

# TI Glossary



Literature Number: ZHCZ004B  
December 2013–Revised February 2014

## 关于这份术语表

---

---

---

这份术语表列出并解释术语、首字母缩略词和定义。

**1 A**

**A0-An** 地址引脚 — 数据和程序存储器或 I/O 器件的外部地址引脚。

**绝对地址[absolute address]** — 永久分配给存储器位置的地址。另请参见 *符号地址*。

**绝对列表器[absolute lister]** — 将链接的文件接受为输入并创建 .abs 文件作为输出的调试工具。这些 .abs 文件可经汇编后产生显示对象代码绝对地址的列表。在没有该工具时，可通过大量手动操作准备绝对列表。

**存取阶段[access stage]** — 主处理器的获取、执行、存取 (FEA) 管道的第三个可选阶段，在此过程中会发生存储器存取（加载或存储操作）。

**累加器 [accumulator , ACC]** — 临时存储算术逻辑单元 (ALU) 运算结果并为随后的 ALU 运算提供输入的寄存器。对 ACC 的访问可分为两部分：累加器高字节 (ACCH) 和累加器低字节 (ACCL)。

**累加器缓冲器 [accumulator buffer, ACCB]** — 临时存储累加器 (ACC) 内容的寄存器。ACCB 具有返回算术逻辑单元 (ALU) 的直接路径，并可与 ACC 一起进行算术和逻辑处理。

**累加器高字节 [accumulator high byte, ACCH]** — 存储在累加器 (ACC) 中的最高有效位。另请参见 *累加器*。

**累加器低字节 [accumulator low byte, ACCL]** — 存储在累加器 (ACC) 中的最低有效位。另请参见 *累加器*。

**确认 [acknowledge, ACK]** — 用于表示接收到的发送信息完整或没有错误，或者表示接收站已经为接受传输做好准备的传输控制字符。

**主动电流辅助和模拟旁路 [active current assist and analog bypass, ACB]** — 一个主动电流模拟旁路控制电路接收并使用电流限制信息、电压错误信息，以及压降信息，以决定适合于降压稳压器电路当前工作状态的辅助电流值。

**主动噪声消除 [active noise cancellation, ANC]** — 消除背景噪声。

**激活时间[active time]** — 显示帧不处于消隐状态的时间间隔。像素被显示的时间间隔。另请参见 *消隐*。

**活动窗口[active window]** — 当前选定用于移动、调整大小、编辑、关闭或一些其它功能的窗口。

**地址** — 存储在存储器中的程序代码或数据的逻辑位置。

**寻址模式[addressing mode]** — 指令解释其操作数来获取所需数据的方法。

**寻址阶段[address stage]** — 并行处理器的获取、寻址、执行 (FAE) 管道的第二阶段，在此期间，计算出地址并将其提供给交叉开关。

**地址单元[address unit]** — 并行处理器上的硬件，在每个周期中计算位地址。每个并行处理器具有两个地址单元：一个全局地址单元和一个本地地址单元。

**地址单元算术[address unit arithmetic]** — 并行处理器对本地和全局地址单元的使用，与数据单元并行执行通用算术。计算出的地址不用于存储器存取，而是存储在目标寄存器中。

**地址可见性 [address visibility , AVIS] 位** — 允许内部程序地址显示在外部地址引脚上的位字段。这样可实现对内部程序地址的跟踪，并且在中断向量驻留在片上存储器中时，能够与中断确认 ( $\overline{IACK}$ ) 信号一起，共同对中断向量进行解码。复位时，AVIS = 0。

**管理权限[administrative privileges]** — 设置软件和硬件访问权限的权限；其中包括安装、管理和维护系统和应用软件，以及网络服务器或个人计算机系统中目录的访问权限。

**聚合类型[aggregate type]** — C 数据类型，例如结构或数组，其中一个变量由多个称为成员的其它变量组成。

**A 定律压缩扩展[A-Law companding]** — 音频信号的量化方案，在此方案中，输入信号在被压缩和处理后，在输出时通过扩展重建。A 定律压缩扩展在欧洲范围内使用。

**别名消歧[alias disambiguation]** — 确定何时两个指针表达式不能指向同一位置的技术，从而使得编译器能够自由地优化此类表达式。

**别名使用[aliasing]** —

1. 定制调试器命令的方法；别名使用提供了一种输入常用命令字符串的简略表达方法。
2. 当指针指向已命名对象时，用多种方式访问单个数据对象的方法。优化器具有检测别名使用的逻辑电路，但别名的使用会使优化器产生问题。
3. 当可以通过多种方式访问单个对象，如两个指针指向单个对象时，就会发生别名使用。它会干扰优化过程，这是因为任何间接引用有可能指向任一其它对象。

**校准[alignment]** — 链接器将输出段置于  $n$  字节边界（其中  $n$  是 2 的幂）内的地址的过程。您可以使用 SECTIONS 链接器指令来指定校准。

**分配[allocation]** — 链接器计算输出段的最终存储器地址的过程。

**分配节点[allocation node]** — 分配给节点间消息的处理器节点。

**ALU 函数[ALU function]** — 对于并行处理器，是指对算术逻辑单元 (ALU) 三个输入进行的操作，其中包括三个输入的任何算术或布尔组合，以及混合算术和布尔函数。

**ALU 函数修饰符** — 对于并行处理器，是指定对由算术逻辑单元 (ALU) 数据路径执行的函数进行修改的 4 位代码（如进位输入或多个算术）。这些函数修饰符在操作码或 D0 寄存器中指定，具体取决于应用。

**ALU 运算** — 对于并行处理器，是由算术逻辑单元 (ALU) 数据路径执行的操作（即，ALU 函数的结果、操作类和任何函数修饰符）。

**美国国家标准研究院 [American National Standards Institute, ANSI]** — 建立由各个行业自愿遵循的标准组织。

**美国国家标准研究院 (ANSI) [ANSI C]** — C 编程语言的一个版本，符合由美国国家标准研究院 (ANSI) 定义的 C 标准。

**信息交换美国标准代码, 1968 [American Standard Code for Information Interchange, ASCII]** — 7 位编码字符标准集（包含奇偶校验在内为 8 位），用于数据处理系统、通信系统和相关设备间的信息交换。ASCII 字符集由控制字符和图形字符组成。

**模拟接口电路 [analog interface circuit, AIC]** — 执行模数 (A/D) 和数模 (D/A) 转换的集成电路。

模拟混合[**analog mixing**] — 两路模拟信号的混合在一起；两路模拟信号复用成一路。

模数转换 [**analog-to-digital, A/D**] — 连续可变的电信号到离散或不连续电信号的转换。

模数转换器 [**analog-to-digital converter, ADC**] — 具有内部采样保持电路，被用来将模拟信号转换为数字信号的转换器。

废止[**annul**] — 任何废止的指令不能完成其管道阶段。

应用编程接口 [**application programming interface, API**] — 用于专有应用程序与通信软件交互或者符合其他供应商产品的协议。

架构[**architecture**] — 全部或部分计算机系统的软件或硬件结构；包括系统的所有详细组成部分。

存档库[**archive library**] — 由归档器将多个单独文件组合成一个单个文件的文件集合。

归档器[**archiver**] — 将多个单独文件集合成为一个存档库的软件程序。借助归档器，您可以添加、删除、提取或替换存档库内的文件。

自变量缓冲器[**argument buffer**] — 放入自变量值的存储器块，伴随发送至服务器并行处理器的命令。

算术逻辑单元 [**arithmetic logic unit, ALU**] — 执行所有算术运算（加、减、乘、除或比较）和逻辑函数的计算机部分。

汇编 — 通过将绝对操作码替换为符号操作码，并将绝对或浮动地址替换为符号地址，由符号语言程序中生成机器语言程序。

汇编程序 — 从包含汇编语言指令和命令的源文件创建机器语言程序的软件程序。汇编程序将绝对操作码替换为符号操作码，并将绝对或浮动地址替换为符号地址。

汇编语言 — 低级符号编程语言，与机器代码语言十分相似，并由多组字母组成 - 每组字母代表一个单个指令；使计算机用户可以使用助记符，而不是数字指令，来编写程序。

汇编语言指令 — 用助记符来表示计算机操作的语言。

汇编模式[**assembly mode**] — 无论当前运行哪种类型的代码，在 **DISASSEMBLY**（反汇编）窗口中显示汇编语言代码但不会显示 **FILE**（文件）窗口的调试模式。

汇编程序优化器[**assembly optimizer**] — 可优化线性汇编代码的软件程序，此线性汇编代码是还未被寄存器分配或调度的汇编代码。当其中一个输入文件具有 **.sa** 扩展名时，使用外壳程序，**cl6x**，自动调用此汇编程序优化器。

汇编时间常数[**assembly-time constant**] — 符号的完整定义，此定义在汇编时指定。

**置为有效[assert]** — 使数字逻辑器件引脚有效的操作。如果引脚是低电平有效，则通过引脚上的低电压将其置为有效。如果引脚是高电平有效，则通过高电压将其置为有效。

**赋值语句[assignment statement]** — 初始化变量值的语句。

**异步发送 [asynchronous transmit, TX] 引脚** — 对来自异步串行端口的数据进行串行传输的引脚；从异步串行端口移位寄存器 (AXSR) 按每次一位的速度接收字符。

**衰减指数 [attenuation index, atten]** — 衰减指数的目的在于使读者了解其他环路滤波器设计图纸中环路滤波器内的 R3 和 C3 组件增加的寄生信号衰减，但是这并非本书的讨论范围。(The attenuation index is intended to give an idea of the spurious attenuation added by the components R3 and C3 in the loop filter of other loop filter design papers, but not this book.) 也为电阻垫以 dB 为单位的衰减提供参考。(Also used in reference to the attenuation of a resistive pad in dB.)

**属性** — 指定应用于随后图形信息的某些特点或特性的参数。

**音频分支线缆[audio breakout cable]** — 将软件开发板 (SDB) 连接到音频输入和输出外设的线缆。它包含立体声线路输出、线路输入和辅助输入的标准 RCA 插孔。

**自动模式** — 上下文关联的调试模式，此模式在 DISASSEMBLY 窗口中显示汇编语言代码与在 FILE 窗口中显示 C 代码之间自动切换，具体取决于当前运行的代码类型。

**自动缓冲接收器使能 [autobuffering receiver enable, BRE] 位** — 启用或禁用自动缓冲接收器的位字段。复位时，BRE = 0。此位存储在缓冲串行端口 (BSP) 控制扩展寄存器 (SPCE) 中。

**自动缓冲接收器停止 [autobuffering receiver halt, HALTR] 位** — 当超出缓冲器的界限时，停止自动缓冲单元 (ABU) 的位字段。复位时，HALTR = 0。此位存储在缓冲串行端口 (BSP) 控制扩展寄存器 (SPCE) 中。

**自动缓冲发送器使能 [autobuffering transmitter enable, BXE] 位** — 启用或禁用自动缓冲发送器的位字段。复位时，BXE = 0。此位存储在缓冲串行端口 (BSP) 控制扩展寄存器 (SPCE) 中。

**自动缓冲发送器停止 [autobuffering transmitter halt, HALTX] 位** — 当超出缓冲器的界限时，停止自动缓冲单元 (ABU) 的位字段。复位时，HALTX = 0。此位存储在缓冲串行端口 (BSP) 控制扩展寄存器 (SPCE) 中。

**自动缓冲单元 [autobuffering unit, ABU]** — 允许缓冲串行端口接口直接读取或写入内部存储器（独立于中央处理单元 (CPU)）的片上模块。自动缓冲功能可以针对发送和接收段分别启用。禁用自动缓冲后，此操作与标准串行端口的操作相类似。

**自动校准[autocalibration]** — 器件的自动调整，这样使输出处于特定输入值的指定范围内。

**autoexec.bat** — 包含初始化 PC 的 DOS 命令的批处理文件。

**自动初始化[autoinitialization]** — 在程序执行开始前，初始化全局 C 变量（包含在 .cinit 段中）的过程。

**加载时的自动初始化** — 链接 C 代码时由链接器使用的初始化方法。链接器在您使用 **-cr** 选项调用链接器时使用此方法。此方法在加载时，而不是运行时，初始化变量。

**运行时的自动初始化** — 链接 C 代码时由链接器使用的初始化方法。链接器在您使用 **-c** 选项调用链接器时使用此方法。链接器将数据表的 .cinit 段加载至存储器，而变量在运行时初始化。

**自动增益控制 [automatic gain control, AGC]** — 根据输入信号振幅改变信号的放大率。

**辅助条目[auxiliary entry]** — 符号在符号表中可能具有的额外条目，其中包含该符号的其他信息（是否为文件名、段名或函数名）。

**辅助寄存器算术单元 [auxiliary register arithmetic unit, ARAU]** — 用于递增、递减或比较辅助寄存器内容的算术单元。其主要功能是为间接寻址处理辅助寄存器值。

**辅助寄存器缓冲器 [auxiliary register buffer, ARB] 位** — 保存辅助寄存器指针 (ARP) 内上一个值的字段。这些位存储在状态寄存器 1 中。

**辅助寄存器文件总线 [auxiliary register file bus, AFB]** — 当前选择的辅助寄存器 (AR) 上对数据存储器位置进行寻址的总线。

**辅助寄存器指针 [auxiliary register pointer, ARP]** — 选择辅助寄存器 (AR) 用于间接寻址的字段。加载 ARP 后，前一个 ARP 值被复制到辅助寄存器缓冲器 (ARB) 中。ARP 可在使用间接寻址时由存储器访问指令 (memory-reference instruction) 修改，也可由 MAR 和 LST 指令修改。这些位存储在状态寄存器 0 中。

**辅助寄存器指针缓冲器 [auxiliary register pointer buffer, ARB]** — 状态寄存器中保存辅助寄存器指针 (ARP) 前一个值的字段。

**2 B**

**B0 片上块[B0 on-chip block]**— 双存取 RAM 的片上块，可配置为数据存储器或程序存储器，具体取决于状态寄存器中配置控制 (CNF) 位的值。

**B1 片上块[B1 on-chip block]**— 可用于数据存储器的双存取 RAM 的片上块。

**B2 片上块[B2 on-chip block]**— 可用于数据存储器的双存取 RAM 的片上块。

**后沿[back porch]** — 同步结束与相应消隐脉冲之间的视频波形间隔。水平后沿指定为整数个帧时钟 (FCLK) 周期；垂直后沿指定为整数线（对于隔行扫描模式为半行）。另请参见 *前沿*。

**桶式转子[barrel rotator]** — 在数据字内旋转位的位置的器件。与桶式移位器相似，不同点在于移出的位将围绕在空出的位周围。

**桶式移位器[barrel shifter]** — 在字中旋转位的单元。另请参见 *后分频器* 和 *预分频器*。

**基值[base]** —

1. 基准值。
2. 由指数表示乘以自身多少倍的数字。

**基本集 ALU** — 并行处理器的算术逻辑单元 (ALU) 运算基本集，其中包括布尔以及混合算术和布尔函数。

**基本集算术[base set arithmetics]** — 并行处理器基本指令集，可在操作码中指定算术逻辑单元 (ALU) 算术运算。此指令集包括 11 项与类无关的算术运算和 6 项特定类算术运算。

**基本集布尔[base set Booleans]** — 并行处理器基本指令集，可在操作码中指定算术逻辑单元 (ALU) 布尔运算。此指令集包括 256 中可能的布尔函数。

**ISDN 的基本速率服务 [basic rate service of ISDN , BRI]** — 综合业务数字网 (ISDN) 的基本速率服务，提供两个 B 通道和一个 16Kbps D 通道。

**批处理文件[batch file]** — 两个不同文件类型中的一个。一类包含供 PC 执行的 DOS 命令。包含调试程序执行所需调试程序命令的第二类批处理文件。PC 不会执行调试程序批处理文件，而调试程序不执行 PC 批处理文件。

**电池管理单元 [battery management unit , BMU]** — 具有主控电量计器件的器件或芯片组，此电量计器件控制受控电池充电器。

**基准[benchmarking]** — 可跟踪代码指定段占用的 CPU 周期数量的程序执行类型。

**BIG 位** — 指定输入/输出 (I/O) 端口等待状态寄存器映射方式的字段。此位存储在等待状态控制寄存器 (CWSR) 中。复位时，BIG = 0。

**大尾序[big-endian]** — 一种寻址协议，字中的字节从左至右编号。字中较高有效字节具有较低编号的地址。端字节排序视硬件而定，并在复位时确定。另请参见 *小尾序*。

**绑定[binding]** — 关联或链接两个互补的软件对象。

**BIO 引脚** — 可通过条件指令进行测试的通用输入引脚；当外部器件驱动 BIO 低时，此类条件指令会产生分支。

**位对齐块传输 [bit-aligned block transfer, bitBLT]** — 将像素块从位图中的一个位置传输至另一个位置。

**位检测[bit detection]** — 支持最左边一个、最右边一个、最左边位变化及最右边位变化检测的特定逻辑。

**位图** —

1. 图像的数字表示，其中位映射到像素。
2. 存储器块，用于以器件专用格式保持光栅图像。

**位平面[bit plane]** — 位存储阵列（平面），用于存储图像每个像素的特定位置。每个像素的 0 位存储在位平面 0 中，每个像素的第一个位存储在位平面 1 中，以此类推。

**位反向寻址[bit-reversed addressing]** — 地址位反向的寻址，以加快算法处理的速度，如傅立叶变换算法。

**位反向索引寻址[bit-reversed indexed addressing]** — 间接寻址方法，通过在基数为 2 的快速傅立叶变换 (FFT) 程序中重新排序数据点来实现高效 I/O 操作。辅助寄存器算术单元 (ARAU) 中进位传播的方向是反向的。

