEM_USAGE_LIMIT 100;
END C6416;
```

MEM_USAGE_LIMIT 不能通过 Code Composer Studio Setup 程序界面进行配置。您必须在所需的仿真器基本配置文件中手动添加一个 MEM_USAGE_LIMIT 条目。

4.9 引脚连接的文件格式

仿真器允许您模拟并监控外部中断信号。

引脚连接工具让您可以为选择的外部中断指定其要发生的间隔。

要模拟外部中断，请执行以下步骤：

1. 创建用于指定中断间隔的数据文件。
2. 启动引脚连接工具。在 Code Composer Studio 的 Tools (工具) 菜单中，选择 Pin Connect (引脚连接)。
3. 连接数据文件到外部中断引脚。
4. 加载程序。
5. 运行程序。

4.9.1 设置输入文件

要模拟外部中断，您必须首先创建用于指定中断间隔的数据文件。中断间隔表达为 CPU 时钟周期的功能。模拟从第一个周期开始。中断会在每个指定的周期发生。

数据文件必须包含 CPU 时钟周期参数，格式如下：

```
[clock_cycle, logic_value] [rpt {n}|EOS}]
```

其中

<i>clock_cycle</i>	CPU 时钟周期参数指定了中断发生的间隔。时钟周期可被指定为绝对周期或相对周期。
<i>logic_value</i>	逻辑值参数仅对波型引脚有效（例如，C6201 仿真器中的 FSX0 引脚）。该值（0 或 1）必须强制引脚值在相应周期变高或变低。逻辑值 0 使引脚值变低，逻辑值 1 使引脚值变高。 例如： [12,1] [56,0] [78,1] 如果连接到 C6201 中的引脚 FSX0，这会使引脚值在第 12 个周期变高，在第 56 个周期变低，然后在第 78 个周期再变高。
<i>rpt</i>	按照固定的次数重复同一模式。
<i>n</i>	为重复次数指定正整数值。
EOS	重复相同的模式直到模拟结束。

4.9.2 绝对时钟周期

要使用绝对时钟周期，周期值必须代表实际 CPU 时钟周期，并在此周期中模拟中断。例如，

```
12 34 56
```

在 CPU 的第 12、第 34 和第 56 个时钟周期模拟中断。未按时钟周期值执行任何操作；中断完全按照所写的时钟周期值发生。

4.9.3 相对时钟周期

您也可以选择一个相对于最后一个事件发生时间的时钟周期。时钟周期前的加号 (+) 表示该值与它之前的所有周期值相加。例如，

```
12 +34 55
```

在此示例中，在第 12、第 46 (12 + 34) 和第 55 个 CPU 时钟周期总共模拟了三次中断。您可以在数据文件中混用绝对周期和相对周期。

4.9.4 模式重复固定次数。

您可以格式化数据文件，以按照固定的次数重复特殊的模式。例如：

```
5 (+10 +20) rpt 2
```

括号中的值代表要重复的部分。因此，在第 5 个 CPU 周期模拟了中断，然后是在第 15 个 (5+10)、第 35 个 (15+20)、第 45 个 (35+10) 和第 65 个 (45+20) CPU 时钟周期模拟了中断。

4.9.5 重复至模拟结束 (EOS)

要在整个模拟期间重复同一模式，请向行中添加字符串 EOS。例如：

```
10 (+5 +20) rpt EOS
```

在第 10 个、第 15 个 (10+5)、第 35 个 (15+20)、第 40th (35+5) 和第 60 个 (40+20) CPU 周期生成中断，并以此类推，重复执行该模式直到模拟结束。

注：

引脚连接文件中不支持注释。

4.10 端口连接的文件格式

端口连接文件包含一个或多个行。每行包含少于 80 个的字符以表示一个数据值。端口连接文件中的数据被解释为十六进制数据，您可在数据前指定 0x 或不作指定。