**每像素位数 [bits per pixel, BPP]** — 用于在数字化图像中表示每个像素的色彩值的位数。

**消隐[blanking]** — 水平和垂直回扫期间扫描光束消失的过程。另请参见 *激活时间*；*消隐区域*。

**消隐区域[blanking area]** — 非活动而是消隐的显示区域。消隐区域中无像素显示。消隐期间会出现垂直和水平回扫。另请参见 *消隐脉冲*。

**消隐脉冲[blanking pulse]** — 回扫期间产生的正或负脉冲，出现在每个字段的结尾；用于在垂直或水平回扫期间消隐扫描线。另请参见 *消隐*。

**已阻止[blocked]** — 任务尚不能执行时的状态。任务被阻止的情况分为挂起或自愿选择等待某事件，如消息或信号的到达。

**块未命中[block miss]** — 高速缓存未命中，其中寻址块未驻留在高速缓存中。最近最少使用 (LRU) 算法决定丢弃哪个现有高速缓存块。如果高速缓存包含任何已修改的数据（仅限主处理器 (MP) 数据高速缓存），那么任何已修改的子块将在请求的子块被送入高速缓存之前写回外部存储器。

**块重复激活标志 [block repeat active flag, BRAF] 位** — 指示当前块重复激活的字段。此位通常在执行重复块 (RPTB) 指令时设置，并在块重复计数器寄存器 (BRCR) 递减至 0 以下时清除。向此位写入 0 将使块重复无效。复位时，BRAFF = 0。

**块写入[block write]** — 使传输控制器 (TC) 能够执行多列写操作的非标准数据包传输。

**bloomer** — 一个非常高的寄生信号，达到 -30dBc，或者更高，并且是有害寄生信号集合的一部分。如果寄生信号在频带内，要被视为 *bloomer*，寄生信号需要为 -10dBc 或者更高。

**引导[boot]** — 将程序加载到程序存储器中的过程。

**引导加载程序[bootloader]** — 通过标准存储器器件（包括 EPROM），有/无握手，或在加电时通过到 RAM 的串行端口加载和执行从主机处理器接收的程序的片上程序。

**BOOT 引脚** — 启用片上引导加载程序的引脚。当 BOOT 保持低电平时，处理器在硬件复位后执行引导加载程序。当 BOOT 保持高电平时，处理器跳过引导加载程序的执行，并在复位时访问片外程序存储器。

**边界扫描[boundary scan]** — 在芯片的边界或逻辑段使用扫描寄存器捕获引脚状态。通过扫描这些寄存器，所有引脚状态可通过 JTAG 端口进行传输，用于分析。

**分支[branch]** — 将程序控制转至非连续程序存储器地址的切换。

**间隔中断 [break interrupt, BI] 位** — I/O 状态寄存器 (IOSR) 内表示何时在异步接收 (RX) 引脚上检测到间隔的位。

**断点[breakpoint]** — 例程中由指令、指令位数或其它条件指定的位置，在该位置上，例程可能因外部干预或监控例程而中断。

**桥接负载 [bridge-tied load, BTL]** — 针对音频放大器的输出配置。

**.bss 段** — 缺省常用对象文件格式 (COFF) 段中的一个。使用 **.bss** 指令可在存储器映射中保留指定量的空间，用于以后存储数据。**.bss** 段未初始化。

**气泡[bubble]** — 传输控制器 (TC) 不执行任何操作的管道周期。气泡出现是因为争用、下一次存取的数据量不足，或者只是无活动请求。

**缓冲串行端口 [buffered serial port , BSP]** — 一个片上模块，此模块包含一个全双工的双缓冲串行端口接口和一个自动缓冲单元 (ABU)。BSP 的双缓冲串行端口是标准串行端口接口的增强版。双缓冲串行端口允许传输连续的通信流 (8, 10, 12 或 16 位数据包)。BSP 的状态和控制由 BSP 控制扩展寄存器 (SPCE) 中指定。

**缓冲池[buffer pool]** — 属于特定任务的一组消息缓冲器，通常大小一致。

**公告牌系统 [bulletin board system , BBS]** — 远程用户可通过调制解调器 (modem) 访问的计算机程序，此程序使他们能发布问题，查看回应，以及下载文件。

**突发模式[burst mode]** — 一种同步串行端口模式，在帧同步脉冲 (FSX 和 FSR) 后传输单个字。

**总线请求 [BR] 引脚** — BR 引脚与 BR 信号关联，后者在全局数据存储器存取启动后被置为有效。

**总线监视[bus watching]** — 处理器的功能，可针对地址总线用量的变化进行计算和调整。

**蝶形快速傅里叶变换 (butterfly)** — 计算  $n$  点快速傅立叶变换 (FFT) 的内核函数，其中  $n$  是 2 的幂。输入的组合样式类似于蝴蝶翅膀。

**字节** — 传统上，一个字节是作为一个单元进行操作操作的 8 个相邻位的序列，。

**字节排序位 [byte-ordering bit, BOB]** — 使用主机端口接口时，影响主机处理器数据和地址传输的字段。只有主机处理器可以切换此位。在首次数据或地址寄存器存取前，BOB 必须被初始化。此位存储在主机端口接口控制 (HPIC) 寄存器中。

**3 C**

**C 编程语言** — 一种高级通用编程语言，用于编写编译程序和操作系统，以及编程微处理器。

**高速缓存[cache]** — 为了实速快速存取，复制较慢存储器内常用数据或指令的快速存储器。快速存取通过高速缓存的高速及其片上与 CPU 的邻近距离实现。

**高速缓存块[cache block]** — 高速缓存存储器的一部分。每个块有一个相关联的标志寄存器，并分为四个子块。高速缓存存储器分配在块大小部分，但高速缓存处理在子块级别执行，根据需要使用子块。

**高速缓存清理[cache clean]** — 一项主处理器 (MP) 操作，通过将修改的（脏）数据高速缓存子块写回存储器来更新外部存储器，从而将子块的脏位复位为 0。

**高速缓存一致性[cache coherency]** — 一个或多个高速缓存存储器一致且准确地表示外部存储器中相应内容时的状态或状况。

**高速缓存刷新[cache flush]** — 一项主处理器 (MP) 操作，通过将修改的（脏）数据高速缓存子块写回存储器来更新外部存储器，从而将子块的存在位和脏位重置为 0。

**高速缓存命中[cache hit]** — 高速缓存存储器包含请求的指令或数据字时的状态或状况。

**高速缓存未命中[cache miss]** — 高速缓存不包含请求的指令或数据字时的状态或状况。

**高速缓存子块[cache sub-block]** — 高速缓存块的四个分区之一。高速缓存子块是在子块未命中时送入高速缓存的存储器单元。每个子块在该块的标志寄存器中有一个存在位（以及一个脏位，仅用于主处理器 (MP) 数据高速缓存）。

**校准 A 检测 [CAD] 位** — 异步串行端口控制寄存器 (ASPCR) 中的一个位，可启用/禁用片上异步串行端口的自动波特率检测逻辑。

**CALLS 窗口** — 列出程序所调用的函数的窗口。（此窗口是所有 TI 调试程序的图形用户界面的组成部分。）

**捕获模式[capture mode]** —

1. 串行寄存器传输 (SRT) 模式，在该模式下，图像在采集后存储到存储器中。与采集的图像不对应的存储器位置可能会被覆盖。另请参见 **显示模式**；**合并模式**。
2. 一种音频子系统模式，在该模式下，直接存储器存取 (DMA) 传输已由音频编解码器采集的已读取音频数据。

**进位位 [C 位]** — 状态寄存器中，由算术逻辑单元 (ALU) 用于扩展算术运算及累加器移位和旋转的位。进位位可通过条件指令进行测试。

**转换[casting]** — C 表达式的一种功能，使您可以将一类数据当做另一类数据使用。

**C 编译程序** — 将 C 源语句转换成汇编语言源语句的软件程序。

**CIO0-CIO3 位** — 异步串行端口控制寄存器 (ASPCR) 中，将引脚 IO0-IO3 单独配置为输入或输出的位。例如，CIO0 配置 IO0 引脚。另请参见 **DIO0-DIO3 位**；**IO0-IO3 位**。

**中央算术逻辑单元 [central arithmetic logic unit, CALU]** — CPU 执行算术和逻辑运算的主要算术逻辑单元。它用一个值进行运算，其输出保存在累加器中。

**中央处理单元 [central processing unit, CPU]** — CPU 是参与算术、移位和布尔逻辑运算，以及数据和程序存储器地址的生成的处理器部分。CPU 包括中央算术逻辑单元 (CALU)，乘法器和辅助寄存器算术单元 (ARAU)。

**信道与信道间隔[channel-and-channel spacing]** — 在很多应用中，将被生成的一组频率之间的间隔是平均的。这些将被生成的频率通常被称为信道，而这些信道之间的间隔通常被称为信道间隔。

**电荷泵[charge pump]** — 与相位频率检测器联合使用，这款器件输出恒定振幅的电流，但是极性和占空比可变。它通常被制成一个输出稳定电流值的器件，此电流值等于输出电流的时间平均值 (It is usually modeled as a device that outputs a steady current of value equal to the time-averaged value of the output current)。

**电荷泵增益 [KΦ]** — 这是以 mA/(弧度  $2\pi$ ) 为单位表示的增益。

**子窗口[children]** — 针对聚合类型打开的其他窗口，是在现有 DISP 窗口中显示的父聚合类型的成员。(这些窗口是所有 TI 调试程序的标准图形用户界面的组成部分。)另请参见 *DISP* 窗口。

**芯片级封装 [chip scale package , CSP]** — 没有任何传统封装的器件；具有焊球的芯片本身。

**色度[chrominance]** — 全国电视标准委员会 (NTSC) 或逐行倒相制式 (PAL) 视频信号包含两部分，它们组成您在屏幕上所看到的内容，它们分别是：黑色和白色部分（亮度）和彩色部分（色度）。另请参见亮度。

**循环寻址[circular addressing]** — 一种寻址模式，在该模式下，辅助寄存器用于在地址范围内循环，以在存储器中创建循环缓冲器。也称为取模寻址。

**循环缓冲器 1 使能 [CENB1] 位** — 启用或禁用循环缓冲器 1 的字段。复位时，CENB1 = 0。此位存储在循环缓冲器控制寄存器 (CBCR) 中。

**循环缓冲器 2 使能 [CENB2] 位** — 启用或禁用循环缓冲器 2 的字段。复位时，CENB2 = 0。此位存储在循环缓冲器控制寄存器 (CBCR) 中。

**与类无关的算术运算[class-independent arithmetics]** — 算术逻辑单元 (ALU) 运算基本集中，并行处理器的 11 项 ALU 算术函数，可用于所有 8 个运算类。

**特定类的算术运算[class-specific arithmetics]** — 算术逻辑单元 (ALU) 运算基本集中，并行处理器算术函数，仅适用于 8 个运算类的子集。

**客户端[client]** — 从服务器程序或任务中请求服务的程序或任务。

**CLKMOD 引脚** — 决定片上时钟发生器是以二分频模式运行，还是运行在二倍频模式下。另请参见 *时钟模式*。

**时钟周期[clock cycle]** — 基于外部时钟输入的周期。

**时钟模式 (时钟发生器) [clock mode (clock generator)]** — 将内部 CPU 时钟频率设置为输入时钟信号 CLKIN 频率的几分之一或倍数频率的模式之一。

**时钟模式 [MCM] 位** —

1. 指定发送时钟输入或输出 (CLKX) 引脚的时钟源的字段。复位时，MCM = 0。此位存储在串行端口控制寄存器 (SPC) 和时分复用 (TDM) 串行端口控制寄存器 (TSPC) 中。
2. 同步串行端口控制寄存器 (SSPCR) 中的位，决定计时同步串行端口传输的信号源是外部的还是内部的。

**时钟模式** — 时钟发生器使用的选项，以将内部 CPU 时钟频率设置为输入时钟信号频率的几分之一或倍数频率。

**时钟极性 [clock polarity, CLKP] 位** — 表示数据何时由接收器采样并由发送器发送的字段。复位时，CLKP = 0。此位存储在缓冲串行端口控制扩展寄存器 (SPCE) 中。

**代码** — 写入来执行任务的一组指令；计算机程序或程序的一部分。

**代码显示窗口[`code-display windows`]** — 显示代码、文本文件、或特定代码信息的窗口。此类别包括 **DISASSEMBLY**、**FILES** 和 **CALLS** 窗口。（这些窗口是所有 TI 调试程序的标准图形用户界面的组成部分。）

**代码发生器[`code generator`]** — 一种编译器工具，获取解析器和优化器生成的文件，然后生成汇编语言源文件。

**编码器-解码器 [`coder-decoder , codec`]** — 代码以一个方向传输，解码以另一个方向传输的器件。

**COFF 幻数[`COFF magic number`]** — 常用对象文件格式 (COFF) 文件标头条目，标识常用对象文件格式的版本或可执行模块的处理器。

**冷启动[`cold boot`]** — 加电时将程序加载到程序存储器中的过程。

**列地址选通 [`CAS`]** — 驱动 DRAM 和 VRAM 列地址选通输入的存储器接口信号。

**列掩码 [`column mask, COLMASK`]** — 串行寄存器传输 (SRT) 控制器用于地址计算的控制掩码。

**命令[`command`]** — 提供给系统（比如汇编程序），表示系统操作请求的字符串。

**命令描述块 [`command descriptor block , CDB`]** — 用于将请求从启动器传递到目标的命令块。

**命令文件** — 用户创建的、为链接器或调试器命名初始化选项和输入文件的文件。

**命令解释程序** — 一个软件例程，从客户端程序接收命令，然后将每个命令派发至适当的子例程，予以执行。

**命令行** — **COMMAND** 窗口中您可以输入命令的部分。

**命令行光标** — 屏幕上的标记，标识命令行中的当前字符位置。

**命令行界面** — 与计算机的操作系统进行通信的方法，在显示器上的特定位置键入一组已定义的命令。基于命令的系统通常是可编程的。

**COMMAND 窗口** — 提供区域让您输入命令，并让调试程序回显命令条目，显示命令输出以及列出进度或错误消息的窗口。

**注释[`comment`]** — 记录或改进源文件可读性的源语句（或源语句的一部分）。注释不编译、汇编或链接；不会影响对象文件。

**常用对象文件格式 [`common object file format, COFF`]** — 一种二进制对象文件格式，通过支持段的概念来改进模块化编程。

**比较频率 [`FCOMP`]** — 由 R 计数器值分频的晶振基准频率。这也被称为基准频率。

**编译器[`compiler`]** — 将一组高级语言指令转换为目标机器汇编语言的翻译程序。

**互补金属氧化物半导体 [`complementary metal oxide semiconductor, CMOS`]** — 数字逻辑电路的一种形式，具有低功耗、宽电源范围和高抗扰度等特点。

**复合区 [`composite area , CAREA`]** — 由帧定时器产生的信号，可用于定义特殊区域，如过扫描边界。此信号在隔行扫描和逐行扫描模式下的作用相同，均定义一个正矩形区域。

- 复合消隐 [CBLNK]** — 一种信号，可将有关水平和垂直定时间隔的信息组合成一个信号，该信号比水平消隐 (HBLNK) 或垂直消隐 (VBLNK) 更复杂。CBLNK 可用于在水平和垂直回扫期间禁用像素采集或显示。另请参见 *消隐脉冲*。
- 复合同步 [CSYNC]** — 一种信号，可将有关水平和垂直定时间隔的信息组合成一个信号，该信号比水平同步 (HSYNC) 或垂直同步 (VSYNC) 更复杂。CSYNC 可用于启用支持复合视频的显示屏的电子束回扫。
- 复合视频 [composite video , CVBS]** — 一种信号，传递有关颜色、亮度的视频图像信息以及水平和垂直扫描的同步信号。
- 复合视频显示 [composite video display]** — 接收一个信号中所有编码视频信息的显示。此信息可以包括颜色、水平同步、垂直同步，或者全国电视标准委员会 (NTSC) 对于电视和视频磁带录制通常所要求的其他信息。
- 压缩和扩展 [compressed and expanded]** — 音频信号的量化方案，输入信号经压缩和处理后，在输出时通过扩展重建。共有两种截然不同的压扩方案 — 在欧洲使用的 A 定律，以及用于美国的  $\mu$  定律。
- 压缩和解压缩 [compression and decompression, codec]** — 代码以一个方向传输，解码以另一个方向传输的器件。
- 条件处理 [conditional processing]** — 根据对指定表达式的评估，处理一个源代码块或一个源代码替代块的方法。
- 条件源 [conditional source]** — 一种源操作数；D 寄存器对中奇数或偶数寄存器，具体取决于负状态位。
- config.sys** — 包含初始化 PC 命令的批处理文件。
- 配置控制 [configuration control, CNF] 位** — 状态寄存器中用于决定片上 RAM 块映射到程序空间或数据空间的位。
- 配置的存储器 [configured memory]** — 链接器命令文件中 MEMORY 指令指定的，并由链接器用于分配程序代码和数据的存取器。
- 连接** — 启动器使用的一种功能，用于选择目标来启动操作。
- 连接器** — 将线缆连接到外围器件，以进行电子通信的硬件耦合器。软件开发板 (SDB) 具有四个将线缆从输入或输出外设连接到该板的连接器。
- 常量 [constant]** — 一个可用作操作数的固定或不变的值或数据项。
- 争用 [contention]** — 两个或更多的器件或单元尝试同时访问相同资源的情况。
- 背景保存和恢复 [context save and restore]** — 当器件进入/或退出子例程（如中断处理例程）时，系统状态（状态寄存器、累加器、乘积寄存器、临时寄存器、硬件堆栈、辅助寄存器）的保存/或恢复。
- 连续模式 [continuous mode]** — 一种同步串行端口模式，在该模式下，只需一个帧同步脉冲来以最大频率发送或接收几个连续的数据包。另请参见 *突发模式*。
- 连续传导模式 [Continuous Conduction Mode , CCM]** — 就直流到直流转换器内的电感器电流而言，连续传导模式 (CCM) 的特点是在部分开关周期内，电感器电流绝不会下降到零值以下。将平均电感器电流减少至峰值到峰值电感器纹波电流的 1/2 以下会使转换器脱离 CCM 模式，并进入断续传导模式 [DCM]。由于电感器电流与所有直流到直流转换器的负载电流相关，此工作模式将随负载电流的变化而变化。电感值和开关频率是决定转换器在这些模式间转换何种负载电流的主要因素。

**连续时间近似法** — 在这个方法中，电荷泵的离散电流脉冲被模拟为一个连续电流，此电流的幅度等于电流脉冲的时间平均值。

**控制电压 [V<sub>TUNE</sub>]** — 控制压控振荡器 (VCO) 频率输出的电压。

**卷积 (convolution)** — 数字滤波的时域基准，广泛利用乘积总数。

**C 优化程序** — 可提高执行速度并减小 C 程序大小的软件工具。

**计数信标[counting semaphore]** — 多处理环境中，限制共享资源（例如，存储）访问的经典方法。信标是受保护的变量（或抽象数据类型），只有特定的测试或递增变量值的操作才能访问它。

**CPU 周期** — 特定执行数据包处于特定管道阶段的时间长度。CPU 周期边界总是出现在时钟周期边界上；但是，存储器停滞可导致 CPU 周期延长多个时钟周期。

**CPU 窗口** — 显示片上寄存器内容（包括程序计数器、状态寄存器及其它寄存器）的窗口。（此窗口是所有 TI 调试程序的标准图形用户界面的组成部分。）

**交叉开关[crossbar]** — 用于多处理器系统的一般可配置高速总线交换网络，允许多个处理器中的任何一个连接至多个存储器模块中的任何一个。

**交叉参考列表器[cross-reference lister]** — 将链接对象文件接受为输入并产生交叉参考列表作为输出的调试工具。

**交叉参考列表[cross-reference listing]** — 由汇编程序创建的输出文件，其中列出其定义的符号、这些符号的位置、哪些行引用了它们以及它们的最终值。

**晶振频率 [X<sub>TAL</sub>]** — 用于基准的稳定且精确的频率。

**当前数据页[current data page]** — 数据存储器页指针 (DP) 内容所表示的数据页。另请参见 *数据页*。

**当前字段光标[current-field cursor]** — 在活动窗口中标识当前字段的屏幕图标。

**周期[cycle]** — 特定执行数据包处于特定管道阶段的时间长度。CPU 周期边界总是出现在时钟周期边界上；但是，存储器停滞可导致 CPU 周期延长多个时钟周期。

## 4 D

**D0-Dn 数据总线引脚** — 在处理器与外部数据或程序存储器或 I/O 器件之间传输数据的外部数据引脚。

**阻尼因子[damping factor]** — 对于一个二阶瞬态响应，阻尼因子决定指数包络的波形，此指数包络会增加频率振铃。

**.data 段** — 缺省通用目标文件格式 (COFF) 段中的一个。 .data 段是包含初始化数据的初始化段。 您可以使用 .data 指令将代码汇编到 .data 段中。

**数据地址生成逻辑电路[data-address generation logic]** — 为数据存储器的读取和写入生成地址的逻辑电路。 这个包括辅助寄存器和辅助寄存器算术单元 (ARAU) 的电路可在每个机器周期内生成一个地址。

**数据总线[data bus]** — 一组用于数据路由的连接。

**数据高速缓存[data cache]** — 保存主处理器 (MP) 所需高速缓存数据的 MP 的两个 SRAM 内存组。 并行处理器的数据 RAM 不会被高速缓存。 另请参见 *SRAM 内存组*。

**数据高速缓存复位 [data-cache reset, DCR]** — 与主处理器 cmdnd 指令一起发送的命令，用于复位所有数据高速缓存标志寄存器和最近最少使用数据 (ILRU) 寄存器。

**数据显示窗口[data-display windows]** — 用于观察和修改各类数据的窗口。 此类别包括 MEMORY, CPU, DISP 和 WATCH 窗口。 (这些窗口是所有 TI 调试程序的标准图形用户界面的组成部分。)

**数据存储器[data memory]** — 用于存储和操作数据的存储器区域。

**数据存储器地址 [DMA] 位** — 直接寻址指令的 7 个最低有效位 (LSB)，此寻址指令在 128 字数据页内包含直接相关地址。 7 个 LSB 与数据存储器页指针 (DP) 连结起来，以组成 16 位的直接存储器地址。 另请参见 *数据存储器页指针 (DP) 位*。

**数据存储器页指针 [DP] 位** — 一个指定当前数据存储器页地址的位字段。 DP 位与指令字的最低有效位 (LSB) 连结起来，以组成直接存储器地址。 这些位存储在状态寄存器 0 (ST0) 中。

**数据存储器页 0[data memory page 0]** — 数据存储器空间中的首页，其中驻留有存储器映射寄存器和暂存式 RAM 块 (B2)。

**数据存储器选择位 [ $\overline{DS}$ ]** — DSP 将  $\overline{DS}$  置为有效，表示到外部数据存储器的访问 (本地或全局)。

**数据页[data page]** — 数据存储器内的邻近块。 定点器件包含数据存储器中的 128 个字的块。 数据存储器包含 512 个数据页。 数据页 0 是数据存储器的首页 (地址为 0000h–007Fh)；数据页 511 是尾页 (地址为 FF80h–FFFFh)。 在浮点器件中，数据页的长度为 64K 个字。

**数据 RAM** — 可供主处理器或并行处理器进行通用数据存储的片上 RAM。

**数据读取地址总线 [data-read address bus, DRAB]** — 承载数据存储器每次读取地址的内部总线。

**数据读取总线 [data read bus, DRDB]** — 将数据从数据存储器传输至中央算术逻辑单元 (CALU) 和辅助寄存器算术单元 (ARAU) 的内部总线。

**数据宽度[data size]** — 用于表示特定数字的位数 (8、16、24、32 或 40)。

**数据空间等待状态 [DSWS] 位** — 等待状态发生器控制寄存器 (WSGR) 中的值，用于决定应用到片外数据空间读取和写入的等待状态数量。

- 数据终端就绪 [data terminal ready , DTR]** — 由 IEEE RS-232 串行标准定义的信号，允许诸如计算机或终端等数据源表示其已为传输做好准备。
- 数据单元[data unit]** — 并行处理器的数据操作硬件单元，其中包括算术逻辑单元 (ALU)，乘法器，mf 扩展器和桶式转子。
- 数据单元操作[data unit operation]** — 对于并行处理器，由数据单元内的硬件执行的操作。数据单元允许在单个周期中执行乘法和算术逻辑单元 (ALU) 数据路径运算。
- 数据写入总线 [data write bus , DWEB]** — 将数据传送至程序存储器和数据存储器的内部总线。
- 死区[dead zone]** — 这是由组件延迟所导致的相位频率检测器的一个属性。由于组成 PFD 的组件具有一个非零延迟时间，所以相位检测器将忽略极小相位误差。
- 死区消除电路[dead zone elimination circuitry]** — 这个电路可被添加至相位检测器，以避免其在死区中运行。这个电路的工作方式是始终让电荷泵的接通时间尽可能的短。(This usually works by causing the charge pump to always come on for some minimum amount of time.)
- 取消置位[deassert]** — 使数字逻辑器件引脚处于未激活状态。如果引脚是低电平有效，则通过引脚上的高电压将其置为无效。如果引脚是高电平有效，则通过低电压将其置为无效。
- 调试程序[debugger]** — 用于识别并清除程序中错误的软件接口。
- 解码阶段[decode phase]** — 指令解码所处的管道阶段。另请参见 管道。
- 缺省任务[default task]** — 无其它任务就绪时运行的任务。
- 延迟槽[delay slot]** — 指令的第一个执行相位 (E1) 后出现的 CPU 周期，此指令的结果不可用。
- Δ 引导传输[delta-guided transfer]** — 引导数据包传输的一个类型，其中引导表包含将被添加至之前二维插入码的起始地址的 Δ 值，以形成新插入码的起始地址。另请参见 引导传输。
- Δ 中断[delta interrupt]** — 如果通用 I/O 引脚 (IO0, IO1, IO2 或 IO3) 中的任何一个发生变化，产生的异步串行端口中断 (TXRXINT)。
- Δ 中断屏蔽 [delta-interrupt mask,DIM] 位** — 异步串行端口控制寄存器 (asynchronous serial port control register, ASPCR) 内的一个位，用于启用和禁用 Δ 检测。Δ 检测功能允许或防止由于 I/O 引脚发生变化而产生的中断。
- Δ-Σ 锁相环 [delta-sigma PLL]** — 通过在两个或更多值之间交替 N 计数器值，可实现小数 N 分频值的分频 PLL。通常情况下为两个值。
- 非常规数[denormal]** — 具有零指数和非零尾数的浮点数。
- 目标控制器[destination controller]** — 传输控制器 (TC) 内处理数据包传输的两个独立控制器中的一个。目标控制器生成将数据包数据写入目标存储器区域所需的地址。
- 目标端口[destination port]** — 消息发送的目标消息端口。另请参见 端口 ID。
- 检测完成 [ADC] 位** — I/O 状态寄存器 (IOSR) 中的一个位；用于在异步串行端口中实施自动波特率检测的标志位。
- Deutsch Industrie Norm [DIN]** — 控制一系列音频信号传输设备的插头、线缆和插座的技术规格的德国国家标准组织。

- 器件驱动程序[device driver]** — 实现计算机硬件与器件之间通信的软件。器件驱动程序也可以转换数据并调用其它驱动程序来将数据实际发送到器件。软件开发板 (SDB) 使用针对 Windows NT 的器件驱动程序，以确保主机和 SDB 之间的通信。
- dhrystones** — 用于基准测试处理器性能的算法。
- 芯片尺寸球状引脚栅格阵列 [die-sized ball grid array, DSBGA]** — 针对芯片尺寸封装 (CSP) 和晶圆级芯片尺寸封装 (WCSP) 的 JEDEC 命规则。
- 数字回路 [digital loopback, DLB] 模式** — 一种同步串行端口测试模式，在该模式下，接收引脚从内部连接到同一器件上的发送引脚。这个模式由 DLB 位启用或禁用，使您能够测试端口是否运行正常。
- 数字回路 [DLB] 位** — 将串行端口置于数字回路模式的字段。复位时，DLB = 0。这个位存储在串行端口控制 (SPC) 寄存器和时分复用 (TDM) 串行端口控制 (TSPC) 寄存器中。
- 数字混合[digital mixing]** — 两个数字信号混合成一种；两个数字信号的代数和。
- 数字信号处理器 [digital signal processor, DSP]** — 将诸如声音或光照等的模拟信号转换为离散或断续电脉冲数字信号的半导体元器件，这样的话，可对这些信号进行操作。
- 数模转换 [digital-to-analog, D/A]** — 离散或断续电气信号到连续可变信号的转换。。另请参见 **数模转换器**。
- 数模转换器 [digital-to-analog converter, DAC]** — 将由一串数字表示的信号（数字）转换为不断变化的信号（模拟）的器件。另请参见 **数模转换**。
- 数字可视接口 [digital visual interface , DVI]** — 数字可视接口被用来连接视频源与显示设备，例如计算机显示器。
- 数字转换器[digitizer]** — 将模拟视频信号转换为数字信号进行解码的视频采集前端部件。
- 分维传输[dimensioned transfer]** — 包含源或目的传输，它可以是数据字节的一个简单连续线性序列，或者也可以包含多个此类区域。另请参见 **引导传输**。
- DIN 连接器** — 一种符合德国国家标准组织 [Deutsch Industrie Norm - (DIN)] 标准的连接器，有时用于计算机和音频连接中。最常见的是 PC 键盘连接器。另请参见 **迷你 DIN 连接器**。
- DIO0-DIO3 位** — I/O 状态寄存器 (IOSR) 中的位。这些位用于异步串行端口启用时（即，当异步串行端口控制寄存器 (ASPCR) 的复位异步串行端口 (URST) 位为 1 时），跟踪之前已知或未知的信号值的变化。例如，DIO0 表示 IO0 引脚上发生的变化。另请参见 **CIO0-CIO3 位；IO0-IO3 位**。
- 直接地址总线 [direct address bus, DAB]** — 为中央处理单元 (CPU) 提供数据地址的总线。
- 直接寻址[direct addressing]** — 指令在数据存储单元内寻址的方法之一。在直接寻址中，数据页指针 (DP) 保存地址（当前数据页）的 9 个最高有效位 (MSB)，而指令字提供地址（偏移量）的 7 个最低有效位 (LSB)。另请参见 **间接寻址**。
- 直接调用[direct call]** — 一个函数使用函数名称调用另一函数的函数调用。
- 直接外部存取 [direct external access, DEA]** — 访问片外（外部）存储器的方法，无需向传输控制器 (TC) 发送数据包传输请求。
- 指令** — 用于控制软件工具的操作和功能的特殊用途命令。