```

12346666
33449999
: 这是一个注释行
5655cccc: 这是一个注释句
89897f7f
  
```

端口连接读文件允许注释。注释应以分号 (;) 开头，如示例文件中所示。

4.11 基本配置文件

在 Code Composer Studio 安装目录下，驱动程序子文件夹中的每个模拟驱动程序都具有相应的配置文件 (.cfg)。这些文件可为对应的仿真器驱动程序所支持的功能保存默认值。仅在您需要更改某些配置的默认值，而又不能通过 Code Composer Studio Setup 程序进行配置时才可以更改这些值，例如，回卷功能的高级选项，或者最大存储器使用量限制。

性能数据

表 5-1 显示了仿真器对于不同设备配置的性能数据。这些数字从计算机上收集而来，计算机的配置为 2.4-GHz Intel® Pentium® 4 处理器和 512 MB 内存。

表 5-1. C6000 仿真器的性能数据

仿真器配置	应用	默认设置下的运行速度		关闭了资源冲突检测功能时的运行速度	
		KIPS ⁽¹⁾	KCPS ⁽¹⁾	KIPS ⁽¹⁾	KCPS ⁽¹⁾
C62xx/C67xx CPU 精确到周期的仿真器	GSM 加强全速率声码器 (GSMEFR)	3800	-	5600	-
C64xx CPU 精确到周期的仿真器	GSMEFR	2700	-	3600	-
C641x/DM642 设备功能仿真器	GSMEFR	2700	-	2750	-
C6713 设备功能仿真器	GSMEFR	2600	-	3300	-
C6x0x 设备仿真器	Reed-Solomon 编码器和解码器	41	188	41	188
C6211/C6713 设备精确到周期的仿真器	GSMHR	74	644	74	644
	GSMEFR	9	195	9	195
	音频性能应用 - PA3	100	294	102	311
C641x/DM642 设备精确到周期的仿真器	GSMHR	101	337	102	355
	GSMEFR	807	2344	892	2354
	视频及成像应用 - MPEG2 解码器	885	223	984	248
	视频及成像应用 - H.263 解码器	776	261	851	279

(1) KIPS = 每秒千指令数，KCPS = 每秒千周期

注:

- 如果因应用的特性而造成性能数据差别巨大时，系统将为多个应用提供性能数据。
- 对于设备精确到周期的仿真器，CPU 负载对于每秒周期的数据影响显著。低 CPU 负载意味着对于同样数量的指令，更多的周期花费在了 CPU 停顿上，因此增加了每秒的周期数。

周期精确性

本章介绍了如何使用应用的基准套件对 TMS320C6000 仿真器的周期精确度进行验证。

主题	页
6.1 C6000 仿真器基准	42
6.2 关于周期精确度的注意事项	43

6.1 C6000 仿真器基准

使用应用的基准套件对 TMS320C6000 仿真器的周期精确度进行了验证。测量设备精确到周期的仿真器分为以下几类进行：

- CPU + L1 高速缓存 + SRAM
- CPU + L1 高速缓存 + L2 高速缓存 + EMIF
- CPU + L1 高速缓存 + L2 高速缓存 + EDMA + EMIF
- 使用以上所有模型和外设的完全应用。

详细的基准数据如表 6-1。

表 6-1. C6000 仿真器的基准数据

用于基准的应用/内核	标准硬件	仿真器配置	周期数的百分比差异
片上存储器存取 (CPU + L1 高速缓存 + SRAM)			
图形分析 - 直方图计算	C6416 DSK	C6416 设备精确到周期的仿真器	1.35
图像过滤 - 3x3 舍入相关	C6416 DSK	C6416 设备精确到周期的仿真器	0
图形分析 - 直方图计算	C6713 DSK	C6713 设备精确到周期的仿真器	2.47
图像过滤 - 3x3 舍入相关	C6713 DSK	C6713 设备精确到周期的仿真器	-0.37
单精度快速傅里叶变换 - SP_FFT	C6713 DSK	C6713 设备精确到周期的仿真器	-1.64
单精度矩阵乘法 - SP_MATMUL	C6713 DSK	C6713 设备精确到周期的仿真器	-0.79
片上与片外存储器存取 (CPU + L1 高速缓存 + L2 高速缓存 + EMIF)			
图形分析 - 直方图计算	C6416 DSK	C6416 设备精确到周期的仿真器	1.56
图像过滤 - 3x3 舍入相关	C6416 DSK	C6416 设备精确到周期的仿真器	-10.89
图形分析 - 直方图计算	C6713 DSK	C6713 设备精确到周期的仿真器	-8.83
图像过滤 - 3x3 舍入相关	C6713 DSK	C6713 设备精确到周期的仿真器	12.66
单精度 FIR 过滤 (通用) - SP_FIR_GEN(L2SRAM)	C6713 DSK	C6713 设备精确到周期的仿真器	9.78
单精度 FIR 过滤 (通用) - SP_FIR_GEN(L2CACHE)	C6713 DSK	C6713 设备精确到周期的仿真器	-11.67
单精度快速傅里叶变换 - SP_FFT	C6713 DSK	C6713 设备精确到周期的仿真器	-35.31
单精度矩阵乘法 - SP_MATMUL	C6713 DSK	C6713 设备精确到周期的仿真器	-9.76
单精度加权矢量和 - SP_W_VEC(L2SRAM)	C6713 DSK	C6713 设备精确到周期的仿真器	11.29
单精度加权矢量和 - SP_W_VEC(L2CACHE)	C6713 DSK	C6713 设备精确到周期的仿真器	-1.58