**直接存储器存取 [direct memory access , DMA]** — 主机处理器以外的器件争用、得到存储器总线控制权的一种机制，这样，数据传输可以在独立于主机的情况下发生。

**直接存储器存取模式 [DMA 模式]** — 一种音频子系统模式，在该模式下，DMA 传输读取已由音频编解码器采集的音频数据，或者 DMA 传输为回放提供音频数据。不能同时进行 DMA 采集和回放。

**脏标志 [dirty flag]** — 与主处理器数据高速缓存存储器的每个子块关联的存储位，用于表示子块是否包含需要写回主存储器的已被修改的数据。

**反汇编 [disassembly]** — 将存储器的内容从机器语言转换为汇编语言的过程（也称为反向汇编）。

**DISASSEMBLY 窗口** — 显示存储器内容反汇编的窗口。

**断开 [disconnect]** — 导致目标释放 SCSI 总线控制的函数（SCSI 总线被放置在 BUS FREE（总线空闲）状态上）。

**不连续** — 由于指令将诸如分支、调用、返回等新值载入 PC，调试程序获取的地址变得不连续的一种状态。

**断续传导模式 [Discontinuous Conduction Mode, DCM]** — 就直流到直流转换器内的电感器电流而言，断续传导模式 (DCM) 的特点是在部分开关周期内，电感器电流下降为零，并且保持为零值。总的来说，将平均电感器电流增加至峰值到峰值电感器纹波电流的 1/2 以上会使转换器脱离 DCM 模式，并进入持续传导模式 [CCM]。由于电感器电流与所有直流到直流转换器的负载电流相关，工作模式将随负载电流的变化而变化。电感值和开关频率是决定转换器在这些模式间转换何种负载电流的主要因素。

**离散余弦变换 [discrete cosine transform, DCT]** — 用于操作压缩静态和移动图片数据的快速傅立叶变换。另请参见 *快速傅立叶变换; JPEG 标准*。

**显示区 [display area]** — COMMAND 窗口或并行调试管理器 (PDM) 窗口的一部分，其中，调试程序或 PDM 回显命令条目，显示命令输出，并且列出进度或错误消息。

**显示模式** — 串行寄存器传输 (SRT) 模式，在该模式下，信息从帧存储器传输至显示器件。另请参见 *捕获模式; 合并模式*。

**DISP 窗口** — 显示聚合数据类型成员的窗口。（此窗口是所有 TI 调试程序标准图形用户界面的组成部分。）

**分频值 [divide-down value]** — 定时器分频寄存器 (TDDR) 中的值。此值是片上定时器时器的预分频计数。分频值越大，定时器中断速率越慢。

**DMA 协处理器 [DMA coprocessor]** — 独立于处理器之外，传输存储器位置内容（除初始化外）的外设。

**D\_OPTIONS** — 可用于确认常用调试程序选项的环境变量。

**点时钟 [dot clock]** — 按照视频数据输出至显示器的频率运转的时钟。

**双缓冲 [double buffering]** — 使用双缓冲器来实现两个处理器之间或一个处理器和一个外设之间高效单向数据传输的方法。每个缓冲器均是一个存储块，通过该存储块，数据从一个处理器（或外设）传输至另一个处理器。接收处理器读取来自一个缓冲器的已发送数据，同时，发送处理器在另一个缓冲器中准备下一次发送的数据。

**双精度浮点 [double-precision floating-point]** — 具有 64 位及一个其他隐藏位的浮点数。

**双字 [doubleword]** — 一个 64 位值。

- DSP 中断 (DSPINT) bit** — 启用或禁用主机处理器到 DSP 中断的字段。DSPINT 位从主机处理器写入；DSP 写入对 DSPINT 位无影响。当 DSPINT = 1 时，将产生 DSP 中断。在写入到字节排序位 (BOB) 或 DSP 到主机处理器中断 (HINT) 位时，主机必须向 DSPINT 位写入一个 0，这样主机就不会调用不必要的 DSP 中断。此位存储在主机端口接口控制 (HPIC) 寄存器中。
- DSP 到主机处理器中断 [HINT] 位** — 用于启用或禁用 DSP 到主机处理器的中断的字段。复位时，HINT = 0。此位存储在主机端口接口控制寄存器 (HPIC) 中。
- 双列直插封装 [dual in-line package, DIP]** — 一款常用的长方形芯片，引线（引脚）位于长方形的两个长边上。
- 虚拟周期[dummy cycle]** — CPU 使用同一地址重新加载程序计数器的 CPU 周期。
- 动态存储器分配[dynamic memory allocation]** — 由多种函数（如 malloc, calloc 和 realloc）用于在运行时为变量动态分配存储器的技术。这通过定义较大的存储器池（堆）并使用函数分配堆中的存储器来实现。
- 动态电源管理 [dynamic power management , DPM]** — 支持 DPM 的充电器具有根据测得的输入电压或电流减少输出功率的内部反馈回路，以防止输入源失效。
- 动态电源路径管理 [dynamic power-path management, DPPM]** — 除了将电池充电集成电路 (IC) 的输出直接接至电池，支持 DPPM 的充电器在加电时提供一个系统电源轨电压，而不是等待已放电电池的电压达到一个最小值。然后，电池由系统电源轨供电的线性稳压器充电，此线性稳压器具有单独的电流监视和控制功能。
- 动态随机存取存储器 [dynamic random-access memory, DRAM]** — 通常用于外部存储器的存储器。特殊的动态存储器电路；它要求每位信息定期刷新或恢复至其编程状态，以保持有效数据。

## 5 E

**EALU|ROTATE** — 并行处理器的扩展算术逻辑单元 (EALU) 变化，使您可以将桶式旋转的结果保存至一个其他目的寄存器。

**EMIF CE 空间** — 四个存储器空间中的一个，由外部存储器接口 (EMIF) 上的一个芯片使能定义。

**仿真器[emulator]** — 目标系统外的调试工具，可提供对目标系统中的处理器的直接控制。

**emurst** — 复位仿真器的调试程序实用工具。

**使能额外索引寄存器 (NDX) 位** — 决定对辅助寄存器 0 (AR0) 的修改或写入是否也对索引寄存器 (INDX) 和辅助寄存器比较寄存器 (ARCR) 进行同样的修改或写入，以保持器件间的兼容性的字段。此位被存储在处理器模式状态寄存器 (PMST) 中。

**使能多 TREG (TRM) 位** — 表示负载 TREG0 (LT/A/D/P/S) 指令是只加载临时寄存器 0 (TREG0)，还是加载全部三个临时寄存器 (TREG0, TREG1 和 TREG2)，以保持兼容性的字段。这个位被存储在处理器模式状态寄存器 (PMST) 中。

**进入点[entry point]** — 目标存储器中程序开始执行的点。

**包络跟踪 [envelope tracking, ET]** — 包络跟踪描述了一个射频 (RF) 放大器的设计方法，在这个方法中，始终对施加到功率放大器上的电源电压进行调节，以确保放大器运行在峰值频率上，满足指定的瞬时输出功率要求。

**环境变量[environment variable]** —

1. 调试程序用来查找目录或获得调试程序选项的特殊系统符号。
2. 可用于修改编译器或链接器的命令行输入，或修改环境的系统符号。
3. 系统符号被定义，并且被分配给一个字符串。它们通常包括在诸如 .cshrc 的批处理文件中。

**收尾程序[epilog]** — 函数中代码的一部分，用于恢复堆栈并返回。另请参见 *管道循环收尾程序*。

**可擦除可编程只读存储器 [erasable programmable read-only memory, EPROM]** — 一种只读存储器，其中存储的数据可通过紫外线灯或其它方式擦除。它可通过适当的电压脉冲逐位重新编程。

**事件[event]** — 可以导致任务开始执行的触发。

**事件标志[event flag]** — 存储器中绑定到一个端口或信标的位，用于指示特定事件是否发生。

**事件引脚[event pin]** — 软件开发板 (SDB) 中断控制器上的一个引脚，在置为有效后可触发事件。

**事件寄存器[event register]** — 任务描述符中包含相关任务的 32 个事件标志的字段。

**异常[exception]** — 正常任务程序流程之外被处理的情况。任务中的异常是相当于处理器中硬件中断的软件状况。

**异常标志[exception flag]** — 任务异常寄存器中的位，表示特定异常类型的状态。

**异常处理程序[exception handler]** — 与任务关联的函数，用于处理任务中发生的异常。异常处理程序响应任务中异常的方式，与中断处理例程 (ISR) 响应处理器中硬件中断的方式类似。

**异常寄存器[exception register]** — 任务描述符中包含相关任务的 32 个异常标志的字段。

**可执行模块[executable module]** — 可在目标系统中执行的链接对象。

**执行数据包[execute packet]** — 并行执行的指令块。

**执行相位[execute phase]** — 多相位处理器管道的最后相位；执行指令的相位。另请参见 管道。

**执行阶段[execute stage]** — 主处理器获取、执行、存取 (FEA) 管道的第二个阶段和并行处理器获取、寻址、执行 (FAE) 管道的第三个阶段。这些阶段的运行对于主处理器和并行处理器是不同的。对于主处理器的执行阶段，将指令解码，从寄存器读取源操作数，执行操作，并将结果写入目的寄存器。对于并行处理器，进行所有数据单元操作，对存储器进行访问（加载和存储）并在寄存器到寄存器之间移动。

**执行程序[executive]** — 多任务软件系统的一部分，负责执行应用任务，从而提供任务间的通信，并管理共享资源。

**退出列表[exit-list]** — 每次任务退出时由内核执行的函数的列表。

**扩展器[expander]** — 对于并行处理器，通过分别将 1 位、2 位或 4 位值扩大 32、16 或 8 倍来扩展到 32 位字的算术逻辑单元 (ALU) 数据路径硬件。要扩展的值可以包含在多标志 (mf) 寄存器中，后者是唯一一连接到并行处理器的扩展器的寄存器。

**出口管制提示 [Export Control Notice, ECN]**— 随技术信息发布版本一起提供给供应商的提示，此提示包含特定用语，以严格遵守政府出口管制条例。

**出口控制分类编号 [Export Control Classification Number, ECCN]**— 商业控制清单 (CCL) 上使用的 5 字符混合符号名称，以表示针对出口管制的两用物项。

**表达式[expression]** — 汇编器编程中的一个或多个运算，由符号、常数和成对括号的组合表示，并由算术运算符隔开。

**扩展算术逻辑单元 [extended arithmetic logic unit, EALU]** — 并行处理器的操作集，用于通过在 D0 寄存器（而非操作码）中指定算术逻辑单元 (ALU) 运算或乘法器功能来扩展普通 ALU 函数。

**扩展精度浮点格式[extended-precision floating-point format]** — 浮点数的 40 位表示，带 32 位尾数和 8 位指数。

**扩展精度寄存器[extended-precision register]** — 主要用于扩展精度浮点计算的 40 位寄存器。浮点运算使用扩展精度寄存器的位 39–0。然而，整数运算只使用位 31–0。

**外部存取有效选通 [ $\overline{\text{STRB}}$ ]** —  $\overline{\text{STRB}}$  在访问外部程序、数据 I/O 空间时被置为有效。

**外部标志 [external flag, XF] 引脚** — 一种通用输出引脚，其状态可通过状态寄存器 ST1 中的 XF 位读取或更改。

**外部标志 [external flag, XF] 引脚状态位** — 驱动外部标志 (XF) 引脚电平的字段。复位时，XF = 1。此位存储在状态寄存器 1 中。

**外部中断** — 由引脚上特定值触发的硬件中断。

**由外部启动的数据包传输 [externally initiated packet transfer, XPT]** — 通过 XPT [2:0] 输入，由外部器件启动的数据包传输。

**外部符号** — 此类符号为：

1. 在当前模块中定义，在另外一个模块中存取。
2. 在当前模块中存取，但在另一个模块中定义。

**6 F**

**快速傅立叶变换 [fast Fourier transform, FFT]** — 计算离散傅立叶变换算法的有效办法，该算法在时间域和频率域之间转换函数。时间域到频率域被称为正向变换，频率域到时间域被称为反向变换。

**快速模式[fast mode]** — 主处理器 (MP) 浮点单元模式，在该模式下，执行 MP 浮点指令时所有非常规数对于输入操作数和输出结果均作为 0 处理。

**故障[fault]** — 导致系统故障的任何情况。

**FET 本身导通电阻 [ $R_{DS(on)}$ ]**— FET 的导通状态电阻（不包括任何 FET 以外的电阻）。

**获取[fetch]** — 计算机周期的一部分，在此期间，将从存储器检索下一指令。

**获取、寻址、执行 [FAE] 管道序列** — 并行处理器的指令执行单元管道。获取阶段包括指令和操作数获取；寻址阶段包括地址计算和可能的交叉访问；执行阶段包括数据单元操作、存储器存取和寄存器到寄存器移动。

**获取、执行、存取 [FAE] 管道序列** — 主处理器的指令执行整数单元管道。获取阶段包括指令和操作数获取；执行阶段包括操作和寄存器到寄存器移动；可选的存取阶段包括存储器访问（加载和存储）。

**获取数据包[fetch packet]** — 包含多达 8 条指令的程序数据块。

**获取阶段[fetch stage]** — 主处理器获取、执行、存取 (FEA) 管道和并行处理器获取、寻址、执行 (FAE) 管道的第一个阶段，在此期间，将获取指令及其操作数。另请参见 *FEA 管道顺序* 或 *FAE 管道顺序*。

**字段[]** — 可用软件配置的数据类型，长度可编程为 1 到 8 位。

**字段提取移动[field extract move]** — 并行处理器的寄存器到寄存器移动，将源寄存器中的指定字节或半字节右对齐，并将带 0 或符号扩展的结果写入目标寄存器。源寄存器必须是 D 寄存器，但目的寄存器可以是任一并行处理器寄存器。