使用 DMA 的存储器存取 (CPU + L1 高速缓存 + L2 高速缓存 + EDMA + EMIF)			
图形分析 - 直方图计算	C6416 DSK	C6416 设备精确到周期的仿真器	1.80
图像过滤 - 3x3 舍入相关	C6416 DSK	C6416 设备精确到周期的仿真器	9.49
图形分析 - 直方图计算	C6713 DSK	C6713 设备精确到周期的仿真器	0.87
图像过滤 - 3x3 舍入相关	C6713 DSK	C6713 设备精确到周期的仿真器	-0.55
单精度 FIR 过滤 (通用) - SP_FIR_GEN	C6713 DSK	C6713 设备精确到周期的仿真器	9.67
单精度快速傅里叶变换 - SP_FFT	C6713 DSK	C6713 设备精确到周期的仿真器	25.22
单精度矩阵乘法 - SP_MATMUL	C6713 DSK	C6713 设备精确到周期的仿真器	9.58
单精度加权矢量和 - SP_W_VEC	C6713 DSK	C6713 设备精确到周期的仿真器	6.64
完全应用 (CPU + L1 高速缓存 + L2 高速缓存 + EDMA + EMIF + 外设 (McBSP, McASP))			
视频及成像应用 - MPEG2 解码器	C6416 DSK	C6416 设备精确到周期的仿真器	-0.12
视频及成像应用 - H.263 解码器	C6416 DSK	C6416 设备精确到周期的仿真器	1.46
音频性能应用 - PA3	C6713 - PA3 参考板	C6713 设备精确到周期的仿真器	-4.66
语音应用 - PCM 声音通道	存取通信处理器评估平台 (C6416 子卡)	C6416 设备精确到周期的仿真器	1.02

6.2 关于周期精确度的注意事项

当比较仿真器和开发底板 (DSK/EVM/TEB) 之间的周期时, 确保满足以下条件:

1. 选择的设备精确到周期的仿真器应适合相应的开发底板。例如, 对于 TMS320C6416 开发底板, 选择 C6416 设备精确到周期的仿真器。有关仿真器不同类别的详细信息, 请参阅在 *Code Composer Studio* 中选择合适的仿真器 (文献编号 SPRA864)。
2. 仿真器上的 EMIF 与 CPU 时钟的比率与其各自的开发底板的比率相匹配。请参阅第 4.4 部分以获取为仿真器配置正确的 EMIF 与 CPU 时钟比率的详细信息。
3. 仿真器的 EMIF 设置与开发底板上的 EMIF 设置相匹配。在 *emif_init gel* 文件中或在程序中为仿真器和开发底板上的 EMIF 寄存器设置相同的值即可确保它们相匹配。
4. 周期差异可能会由于仿真断点处理而扩大。要获得更精确的数字, 请使用计时器, 而不要使用分析器。

重要声明

德州仪器 (TI) 及其下属子公司有权在不事先通知的情况下, 随时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, 并有权随时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 保证其所销售的硬件产品的性能符合 TI 标准保修的适用规范。仅在 TI 保修的范围内, 且 TI 认为有必要时才会使用测试或其它质量控制技术。除非政府做出了硬性规定, 否则没有必要对每种产品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 产品或服务的组合设备、机器、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 的数据手册或数据表, 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。在复制信息的过程中对内容的篡改属于非法的、欺诈性商业行为。TI 对此类篡改过的文件不承担任何责任。

在转售 TI 产品或服务时, 如果存在对产品或服务参数的虚假陈述, 则会失去相关 TI 产品或服务的明示或暗示授权, 且这是非法的、欺诈性商业行为。TI 对此类虚假陈述不承担任何责任。

可访问以下 URL 地址以获取有关其它 TI 产品和应用解决方案的信息:

产品

放大器	http://www.ti.com.cn/amplifiers
数据转换器	http://www.ti.com.cn/dataconverters
DSP	http://www.ti.com.cn/dsp
接口	http://www.ti.com.cn/interface
逻辑	http://www.ti.com.cn/logic
电源管理	http://www.ti.com.cn/power
微控制器	http://www.ti.com.cn/microcontrollers

应用

音频	http://www.ti.com.cn/audio
汽车	http://www.ti.com.cn/automotive
宽带	http://www.ti.com.cn/broadband
数字控制	http://www.ti.com.cn/control
光纤网络	http://www.ti.com.cn/opticalnetwork
安全	http://www.ti.com.cn/security
电话	http://www.ti.com.cn/telecom
视频与成像	http://www.ti.com.cn/video
无线	http://www.ti.com.cn/wireless

邮寄地址: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2006, Texas Instruments Incorporated