**字段复制移动[field replicate move]** — 并行处理器的寄存器到寄存器移动，用于复制源寄存器中的低字节或半字节来填充 32 位，并将结果写入目的寄存器。源寄存器必须是 D 寄存器，但目的寄存器可以是任一寄存器。

**FIFO 缓冲器[FIFO buffer]** — 数据存储后，将按照与存储相同的顺序进行检索的存储器部分。同步串行端口具有两个四字深 FIFO 缓冲器：一个用于发送操作，一个用于接收操作。

**FIFO 标志** — 根据 FIFO 状态设定或清除的指示符。

**FIFO 接收中断位 [FR0], [FR1]** — 同步串行端口控制寄存器 (SSPCR) 中的位，用于根据接收 FIFO 缓冲器中的字数设置中断触发条件。

**FIFO 发送中断位 [FR0], [FR1]** — 同步串行端口控制寄存器 (SSPCR) 中的位，用于根据发送 FIFO 缓冲器中的字数设置中断触发条件。

**文件标头[file header]** — 常用对象文件格式 (COFF) 对象文件的一部分，包含关于该对象文件的一般信息，诸如段标头数，对象文件可被下载的系统类型，符号表中的符号数量，以及符号表的起始地址。

**文件级优化[file-level optimization]** — 一种优化级别，汇编程序使用其具有的有关整个文件的信息来优化代码（相对于程序级优化，汇编程序使用其具有的有关整个程序的信息来优化代码）。

**文件窗口[file window]** — 显示当前 C 代码内容的窗口。FILE 窗口主要用于显示 C 代码，但可用于显示任何文本文件。（此窗口是所有 TI 调试程序的标准图形用户界面的组成部分。）

- 填充值传输[fill-with-value transfer]** — 一种源传输类型，此类型不传输数据，而是在数据包传输参数中指定一个源值。此填充值用于填充由目标传输参数定义的目标存储器。
- 先入、先出 [first in, first out, FIFO]** — 一个队列；项目按其输入顺序输出的数据结构或硬件缓冲队列。缓冲未同步的发送器和接收器之间的数据流时，FIFO 非常有用；即，接收器和发送器未按完全相同的速率发送和接收项目。如果在一个方向上速率差别太大且时间太长，FIFO 会变满（阻断发送器）或变空（阻断接收器）。
- 固定插入码引导传输[fixed-patch guided transfer]** — 使用包含 32 位字对齐条目的片上引导表的引导传输。另请参见 *引导传输*。
- 标志[flag]** — 二进制状态指示符，其状态表示特定条件是否已发生或生效。
- 标志偏移量[flag offsets]** — FIFO 存储器件中的偏移量，决定何时设定或清除 FIFO 标志。
- 闪存存储器[flash memory]** — 非易失性只读存储器，可电子擦除和编程。
- 浮点加法单元忙[floating-point add unit busy]** — 此情况下，浮点加法单元处于停滞状态或其管道已满；因此该单元无法接收新指令。
- 浮点为空[floating-point empty]** — 此情况下，浮点单元的任何管道阶段中均无指令。
- 浮点异常处理程序例程[floating-point exception handler routine]** — 用于处理主处理器浮点异常陷阱，以向用户提供信息或尝试恢复的例程。
- 浮点乘法单元忙[floating-point multiply unit busy]** — 此情况下，主处理器浮点乘法单元处于停滞状态或其管道已满；因此该单元无法接收新指令。
- 浮点单元 [floating-point unit , FPU]** — 主处理器的 IEEE 标准 754 硬件，包含一个完整的双精度浮点加法单元和一个具有单精度内核的双精度浮点乘法单元。
- 外部节点[foreign node]** — 特定任务的可执行代码所驻留节点以外的处理器节点。此项是相对的。在特定节点  $n$  上执行的任务将此节点当做其本地节点，但在其它节点上执行的任务会将节点  $n$  当做外部节点。
- 格式 [format, FO] 位** — 指定串行端口发送器和接收器的字长的字段。数据传输时，最高有效位 (MSB) 在前。复位时， $FO = 0$ 。此位存储在串行端口控制寄存器 (SPC) 和 TDM 串行端口控制寄存器 (TSPC) 中。
- 格式扩展 [format extension, FE] 位** — 与格式 (FO) 位配合使用来指定缓冲串行端口 (BSP) 发送器和接收器的字长的字段。当  $FO = FE = 00$  时，格式为 16 位字；当  $FO = FE = 01$  时，格式为 10 位字；当  $FO = FE = 10$  时，格式为 8 位字；当  $FO = FE = 11$  时，格式为 12 位字。对于 8、10 和 12 位字，接收的字将右对齐，且符号位将被扩展，以形成 16 位字。要发送的字必须右对齐。复位时， $FE = 0$ 。此位存储在 BSP 控制扩展寄存器 (SPCE) 中。
- 小数模数 [fractional modulus , FDEN]** — 用于分频 PLL 中小数字的分母。
- 小数 N 分频 PLL** — N 分频值可以是一个小数的 PLL。
- 低分数寄生信号[fractional spur]** — 出现在运行频率为比较频率数倍的小数 N 分频 PLL 上的寄生信号，此比较频率由 PLL 导致的小数模数分频。(Spurs that occur in a fractional N PLL at multiples of the comparison frequency divided by the fractional modulus that are caused by the PLL.)
- 帧** — 单个垂直扫描期间的屏幕图像输出。

- 帧忽略 [frame ignore, FIG] 位** — 仅用于具有外部帧的发送连续模式和接收连续模式中的字段。复位时, FIG = 0。此位存储在 BSP 控制扩展寄存器 (SPCE) 中。
- 帧锁定 [frame lock]** — 视频控制器 (VC) 模式, 允许两个帧定时器锁定在一起, 这样它们就能保持同步。这在采集和显示系统中需要不同的定时信号参数时很有用, 但是另一方面需要协调这两个系统。
- 帧同步模式 [frame synchronization mode, FSM] 位** — 指定串行端口操作是否需要帧同步脉冲 (FSX 和 FSR) 的字段。复位时, FSM = 0。此位存储在串行端口控制寄存器 (SPC) 和 TDM 串行端口控制寄存器 (TSPC) 中。
- 帧同步极性 [frame synchronization polarity, FSP] 位** — 决定帧同步脉冲状态的字段。复位时, FSP = 0。此位存储在 BSP 控制扩展寄存器 (SPCE) 中。
- 帧同步 [帧同步] 模式 [frame synchronization [frame sync] mode]** — 同步串行端口中决定连续数据传输之间是否需要帧同步脉冲的两个模式中的一个。另请参见 *突发模式; 连续模式*。
- 帧同步 [帧同步] 脉冲** — 标志着从同步串行端口开始发送或开始接收到同步串行端口的脉冲。
- 帧定时器 [frame timer]** — 视频控制器 (VC) 中提供视频定时控制, 并在需要串行寄存器传输 (SRT) 时向 SRT 控制器作出指示的定时器。
- 组帧错误 [framing error]** — 当异步串行端口收到没有有效停止位的数据字符时出现的错误。另请参见 *组帧错误指示符 [FE] 位*。
- 组帧错误指示符 [framing error indicator, FE] 位** — I/O 状态寄存器 (IOSR) 中的位, 指示将字符接收到异步串行端口中期间是否检测到有效的停止位。
- 自由位 [free bit]** — 当在高级语言调试程序中遇到断点时, 与 **Soft** 位配合使用来决定串行端口或定时器状态的字段。复位时, Free = 0。
- 频率跃变 [F<sub>j</sub>]** — 在讨论 PLL 的瞬态响应时, 这是指 PLL 初始频率与最终目标频率间的频率差异。
- 频率合成器 [frequency synthesizer]** — 这是一个具有高频分频器 (N 分频) 的 PLL, 可用于合成多种信号。
- 频率容差 [frequency tolerance, tol]** — 在计算或测量锁定时间时, 这是可接受的频率误差。如果频率误差小于频率容差, PLL 被称为处于锁定状态。它的典型值为 500Hz 或 1kHz。
- 前沿 [front porch]** — 消隐开始和相应同步脉冲之间的视频波形间隔。水平前沿被指定为整数个帧时钟 (FCLK) 周期; 垂直前沿被指定为整数行 (对于隔行扫描模式为半行)。另请参见 *后沿*。
- 函数内联 [function inlining]** — 在调用点为函数插入代码的过程。这可以节约函数调用的开销, 并允许优化程序在周围代码的上下文中优化函数。

## 7 G

**增益级 [gain stage]** — 增加信号的那部分电路；以及让数字信号获得增益的那部分算法。

**伽玛优化参数 [gamma optimization parameter]** — 对锁定时间有一定影响的环路滤波器参数。通常选择接近于 1 的值，但是不精确。

**通用输入/输出 [GPIO] 引脚 [general-purpose input/output [GPIO] pins]** — 可用于接收输入信号或发送输出信号，但未链接到特定用途的引脚。

**全局地址单元 [global address unit]** — 使用全局寻址寄存器（a8-a12, a14, a15），通过全局索引寄存器（x8-x10）的预加、后加、预减或后减，或者直接偏移量来生成地址的并行处理器硬件。

**全局数据存储器空间 [global-data memory space]** — 四个存储器空间中的一个。全局数据存储器空间可以与系统内的其它处理器共享数据或作为其他的数据存储器空间。

**全局中断使能 [global interrupt enable, GIE]** — 控制状态寄存器 (CSR) 中的位，用于启用或禁用可屏蔽中断。

**全局合法 [global legal]** — 并行处理器可对全局端口进行访问的权限，这是因为地址处于共享 RAM 中。如果访问并非全局合法，将由传输控制器 (TC) 执行直接外部存取 (DEA)。并行处理器的管道将在 DEA 完成前处于停滞状态。

**全局端口 [global port]** — 可访问任何共享 RAM（任何并行处理器数据 RAM 或参数 RAM）的数据端口。

**全局符号 [global symbol]** — 此类符号为：

1. 在当前模块中定义，并在另一个模块中访问的符号。
2. 在当前模块中访问，但是在另一个模块中定义的符号。

**全局传输 [global transfer]** — 由全局地址单元执行的并行处理器运行。这可以是存储器访问（加载或存储）、地址单元算术或寄存器到寄存器移动。

**图形用户界面 [graphical user interface, GUI]** — 提供窗口、菜单、对话框、图标、列表和选项的屏幕格式，使您可以通过指向图标，使用鼠标选择项目或使用键盘来启动程序或者执行任务。

**图形输出显示器线缆 [graphics output monitor cable]** — 显示器制造商提供的线缆，可将视频图形阵列 (VGA) 源连接至显示器。

**灰阶或灰度 [grayscale or greyscale]** — 一系列精确已知的灰度梯度，打印出来用于校准显示器或打印机中的相关梯度。在图形中，由不连续的灰度梯度组成。对于显示器，颜色格式中的每一个像素都由一个 8 位值决定。此值通过 8 位值带有的红色、绿色和蓝色分量映射到 RGB 空间中。结果是从黑到灰，再到纯白的像素。

**接地带 [ground strap]** — 条状导电材料，一端接地，另一端连接到人体或物体上，这样静电就可以消散到地面并绕过电子电路。通常在人们处理对静电敏感的各类电子元件或组件时使用。

**组 [group]** — 向 TI 代码生成工具集中的链接器发出的指令，迫使连续分配指定的输出段（作为组）。

**引导传输 [guided transfer]** — 一种传输，其中维地址的序列通过片上存储器表来引导，而不仅仅是通过数据包传输参数值来计算。另请参见 *分维传输 [dimensioned transfer]*。

**引导表 [guide table]** — 描述数据包传输中单独插入码的参数表。另请参见 *插入码 [patch]*。

## 8 H

半字[**halfword**] — 一个 16 位值。

硬件中断[**hardware interrupt**] — 通过与片上外设或外部器件的物理连接触发的中断。

堆[**heap**] — 大型的存储器池，可由应用程序和执行程序在运行时分配。

十六进制命令文件[**hex command file**] — 包含十六进制转换实用工具选项，并为十六进制转换实用工具的输入文件命名的文件。

十六进制转换实用工具 — 该程序接受常用对象文件格式 (COFF) 文件，并将其转换为适合加载到 EPROM 编程器中的多种标准 ASCII 十六进制格式中的一个。

高阻抗模式 - [high-impedance mode , HiZ] 模式 — 对于电池充电器，这个运行模式关闭器件的非必要功能，从而使器件消耗尽可能少量的电流。

高级语言调试[**high-level language debugging**] — 编译程序保留符号和高级语言信息（如类型和函数定义）的能力，这样调试工具就可以使用此类信息。

命中[**hit**] — 处理器获取的指令存在于高速缓存中的情况。

**HOLD** — 使外部器件可以请求外部总线控制的输入信号。如果外部器件将 **HOLD** 引脚驱动为低电平且 CPU 在 **HOLDA** 引脚上发送确认，则外部器件具有总线控制权，直到它将 **HOLD** 驱动为高电平或生成了不可屏蔽的硬件中断。如果未使用 **HOLD**，应将其拉至高电平。

**HOLD** 确认信号 [**HOLDA**] — 由 CPU 发送至 **HOLDA** 引脚的输出信号，以确认正确启动的 **HOLD** 操作。当 **HOLDA** 为低电平时，处理器处于保持状态，并且地址、数据和存储器控制线都可用于外部电路。

保持模式 [hold mode , HM] 位 — 一个 1 位字段，当 **HOLD** 信号启动断电模式时，决定 CPU 是停止还是继续。复位时，HM = 1。此位存储在状态寄存器 1 (ST1) 中。

保持操作[**hold operation**] — 允许直接访问存储器和 I/O 器件的操作。当 **HOLD** 信号启动 **HOLD** 操作后，外部总线进入高阻抗状态，并将 **HOLDA** 信号置为有效。当主机处理器取消触发 **HOLD** 高电平时，总线返回其正常状态，**HOLD** 操作结束。

空洞[**hole**] — 存储器中的区域，未包含任何实际的代码或数据。

水平消隐 [HBLNK] — 双向定时信号，用于启用或禁用像素采集和显示。水平消隐每行出现一次；其脉冲持续时间定义为整数个帧时钟 (FCLK) 周期。另请参见 消隐。

水平同步 [HSYNC] — 复合视频信号的一部分，指示接收器将图像按从左至右的维度放到哪个位置。水平同步脉冲将指示接收系统新扫描线的开始位置。另请参见 垂直同步。

仅主机模式 [**host-only mode , HOM**] — 仅允许主机访问主机端口接口 (HPI) 存储器的模式。在 HOM 下, CPU 无法访问 HPI 存储器块。

主机端口接口 [**host port interface, HPI**] — 一个 8 位并行接口, 供 CPU 用来与主机处理器进行通信。

热插入检测 [**hot-plug detect, HPD**] — 检测插入已加电系统的组件。

整理[**housekeeping**] — 不直接提供计算机程序解决方案, 而是提供程序组织的操作或例程。

HPI 地址寄存器高 [**HPIAH**] 字节 — 存储在主机端口接口地址 (HPIA) 寄存器中的高 8 位。

HPI 地址寄存器低 [**HPIAL**] 字节 — 存储在主机端口接口地址 (HPIA) 寄存器中的低 8 位。

HPI 控制寄存器高 [**HPICH**] 字节 — 存储在主机端口接口控制 (HPIC) 寄存器中的高 8 位。

HPI 控制寄存器低 [**HPICL**] 字节 — 存储在主机端口接口控制 (HPIA) 寄存器中的低 8 位。

**9 I**

**标识符 [identifier, ID]** — 包含资源表索引、序列号和资源类型代码的字段；它标识内核资源，比如端口、信标或任务。

**IEEE-754 浮点单元[IEEE-754 floating-point unit]** — 包含在 TMS320C8x 主精简指令集计算机 (RISC) 处理器内核中的浮点数学单元。

**IEEE-754 标准[IEEE-754 standard]** — 由 IEEE 计算机协会标准委员会推行的一项标准，描述了浮点数的各个方面，比如定义、格式、异常和舍入详细信息等。

**IEEE 1149.1 标准[IEEE 1149.1 standard]** — “IEEE 标准测试访问点和边界扫描架构”，1990 年首次发布。另请参见 *JTAG*。

**IEEE 模式** — 一种主处理器 (MP) 浮点单元模式，在该模式下执行 MP 浮点指令时，非常规数将按 IEEE-754 标准中所定义的进行处理。

**IEEE 标准 1149.1-1990** — 请参见 *JTAG*。

**立即寻址[immediate addressing]** — 获取指令使用的数据值的方法之一。数据值是直接嵌入指令字的常数；不访问数据存储器。

**立即操作数，立即值[immediate operand, immediate value]** — 使用立即寻址的指令中作为操作数给出的常数。

**增量链接[incremental linking]** — 具有几个通路的链接文件。(Linking files in several passes) 增量链接对于大型应用程序十分有用，因为您可以对应用程序分区，分别链接各个部分，然后将所有部分链接在一起。

**间接寻址[indirect addressing]** — 获取指令使用的数据值的方法之一。当一个指令使用间接寻址，数据存储器将由当前辅助寄存器寻址。另请参见 *直接寻址*。

**间接调用[indirect call]** — 一种函数调用，一个函数通过给出被调用函数的地址来调用另一个函数。

**行业标准架构 [Industry Standard Architecture, ISA]**— 扩展行业标准架构 (EISA) 标准的子集。

**init.cmd** — 包含调试程序初始化命令的批处理文件。如果初次调用调试程序时此文件不存在，则所有存储器无效。（这是 TI 调试程序环境的标准组成部分。）

**initdb.bat** — 作为正常调试程序使用的一部分，必须输入 DOS 命令来设置调试程序环境。这样做的最简便方法是编辑 PC 的 autoexec.bat 或者创建只用于此目的的 initdb.bat 文件。

**加载时初始化[initialization at load time]** — 链接 C 代码时由链接器使用的自动初始化方法。链接器在您使用 `-cr` 选项调用链接器时使用此方法。此方法在加载时，而不是运行时，初始化变量。

**初始化段[initialized section]** — 包含可执行代码或数据的常用对象文件格式 (COFF) 段。初始化段可以使用 `.data`、`.text` 或 `.sect` 指令构建。

**启动器[initiator]** — 请求另一 SCSI 器件执行操作的小型计算机系统接口 (SCSI) 设备（通常为主机计算机）。

**init-list** — 每次创建新任务时由内核执行的函数列表。

**输入时钟信号 [input clock signal, CLKIN]** — 在 CLKIN/X2 引脚上提供给片上时钟发生器或由片上振荡器内部生成的时钟源信号。时钟发生器除以或乘以 CLKIN 来生成 CPU 时钟信号 (CLKOUT1)。

- 输入数据定标移位器[input data-scaling shifter]** — 16 到 32 位桶式左移位器，在无周期开销的情况下，在获取周期内将进入的 16 位数据向左移位 0 到 16 个位置（相对于 32 位输出）。也称为 **输入移位器**。
- 输入段[input section]** — 对象文件中链接到可执行模块的段。
- 指令** — 导致操作执行的编程基本单元；它由操作码、操作数及可选的标签和注释组成。
- 指令总线** — 独立于处理器的总线，用于访问来自片上静态 RAM (SRAM) 的指令。每个并行处理器使用 64 位指令总线，而主处理器使用 32 位指令总线。
- 指令高速缓存[instruction cache]** — 片上静态 RAM (SRAM)，包含其中一个处理器执行的当前指令。高速缓存未命中由传输控制器处理。
- 指令高速缓存复位 [instruction-cache reset, ICR]** — 与主处理器 cmdnd 指令一起发送的命令，用于复位所有指令高速缓存标志寄存器和指令最近最少使用 (ILRU) 寄存器。
- 指令解码相位[instruction-decode phase]** — 管道的第二相位；在该相位中，指令将解码。另请参见 **管道**。
- 指令执行相位[instruction-execute phase]** — 管道的第四相位；在该相位中，指令将执行。另请参见 **管道**。
- 指令获取相位[instruction-fetch phase]** — 管道的第一相位；在该相位中，指令从程序存储器中获取。另请参见 **管道**。
- 指令指针 [instruction pointer, IP]** — 指向当前处于管道获取阶段的指令的主处理器寄存器。
- 指令指针寻址 [instruction pointer address, IPA]** — 指向程序寻址管道阶段中当前语句的并行处理器寄存器。
- 指令指针执行 [instruction pointer execution, IPE]** — 指向程序执行管道阶段中当前语句的并行处理器寄存器。
- 指令端口** — 并行处理器用来与其指令高速缓存通信的端口。
- 指令字** — 定点 DSP 编程中，一个指令字是一个 16 位值，表示全部或一半的指令。由 16 位完全表示的指令使用一个指令字。必须由 32 位表示的指令使用两个指令字；第二个字是常数。（在浮点器件中，所有指令字都是 32 位的。）
- 集成电路 [IC]** — 基板材料的微型晶圆，在其上蚀刻或压印有复杂电子元件，以及他们之间的连接线。
- 集成预处理器[integrated preprocessor]** — 与解析器合并的 C 预处理器，可实现更快速编译。同时提供独立预处理或已预处理的列表。
- 集成服务数字网络 [Integrated Services Digital Network, ISDN]** — 从现有电话服务中发展成的全球数字通信网络。其目标是用能传送各种数据（从语音到计算机传输、音乐和视频）的全数字交换和传输设施，来替换现今要求数模转换的电话线。ISDN 构建在两类主要的通信通道之上：**B 通道**，以 64 Kb/s 的速度传送数据；**D 通道**，以 16 或 64 Kb/s 的速度传送控制信息。计算机和其它器件通过简单标准的接口来连接 ISDN 线路。
- 器件内部声音 [I<sup>2</sup>S]** — 数字音频通信方法。
- 内部集成电路（串行通信总线）[I<sup>2</sup>C]** — 串行通信方法。

**隔行扫描模式[interlaced mode]** — 每帧包含两个垂直场的视频模式。一个场显示奇数水平线，另一个场显示偶数水平线。实际上，传输的图片数量加倍，从而减少闪烁。

**列表内实用工具[interlist utility]** — 将您的原始 C 源语句作为注释插入汇编程序的汇编语言输出的实用工具。C 语句被插入等效汇编指令的旁边。

**内部地址[internal address]** — 多媒体视频处理器 (MVP) 芯片内部的地址。从 0x0000 0000 到 0x1FFF FFFF 的地址是片上地址。也称为 *片上地址*。

**内部数据存储块[internal data memory block]** — 组成连续地址组的一组存储器组。'C6x 存储器组是相互交错的，因此半字地址从块中的一个存储器组跨越到下一个组。

**内部中断[internal interrupt]** — 由片上外设引起的硬件中断。

**内部存储器组[internal memory bank]** — 存储器的物理内存组。一次只允许访问单个组一次。

**内部发送时钟分频因子 [internal transmit clock division factor, CLKDV] 位** — 决定内部发送时钟占空比的字段。复位时，CLKDV = 00011。此位存储在缓冲串行端口 (BSP) 控制扩展寄存器 (SPCE) 中。

**节点间消息管理器[internode message manager]** — 一项特殊任务，管理与另一处理器节点的接口来使接口上的消息传输变得容易。跨节点边界路由消息是多媒体视频处理器 (MVP) 执行程序提供的系统服务之一。

**处理器间命令** — 通过交叉开关发送至另一片上处理器的消息。

**中断** — 由硬件或软件发送至处理器，请求注意的信号。中断将指示处理器暂停其当前操作，保存当前任务状态，并执行一组特别指令。中断与操作系统通信并确定要执行任务的优先级。

**中断确认信号 [IACK]** — 一种输出信号，指示已收到中断，且程序计数器正在获取可迫使处理器进入适当中断处理例程的中断向量。

**中断延迟[interrupt latency]** — 请求中断到中断被处理的时间延迟。

**中断模式 [interrupt mode , INTM] 位** — 在全局屏蔽或启用全部中断的字段。当 INTM = 0 时，将启用所有未屏蔽中断。当 INTM = 1 时，将禁用所有可屏蔽中断。INTM 对不可屏蔽的  $\overline{RS}$  和  $\overline{NMI}$  中断没有影响。复位时，INTM = 1。此位存储在状态寄存器 0 (ST0) 中。

**中断处理取指令数据包 [interrupt service fetch packet, ISFP]** — 用来处理中断的取指令数据包。如果 8 个指令不够，用户必须对此块执行分支操作，以获得其他中断处理。如果分支的延迟槽未驻留在 ISFP 中，将从下一 ISFP 中的执行数据包继续执行。

**中断处理例程 [interrupt service routine, ISR]** — 为响应硬件或软件中断而执行的代码模块。

**中断处理表 [interrupt service table, IST]** — 十六个连续的中断处理获取数据包 (ISFP)，每个均对应中断标志寄存器 (IFR) 中的一种情况。IST 驻留在程序存储器系统可访问的存储器中。IST 不得超出 256 字边界 (32 个取指令数据包  $\times$  每个取指令数据包 8 个字)。尽管只定义了 16 个中断，但 IST 中保留了 32 个中断的空间，用于将来的扩展。IST 的位置由中断处理表指针 (ISTP) 寄存器决定。

**中断陷阱表指针 [interrupt-trap table pointer, ITTP]** — 状态寄存器中表示中断陷阱向量表的开始位置 (基址) 的位字段。基址通过将 ITTP 位字段值左移位 8 位形成。

**中断向量** — 使 CPU 进入中断处理例程 (ISR) 的分支指令。

**中断向量位置** — 程序存储器中断向量驻留的地址。确认中断后，CPU 将分支到中断向量位置并获取中断向量。

**中断向量指针 [interrupt vector pointer, IPTR] 位** — 标识中断向量当前在系统中所驻留在的 2K 页的字段。IPTR 位可将中断向量重新映射到 RMA，进行引导加载的操作。复位时，IPTR = 0。这些位存储在处理器模式状态寄存器 (PMST) 中。

**中断向量表 [interrupt vector table, IVT]** — 已排序的地址列表，其中每个地址对应一个中断；当中断发生并启用后，处理器将向存储在中断向量表中相应位置的地址执行分支操作。

**中断向量表指针 [interrupt vector table pointer, IVTP]** — CPU 扩展寄存器文件中包含中断向量表起始地址的寄存器。

**INTn 引脚** — 用于生成通用硬件中断的外部引脚。

**IO0-IO3 位** — IOSR 的 0-3 位。当引脚 IO0-IO3 配置为输入后，这些位将反映引脚上的当前逻辑电平。例如，IO0 位反映 IO0 引脚上的电平。另请参见 CIO0-CIO3 位；DIO0-DIO3 位。

**IO0-IO3 引脚** — 可单独配置为输入或输出的四个引脚。这些引脚可用于连接异步串行端口或作为通用 I/O 引脚。另请参见 CIO0-CIO3 位；DIO0-DIO3 位；IO0-IO3 位。

**I/O 空间选择引脚 [IS]** — DSP 将 IS 置为有效来表示到外部 I/O 空间的访问。

**I/O 空间等待状态位 [I/O-space wait-state bits, ISWS]** — 等待状态发生器控制寄存器 (WSGR) 中的值，用于决定应用到片外 I/O 空间读写的等待状态数。在 'C209 上，ISWS 是 WSGR 的位 2；在其它 'C2xx 器件上，ISWS 是位 11-9。

**I/O 开关** — 仿真器或评估模块 (EVM) 板上的硬件开关，可确定用于仿真器调试器或 EVM 调试器通信的 PC I/O 存储器空间。

## 10 J

日本电子和信息技术行业协会 [**Japan Electronics and Information Technology Industries Association, JEITA**]

一家推动电子产品和元器件健康制造、国际贸易和消费的日本协会，致力于电子和信息技术 (IT) 产业的全面发展，并由此进一步推动日本的经济发展和文化繁荣。

**联合图象专家组 [JPEG] 标准** — 用于压缩静态图片数据的标准。

**联合测试行动组 [Joint Test Action Group, JTAG]** — 联合测试行动组成立于 1985 年，旨在为针对复杂集成电路设计，以及采用表面贴装技术组装的系统开发经济的测试方法。联合测试行动组起草了一份标准，随后由 IEEE 采纳为 IEEE 标准 1149.1-1990，“IEEE 标准测试访问端口和边界扫描架构”。另请参见 *边界扫描：测试访问端口 (TAP)*。

## 11 K

**内核** —

1. 操作系统中管理存储器、文件和外设的核心部分。它还能启动应用程序，分配资源以及维护时间和日期。
2. TMS320C8x 多任务执行程序中的一组低级软件基元，用于执行多任务、消息和信号的传递，以及多个事件的监控。

**内核资源** — 由内核创建并在内核中使用的数据项或对象。内核定义三类资源端口、信标和任务。另请参见 *端口、信标、任务*。

**Kernighan 和 Ritchie [K & R]** — 《C 编程语言》（第二版），作者 Brian Kernighan 和 Dennis M. Ritchie, Prentice-Hall 出版, Englewood Cliffs, 新泽西, 1998 年。此书的作者。

**12 L**

**LA0–LAN[LA0–LAN address pins]** 地址引脚 — 用于数据/程序存储器或 I/O 器件的外部地址引脚。这些引脚位于本地总线上。

**标签 [label]** — 从汇编程序源语句第 1 列开始并对应于该语句的地址的符号。标签是唯一能从第 1 列开始的汇编程序语句。

**锁相位 [latch phase]** — 内部值保持不变的 CPU 周期相位。另请参见 *逻辑相位*；*CLKOUT1*。

**延时 [latency]** — 情况发生时与器件对情况做出反应时之间的延迟。另外，在同一管线中，两个指令的执行间隙需要时延来确保第二个指令使用的值是正确的。

**LD0–LDn 数据总线引脚[LD0–LDn data bus pins]** — 外部数据总线引脚，用于在处理器和外部数据/程序存储器或 I/O 器件之间传输数据。另请参见 *D0–Dn*。

**最低有效位 [least-significant bit, LSB]** — 对二进制数字值影响最小的位，通常是最右边的位。

**最低有效字节 [least-significant byte, LSByte]** — 多字节字中对字值影响最小的字节。

**最左边 1 [leftmost 1, LMo]** — 主处理器 (MP) 或并行处理器的操作，返回值为 1 的最左边位的位置。在 MP 中，*lmo* 是指令；而在并行处理器中，它是位检测函数。

**最左边位变化 [leftmost-bit change, LMBC]** — 并行处理器的操作，返回具有不同于符号位的值的最左边位的位置。

**发光器件 [light emitting device, LED]** — 将施加的电压转换为光照的半导体二极管，用于光源和数字显示。

**线性汇编 [linear assembly]** — 未经寄存器分配或调度的汇编代码，用作汇编程序优化器的输入。线性汇编文件具有 *.sa* 扩展名。

**线路压降 [line dropping]** — 从图像中消除视频线路来垂直调整图像大小的过程。

**行号条目 [line number entry]** — 常用对象文件格式 (COFF) 输出模块中的条目，用于将各行汇编代码映射回创建它们的原始 C 源文件中。

**链接器 [linker]** — 一种软件程序，用于组合对象文件来组成可分配到系统存储器中并由器件执行的对象模块。

**连接器命令文件 [linker command file]** — 包含链接器选项并为连接器命名输入文件的文件。

**列表文件 [listing file]** — 由汇编程序创建的输出文件，用于列出源语句、其行号、任何未解析的符号或操作码，以及它们对段程序计数器 (SPC) 的影响。

**小端序 [little endian]** — 一种寻址协议，字中的字节从右至左编号。字中较高的有效字节具有较高编号的地址。端字节排序视硬件而定，并在复位时确定。另请参见大端序。

**进驻值 [live in]** — 进程之前已定义且作为该进程输入的值。

**派出值 [live out]** — 进程内已定义且作为该进程输出的值。

**加载 [load]** — 将数据输入存储或工作寄存器。

**载入地址 [load address]** — 加载代码段的起始地址。

- 加载颜色寄存器 [LCR] 周期 [load color register [LCR] cycle]** — 由传输控制器 (TC) 生成的存储器周期，可将指定值写入视频 RAM (VRAM) 颜色寄存器，在块写入周期内使用。LCR 周期仅在 64 位数据总线上受支持，由 STATUS [5:0] = 001101 指示。
- 加载器 [loader]** — 将可执行模块放入系统存储器的器件。
- 加载地址单元 [local address unit]** — 并行处理器硬件，使用本地地址寄存器，通过指数寄存器的预加、后加、预减或后减，或者直接偏移来生成地址。
- 本地总线 [local bus]** — 分配至每个并行处理器的总线，用于访问单周期中的在线 SRAM 数据。
- 本地数据空间 [local data space]** — 未由全局存储器分配寄存器 (GREG) 分配为全局的数据存储器地址部分。如果没有数据存储器地址分配用于全局，则所有数据空间为本地。另请参见 *全局数据空间*。
- 本地合法 [local legal]** — 并行处理器的本地 RAM 中可通过本地端口执行的访问。此类访问在单周期中完成。如果访问尝试通过本地端口执行但并非本地合法，则访问必须转移到全局端口，从而导致管道停滞。
- 本地节点 [local node]** — 本地处理器节点。此术语是相对的。在特定节点 *n* 上执行的任务将节点 *n* 当做其本地节点，但在其它节点上执行的任务会将节点 *n* 当做外部节点。
- 本地端口 [local port]** — 并行处理器数据端口，仅可访问并行处理器的本地 RAM。
- 本地 RAM [local RAM]** — 与多媒体视频处理器 (MVP) 中特定并行处理器关联的片上 RAM。
- 本地传输 [local transfer]** — 本地地址单元运行。这是通常通过本地端口发生的存储器存取。
- 已锁锁相环 [locked PLL, PLL]** — 由 *N* 分频的输出频率等于可接受容差范围内比较频率的 PLL。
- 锁定时间 [lock time]** — 对于指定容差范围内给出的频率跃变，PLL 从初始频率切换至最终频率所花费的时间。
- 逻辑地址 [logical address]** — 器件指令引用的存储器位置；逻辑地址是处理器的逻辑存储器映射的一部分。
- 逻辑单元 [logical unit]** — 可通过目标寻址的物理或虚拟器件。
- 逻辑单元编号 [logical unit number, LUN]** — 逻辑单元的 3 位代码。
- 逻辑相位 [logic phase]** — 内部值变化的 CPU 周期相位。另请参见 *锁相位; CLKOUT1*。
- 长立即数 [long immediate]** — 在主处理器 (MP) 上的 32 位值，可以是有符号或无符号的整数、单精度浮点常数、相对分支字偏移量或者绝对地址。MP 长立即指令格式要求两个 32 位指令字。
- 长立即值 [long-immediate value]** — 作为指令（使用立即寻址）操作数的 16 位常数。
- 长偏移量 [long offset]** — 半字或字传输的无符号 15 位偏移量，或者字节传输的无符号 16 位偏移量。
- 查询表 [look-up table, LUT]** — 数字图像的扫描转换过程中使用的表，用于将颜色映射地址转换为显示的实际颜色值。
- 循环 [loop]** — 重复执行的指令序列，直至终止条件发生。
- 环路带宽 [loop bandwidth,  $\omega_c$  或  $F_c$ ]** — 开环传输函数幅度等于 1 时的频率。 $\omega_c$  是以弧度为单位的环路带宽，而  $F_c$  是单位为 Hz 的环路带宽。

- 环路滤波器 [loop filter]** — 获取电荷泵输出电流，并将其转换为压控振荡器 (VCO) 所使用的调节电压的低通滤波器。Z(s) 通常用来代表这个功能的阻抗。虽然不是完全准确，某些人将环路滤波器视为一个积分器。
- 环路滤波器阻抗 [loop filter impedance, G(s)]** — 这代表乘以 VCO 增益和电荷泵增益，并除以 s 的环路滤波器阻抗。
- 环路增益常量 [loop gain constant, K]** — 用来推导出很多结果的中间计算值。
- 循环展开 [loop unrolling]** — 用来扩展小循环的优化，使循环的每次迭代出现在你的代码中。尽管循环展开增大了代码大小，它可以提高代码的效率。
- 低颜色寄存器周期 [low color register cycle, LCR]** — 由传输控制器 (TC) 生成的存储器周期，可将指定值写入视频 RAM (VRAM) 颜色寄存器，在块写入周期内使用。LCR 周期仅在 64 位数据总线上受支持，由 STATUS [5:0] = 001101 指示。
- 低压降 [low dropou, LDO]** — 可以以极低小输入/输出差分电压运行的直流线性稳压器。
- 低通滤波器 [low-pass filter, LPF]** — 过滤频率在某一阈值以上的滤波器。
- LPRAM 处理器 [LPRAM processor]** — 到并行处理器的本地 2K 字节参数 RAM 的并行处理器相关引用。
- LRAM0 处理器 [LRAM0 processor]** — 到并行处理器的本地 2K 字节数据 RAM 组（具有最低地址）的并行处理器相关引用。
- LRAM1 处理器 [LRAM1 processor]** — 到并行处理器的本地 2K 字节数据 RAM 组（具有第二低地址）的并行处理器相关引用。
- LRAM2 处理器 [LRAM2 processor]** — 到并行处理器的本地 2K 字节数据 RAM 组（具有最高地址）的并行处理器相关引用。
- LRU 缓存替换 [LRU cache replacement]** — 最近最少使用的高速缓存替换  
一种高速缓存管理策略，用于发生高速缓存块未命中时，替换存储器中最近最少使用的高速缓存块（同时保留最近较常使用的块）。
- 亮度 [luminance]** — 美国国家电视标准委员会 (NTSC) 或逐行倒相制式 (PAL) 视频信号包含两部分，它们组成您在屏幕上所看到的内容，它们分别是：黑色和白色部分（亮度）和彩色部分（色度）。另请参见 色度。

**M**

宏[**macro**] — 可用作指令的用户定义例程。

宏调用[**macro call**] — 调用一个宏的过程。

宏定义[**macro definition**] — 定义组成宏的名称和代码的源语句块。

宏扩展[**macro expansion**] — 取代宏调用并随后进行汇编的源语句。

宏库[**macro library**] — 由宏组成的存档库。库中的每个文件必须包含一个宏；其名称必须与其定义的宏名称一致，且必须具有 .asm 扩展名。

幻数[**magic number**] — 常用对象文件格式 (COFF) 文件标头条目，用于将对象文件标识为可执行的模块。

邮箱[**mailbox**] — 并行处理器的参数 RAM 中的 32 位字，通过其向客户端（主要是主处理器）传递消息。

尾数[**mantissa**] — 包含小数和符号位的浮点数的组成部分。尾数代表正规化小数，其二进制点按指数移位。

映射文件[**map file**] — 由链接器创建的输出文件，显示存储器配置、段组成、段分配、符号定义，以及为您的程序定义符号的地址。

可屏蔽中断[**maskable interrupt**] — 可通过软件启用或禁用的中断。

屏蔽发生器[**mask generator**] — 并行处理器的算术逻辑单元 (ALU) 数据路径硬件器件，用于获取 5 位输入 n，并在其右对齐后输出包含由输入指定的 n 个 1 的屏蔽。

屏蔽[**masking**] — 选择性忽略一部分数据字的功能。

主时钟输出信号 [**master clock output signal , CLKOUT1**] — 片上时钟发生器的输出信号。CLKOUT1 高脉冲表示 CPU 的逻辑阶段（当内部值发生变化时），而 CLKOUT1 低脉冲表示 CPU 的锁相位（当相关值保持不变时）。

主相位[**master phase**] — 通用精简指令集计算机 (RISC) 处理器，用于协调多媒体视频处理器 (MVP) 上其他处理器的活动。MP 包括 IEEE-754 浮点硬件单元。

主处理器 [**master processor, MP**] — 请见逻辑相位部分。

最大功率点（跟踪）[**maximum power point (tracking), MPP(T)**] — 提供最大输出功率的高阻抗源（太阳能电池、压电能量采集器等）的状态（电压、频率等）。当这个点随环境条件发生变化时，需要一个跟踪电路/算法来保持最大功率提取。

MCBL/MP 引脚 — 用于在微处理器模式和微型计算机模式之间做出选择的引脚。MCBL/MP 高电平选择微型计算机模式；MCBL/MP 低电平选择微处理器模式。

兆赫 [**MHz**] — 每秒百万个周期。

成员[**member**] — 结构、联合或枚举的元素。

存储器映射[**memory map**] — 计算机系统存储器的图形表示，显示程序空间、数据空间、保留空间和其它存储器驻留元素的位置。

存储器映射寄存器[**memory-mapped register**] — 映射到存储器中地址的片上寄存器。一些存储器映射寄存器映射到数据存取器，另一些映射到输入/输出存储器。

存储器停滞[**memory stall**] — 等待存储器加载或存储时，CPU 处于停滞状态。

- 存储器宽度[memory width]** — 可存储在单个外部存储器地址中的位数。
- 存储器窗口[memory window]** — 显示存储器内容的窗口。（此窗口是所有 TI 调试程序的标准图形用户界面的组成部分。）
- 菜单栏** — 调试程序显示屏顶部的一排下拉菜单选项。
- 合并模式[merge mode]** — 视频控制器 (VC) 的串行寄存器传输 (SRT) 模式，在该模式下，图像在采集后存储在存储器中。存储器位置不对应已采集图像的保存位置。另请参见捕获模式、显示模式。
- 消息缓冲池[message buffer pool]** — 属于特定任务的一组消息缓冲器，通常大小一致。
- 消息事件[message event]** — 消息到达端口后引起的事件。如果任务正在等待事件，则该任务已计划开始执行。
- 消息头 r** — 消息缓冲器开始的固定大小的结构，包含消息信息、连接至消息处理内核函数的目的端口等信息。
- 消息中断[message interrupt]** — 多媒体视频处理器 (MVP) 硬件机制，通过该机制，一个片上处理器可以在中断启用后向另一片上处理器发出中断信号。
- 消息路由表[message-routing table]** — 为发送至外部处理器节点的消息指定路由端口的表。
- 度量参数[metric parameters]** — 定义音频子系统、显示子系统或视频采集子系统的状态维度和属性的一组参数。
- 微调用堆栈 [microcall stack, MCS]** — 当预取计数器 (PFC) 通过块移动 (BLDD/BLPD)、乘法累加 (MAC/MACD) 和表读写 (TBLR/TBLW) 指令来寻址数据存储寄存器时，临时存储 PFC 内容的单字堆栈。
- 微型计算机[microcomputer]** — 包含微处理器、控制器、存储寄存器、某种算术逻辑单元 (ALU) 以及存储器的集成电路。
- 微型计算机模式[microcomputer mode]** — 启用片上 ROM 或闪存存储器的模式。此模式通过 MP/MC 引脚选择。另请参见 MP/MC 引脚、微处理器模式。
- 微处理器[microprocessor]** — 可由存储的指令进行编程来执行多种功能的集成电路。
- 微处理器/微型计算机 [microprocessor/microcomputer, MP/MC] 位** — 用来选择微处理器模式或微型计算机模式的 1 位字段。MP/MC 高电平选择微处理器模式。MP/MC 低电平选择微型计算机模式。
- 微处理器模式[microprocessor mode]** — 禁用片上 ROM 或闪存存储器的模式。此模式通过 MP/MC 引脚选择。另请参见 MP/MC 引脚；微型计算机模式。
- 微堆栈 [microstack , MSTACK]** — 当指令需要使用程序计数器 (PC) 来寻址第二个操作数时，用于 PC 值临时存储的寄存器。
- 每秒百万次浮点运算 [millions of floating-point operations per second, MFLOPS]** — 浮点处理器速度的测量方法，计算每秒进行的浮点运算数。
- 多指令流、多数据流 [multiple instruction stream, multiple data stream, MIMD]** — 多个独立处理器组成的并行处理结构。
- 微型 DIN 连接器[mini-DIN connector]** — 按照德国国家标准组织标准使用多个引脚的，与 Deutsch Industrie Norm (DIN) 连接器类似（只是小得多）的连接器。软件开发板 (SDB) 上使用两个 8 引脚 miniDIN 连接器来连接输入/输出外设。

**最小模式[minimal mode]** — 仅显示 COMMAND 窗口, WATCH 窗口和 DISP 窗口的调试模式。

**每秒百万条指令 [million instructions per second, MIPS]** — 每秒百万条指令。

**混合操作[miscellaneous operation]** — 不涉及数据单元的几项操作之一, 包括 nop, eint 和 dint。数据单元的操作码部分用于指定该操作。

**未命中[miss]** — 处理器获取指令时, 指令不在高速缓存中的情况。

**混合模式[mixed mode]** — 在 DISASSEMBLY 窗口中显示汇编语言代码, 同时在 FILE 窗口中显示 C 代码的调试模式。

**助记符[mnemonic]** — 设计用来帮助人类记忆的字符数字符号; 通常表示由汇编程序转换成机器代码的汇编语言指令名称的操作代码。

**移动行业处理器接口 [mobile industry processor interface, MIPI]** — 为移动和移动应用行业开发接口技术规范开放式成员组织。

**模式位[mode bit]** — 中断控制寄存器 (ICR) 中的位, 决定 HOLD/INT1 引脚仅下降沿敏感, 还是下降沿和上升沿均敏感。

**模式 1 语句** — 每次调用宏时, 宏定义中汇编的指令或汇编指令。

**模块端口扫描器件 [modular port scan device, MPSD]** — 在 'C3x 的串行扫描路径上实现完全仿真的技术。

**调制域分析器[modulation domain analyzer]** — 显示输入信号频率与时间之间关系的 RF 设备。

**调制指数 [ $\beta$ ]** — 这参考的是正弦调制 RF 信号。(This is in reference to a sinusoidally modulated RF signal.)

**监视器定时[monitor timing]** — 显示应用编程接口 (API) 用来决定需要怎样的信号速率来驱动监视器的参数。

**单通道模式[mono mode]** — 一种音频编解码器模式, 在该模式下, 只存在一个声道。

**最高有效位 [most significant bit, MSB]** — 一个字中的最高顺序位。复数形式 (MSBs) 代表指定数目的高位, 以最高序位开始, 向右计数。例如, 16 位值的 8 个 MSB 是指位 15 到 8。

**最高有效字节 [most significant byte, MSByte]** — 多字节字中对字值影响最大的字节。

**主板[motherboard]** — 包含处理器、主存储器、电路、总线控制器、连接器和计算机主要组件的主电路板。TMS320C8x 软件开发板 (SDB) 通过外设组件互连 (PCI) 本地总线连接至主板。取决于计算机系统, 子板、扩展板、输入/输出板、其它板或其他存储器, 也可能通过总线控制器连接至主板。

**鼠标光标[mouse cursor]** — 在整个显示屏上跟踪鼠标移动的块状光标。

**运动图象专家组标准 [moving picture experts group standard, MPEG]** — 建议的压缩视频数据标准。

**主处理器编译器 [master processor compiler, MPCL]** — 调用多媒体视频处理器 (MVP) 主处理器编译器、汇编器和链接器来创建主处理器程序的可执行对象文件版本的外壳实用工具。

**MP/MC 引脚** — 用于在微处理器模式和微型计算机模式之间做出选择的引脚。MP/MC 高电平选择微处理器模式; MP/MC 低电平选择微型计算机模式。

**多媒体视频处理器 [multimedia video processor , MVP]** — 加速诸如视频压缩和解压缩、图像处理和图形等应用的单芯片多处理器器件。多媒体视频处理器包含主处理器和一到八个并行处理器，这具体取决于器件版本。

**多路复用[multiplexing]** — 通过单根电线或一个通信链接传输多组信号的过程。（也被称为复用。）

**多元算术[multiple arithmetic]** — 并行处理器的算术逻辑单元 (ALU) 操作，其中 ALU 内某些点上的位到进位被禁用，导致 ALU 运行为多个较小的 ALU。每个 ALU 段的状态位（进位输出或 0）保存在 mf 寄存器中。多元算术也称为拆分 ALU。

**多字节算术[multiple-byte arithmetic]** — ASize 设为字节的多元算术。算术逻辑单元 (ALU) 运行为四个并行字节 ALU，因为进位路径在位 24 和 23，位 16 和 15 以及位 8 和 7 之间断开。

**多半字算术[multiple-halfword arithmetic]** — ASize 设为半字的多元算术。算术逻辑单元 (ALU) 运行为两个并行 ALU，因为进位路径在位 16 和 15 之间断开。

**乘法器 MULT [multiplier]** — 执行乘法并生成乘积的 CPU 部分。乘法器使用符号或有符号补码算术进行运算。

**乘法和累加 [multiply and accumulate, MAC]** — 计算两个数的乘积，并将此乘积添加至累加器中的常用步骤。

**多频显示器[multisync monitor]** — 自行调整至视频信号的水平同步率和垂直同步率的显示器。多频显示器可与多种视频适配器配合使用。

**多任务执行[multitasking executive]** — 多任务软件系统的一部分，负责执行应用任务，实现任务间的通信，并管理共享资源。另请参见 *执行程序*。

**互斥[mutual exclusion]** — 一组资源共享技术，这样不同的用途就不会冲突和产生不必要的交互。最常用的互斥技术之一为信标。另请参见信标。

**N**

**N分频器[divider]** — 用因子 N 对高频（和相位）输出进行分频的分频器。

**命名段[named section]** —

1. 由 .sect 指令定义的已初始化段。
2. 由 .usect 指令定义的未经初始化的段。

**全国电视标准委员会 [National Television Standards Committee, NTSC]** — 一款彩色电视广播标准，其中的图像由 525 扫描行的格式组成。60Hz 场频 4MHz 广播带宽 15.75kHz 线频 三十分之一秒帧频 3.58MHz 色彩子载波频率 另请参见 PAL。

**固有频率 [ $\omega_n$ ]** — 对于一个二阶瞬态响应，此频率为频率响应振铃的频率。

**负温度系数 [negative temperature coefficient, NTC]** — 材料中随温度上升而下降的物理属性。

**嵌套中断[nested interrupt]** — 完成当前中断处理例程 (ISR) 前必须执行的较高优先级中断。执行 ISR 可以设置中断屏蔽 (IMR) 位来防止让另一中断挂起。

**下一辅助寄存器[next auxiliary register]** — 完成执行修改辅助寄存器指针 (ARP) 的指令后，ARP 指向的寄存器。另请参见辅助寄存器；当前辅助寄存器。

**下一程序地址寄存器 [next program address register, NPAR]** — 程序地址生成逻辑电路的组成部分。此寄存器将下一指令的地址提供给程序计数器 (PC)，程序地址寄存器 (PAR)，宏堆栈 (MSTACK) 或堆栈。

**节点[node]** — 能发送和接收消息的任何实体，比如处理器端口。

**全局节点端口[node-global port] ID** — 包含范围在 0 到 127 之内的显式节点数的端口 ID。在多处理器系统中，需要全局节点目的端口 ID 跨越处理器节点边界来发送消息。

**本地节点端口[node-local port] ID** — 不包含显式处理器节点号，但处理器节点数为本地隐式的端口 ID。本地节点端口 ID 的八个最高有效位 (MSB) 中的资源类型代码被设定为 -1。

**节点号[node number]** — 唯一标识多处理器系统中每个处理器节点的小值非负整数。全局节点端口 ID 包含一个 0 到 127 之间的节点号。

**非 D 操作数[non-D operand]** — 用作并行处理器指令中的操作数的非 D 寄存器。

**逐行扫描图形模式[noninterlaced graphics mode]** — 一种视频控制器 (VC) 模式，在该模式下，每帧包含单个垂直场。一种扫描视频显示的方法，帧中的所有线连贯按序扫描。也称为顺序扫描。

**逐行扫描模式[noninterlaced mode]** — 一种视频控制器 (VC) 模式，在该模式下，每帧包含单个垂直场。

**不可屏蔽中断 NMI** — 无法屏蔽或禁用的中断。

**非确认 [not acknowledge, NACK]** — 用来表示发出的消息已损坏或有错误的传输控制字符，或者表示接收站未作好接受准备的字符。

## O

对象文件 **[object file]** — 作为一个单元处理的一组相关记录，对于汇编器或编译器是输出，对于链接器是输入。

对象格式转换器 **[object format converter]** — 将常用对象文件格式 (COFF) 对象文件转换为 Intel 格式，Tektronix 格式，TI 标记的格式或者 Motorola-S 格式对象文件的程序。

对象库 **[object library]** — 由单独的对象文件组成的存档库。

片外 **[off-chip]** — 器件外组件。

片外地址 **[off-chip address]** — 芯片外地址。从 0x0200 0000 到 0xFFFF FFFF 的地址为片外地址。另请参见 *片上地址*。

片上 **[on-chip]** — 器件内部的元件或模块。

片上地址 **[on-chip address]** — 多媒体视频处理器 (MVP) 芯片内部的地址。从 0x0000 0000 到 0x1FFF FFFF 的地址是片上地址。另请参见 *片外地址*。

偏移量引导的查找表传输 **[offset-guided look-up table transfer]** — 一种引导传输类型，其中，引导表包含左移（填零）0、1、2 或 3 位并被添加至数据包传输参数中给定基址的值，以形成每个插入码起始地址的值。另请参见 *引导传输*。

偏移量引导传输 **[offset-guided transfer]** — 一种源引导传输，其中，引导表包含要添加到数据包传输参数中给定基址的值，以形成每个插入码起始地址的值。另请参见 *引导传输*。

如影随形 **[on-the-go, OTG]** — 如影随形是将便携式器件的降压电池充电器反向运行为升压转换器的 USB 标准；从而使用电池为接至便携式器件的正常输入 USB 连接器的外设提供一个经稳压 5V 电源。

操作码 **[operation code, opcode]** — 大多数情况下，描述中央处理单元 (CPU) 操作类型和操作数组合的机器代码的首个字节。

集电极开路输出 **[open-collector output]** — 仅有源驱动低逻辑电平的输出电路。

开环传输函数 **[open loop transfer function, G(s)]** — 通过将 VCO 增益、充电泵增益以及开环阻抗的乘积除以 N 而获得的传输函数。

操作数 **[operand]** — 指定中央处理单元 (CPU) 从何处获取或存储数据的指令部分。操作数包含汇编语言指令、汇编程序指令或宏指令的自变量或参数。

操作数获取相位 **[operand-fetch phase]** — 管道的第三相位；在该相位中，从存储器获取一个操作数或多个操作数。另请参见 *管道*。

操作 **[operation]** — 定义的操作；即，通过一个或多个操作数获取结果的操作，该操作的原则是完全指定允许的操作数组合的结果。由规则指定的一组此类操作，或是规则本身。单个计算机指令指定的操作。由计算机承担或执行的程序步骤，例如，加、乘、提取、比较、移位、传输等。由逻辑元件执行的指定操作。

操作类 **[operation class]** — 通过控制到算术逻辑单元 (ALU) 的操作数路由，使同一 ALU 函数能执行各种完全不同操作的特定多路复用器设置组合。8 个操作类受 ALU 操作基本集的支持。

优化器 **[optimizer]** — 可提高执行速度并减小 C 程序大小的软件工具。

优化 **[optimizer]** — 提高程序执行速度或者减小 C 程序尺寸。

- 选项 [options]** — 调用软件工具时，允许您请求其他的或指定函数的命令参数。
- 可选标头 [optional header]** — 常用目标文件格式 (COFF) 对象文件的一部分，由链接器用于在下载时执行重定位。
- 输出数据调整移位器** — 32 至 16 位桶式左移位器。将 32 位累加器输出向左移位 0 到 7 位以实现量化管理，并将已移位的 32 位数据的高或低 16 位输出至数据写入总线。也称为输出移位器。
- 输出模块 [output module]** — 在目标系统上下载和执行的可执行链接对象文件。
- 输出段 [output section]** — 可执行的链接模块中最终分配段。
- 过流保护 [over-current protection, OCP]** — 在电流达到特定水平时将其关闭，以防止过多电流对器件造成损坏的安全防护措施。
- 溢出** — 算术运算结果超出用于保存该结果的寄存器容量的情况。
- 溢出（寄存器） [overflow (register)]** — 算术运算结果超出用于保存该结果的寄存器容量的情况。
- 溢出（同步串行端口） [overflow (synchronous serial port)]** — 端口的接收 FIFO 缓冲器已满，并且接收移位寄存器 (RSR) 中接收到另一个字时的情况。（此新词未覆盖 FIFO 缓冲器中的任何内容。）
- 溢出标志 [OV] 位 [overflow flag [OV] bit]** — 指示算术运算是否已超出相应寄存器容量的状态位。2) 寄存器 ST0 的位 12；指示算术运算的结果是否已超出累加器的容量。
- 溢出模式 [overflow mode]** — 此模式决定发生溢出时累加器的运行方式。在饱和模式下，累加器中发生溢出将导致累加器加载预设值。如果溢出是正向的，累加器将加载其最大的正数。如果溢出是负向的，累加器将加载其最大的负数。
- 溢出模式 [OVM] 位 [overflow mode [OVM] bit]** — 决定算术逻辑单元中的溢出是否会回绕或饱和的 1 位字段。此位存储在状态寄存器 0 (ST0) 中。
- 覆盖页 [overlay page]** — 物理存储器中映射到与另一存储器段中相同地址范围的部分。硬件开关决定哪个范围处于激活状态。
- 覆盖模式 [overlay mode]** — 混合视频模式。视频图形阵列 (VGA) 直通线缆的输入与随机存取存储器数模 (RAMDAC) 转换器输出的数据混合，以形成 VGA 上显示的视频。
- 溢出 [overrun]** — 异步串行端口的接收器中的一种情况。当异步数据发送和接收寄存器 (ADTR) 中的未读字符被新字符覆盖时会发生溢出。
- 过冲 [overshoot]** — 对于二阶瞬态响应，这是最终稳定至正确频率前，目标频率最初超出的量。
- 溢出（同步串行端口） [OV] 位 [overflow (synchronous serial port) [OV] bit]** — 同步串行端口控制寄存器 (SSPCR) 中的位，指示端口的接收 FIFO 缓冲器已满，而接收移位寄存器 (RSR) 中接收到另一个字。此新词未覆盖 FIFO 缓冲器的内容。

- Px64** — 国际电话和电报咨询委员会 (CCITT) 设计的视频会议标准，用来在位速率为 Px64 ( $1 < P < 30$ ) 的数据线路上实现全双工视频显示。
- 封装字节[packed bytes]** — 包含在一个 32 位字中的四个字节。封装字节字中的单个字节通常使用多字节算术进行并行操作。
- 封装半字[packed halfwords]** — 包含在一个 32 位字中的两个半字。封装半字字中的单个字节通常使用多半字算术进行并行操作。
- 数据包** — 一组数据插入码。另请参见插入码。
- 数据包传输存取模式 [packet transfer access mode, PAM]** — 修改将元数据写入目标的方法（包括正常、外设器件传输、块写入、串行寄存器传输 (SRT)，以及 8 位，16 位，32 位和 64 位源透明度）的模式。
- 数据包传输 [packet transfer, PT]** — 两个存储器区域之间的数据块传输。多媒体视频传输 (MVP) 支持一维、二维或三维数据包传输。另请参见分维传输、引导传输。
- 数据包传输选项字段[packet transfer options field]** — 为来源和目的选择传输形式的数据包传输参数字段。该字段决定数据包是否结束链路列表并启用其他功能（比如特殊传输模式）的选择。
- 数据包传输请求[packet transfer request]** — 当一个数据块要通过数据包传输移动时，提交至传输控制器 (TC) 的 I/O 请求。数据包传输请求可由主处理器、并行处理器、视频控制器 (VC) 或外部器件提交。
- 并行桥接负载 [parallel bridge-tied load, PBTL]** — 使用并联的两个 BTL 输出来驱动一个扬声器。
- 并行调试管理器 [parallel debug manager , PDM]** — 创建并控制多个调试器的程序，以达到在并行处理环境中调试代码的目的。
- 并行逻辑单元 [parallel logic unit, PLU]** — 直接在数据位置上根据长立即操作数或动态位操作寄存器 (DBMR) 的内容执行逻辑操作，而不会影响累加器 (ACC) 或乘积寄存器 (PREG) 内容的 16 位逻辑单元。
- 并行端口[parallel port]** — 尽管并行端口也可用于其它外设，并行打印机端口接口主要用于将打印机连接至计算机系统。在这种情况下，'C3x 数字信号处理器入门套件 (DSK) 被接至并行打印机端口。
- 并行处理器 [parallel processor, PP]** — 一种高级数字信号处理器，用于视频压缩/解压缩 (PX64 或 MPEG)，静态图像压缩/解压缩 (JPEG)，2D 和 3D 图形函数（比如画线、梯形填充、抗锯齿），以及图像数据上的各种高速整数运算。'C8x 单芯片多处理器器件可能包含一到八个并行处理器，具体取决于器件版本。
- 并行处理器命令界面[parallel processor command interface]** — 一种软件界面，主处理器（或其它客户端处理器）通过其发出由服务器并行处理器执行的命令。
- 并行传输** — 与数据单元操作并行指定的地址单元操作。
- 参数 RAM** — 与特定处理器关联的通用 2K 字节 RAM，其中一部分专用于数据包传输信息和处理器中断向量。
- 参数表** — 八个双字长的一组参数，用来描述数据包及其如何从源移动到目的地。
- 解析器[parser]** — 一种软件工具，可读取源文件、执行预处理功能、检查语法，并生成可用作优化器或代码发生器的输入的中间文件。

**部分链接[partial linking]** — 采用多通路的链接文件。Linking files in several passes. 增量链接对于大型应用程序很有用，这是因为可以对应用程序分区，分别链接各个部分，然后将所有部分链接在一起。

**插入码[patch]** — 等长的一组行，其起始地址的间隙等距。

**PC 寄存器[PC register]** — 包含下一指令地址的寄存器。

**挂起中断[pending interrupt]** — 已成功请求但正在等待 CPU 确认的可屏蔽中断。

**周期事件[periodic event]** — 定期重复的事件，相对于仅发生一次或不定期重复的事件。

**周期 [PRD] 寄存器** — 指定片上定时器的周期的存储器映射寄存器。当定时器计数寄存器 (TIM) 递减到零以下时，TIM 将加载 PRD 中的值。

**外设总线[peripheral bus]** — CPU 用于与直接存储器存取 (DMA) 协处理器、通信接口和定时器通信的总线。

**外设线缆[peripheral cable]** — 用于将外设连接至软件开发板 (SDB) 的线缆。SDB 的外设线缆包括视频图形阵列 (VGA) 直通线缆，图形输出显示器线缆，S-VHS 到 RCA 适配器线缆，以及音频输出线缆。

**外设组件互连 [peripheral component interconnect, PCI]** — 支持以 33 MHz 率每秒传输数据高达 132M 字节的高速本地总线。

**个人计算机 [PC]** — 在与硬件和电路板相关的安装说明或信息中，PC 是指个人计算机 (IBM PC)。

**物理地址** — 显示在器件地址引脚的地址。

**逐行倒相制式 [phase alternation line, PAL]** — 与美国全国电视标准委员会 (NTSC) 信号不同的欧洲制式信号，其格式为 625 行，频率为 50Hz。另请参见美国全国电视标准委员会。

**相位检测器[phase detector]** — 产生一个输出信号的器件，此输出信号与其两个输入的相位差成正比。

**相位频率检测器 [phase-frequency detector, PFD]** — 与相位检测器十分相似，但是它还产生一个与频率误差成比例的输出信号。

**锁相环 [phase-locked loop, PLL]** — 使用反馈控制，从固定晶振基准频率中产生输出频率的电路。谨记，PLL 不一定具有 N 分频器。在本例中它的确如此，那么此 PLL 就被称为频率合成器，而这正是本书的主题。

**相位裕度 [Ø]** — 180 度减去环路带宽上开环传输函数的相位后的值。(180 degrees minus phase of the open loop transfer function at the loop bandwidth.) 环路滤波器通常针对 30 至 70 度之间的相位裕度而设计。仿真显示，大约 48 度时产生最快的锁定时间。

**相位噪声[phase noise]** — PLL 输出相位上的噪声。由于相位和频率是相关的，它在频谱分析仪上可见。在环路带宽内，PLL 是主要的噪声源。使用的度量标准为 dBc/Hz (每 Hz 相对于载波的分贝值)。通过减去频谱分析仪的 10 倍频 (分辨率带宽) (subtracting 10x(resolution bandwidth) of the spectrum analyzer)，通常将其标准化为 1Hz 带宽。

**相位噪底[phase noise floor]** — 这是相位噪声减去  $20 \times \log(N)$  后的值。谨记，由于它往往由电荷泵决定，所以它通常不是一个常数，导致在较高比较频率时，变得更加嘈杂。

**引脚栅格阵列 [pin grid array, PGA]** — 集成电路封装。

**引脚到引脚导通电阻 [R<sub>ON</sub>]** — 导通状态下全部引脚到引脚电阻 (包括 R<sub>DS(on)</sub>)，接续线，以及器件整个电流路径内任何其他电阻路径)。

**管道** — 以流水线的方式执行指令的方法。

**管道循环收尾程序[*pipelined-loop epilog*]** — 在软件管道循环中用尽管道的代码部分。另请参见收尾程序。

**管道循环逻辑程序[*pipelined-loop prolog*]** — 在软件管道循环中作为管道开头的代码部分。另请参见逻辑程序。

**管道模式[*pipelined mode*]** — 主处理器 (MP) 浮点单元模式，在此模式下，多 MP 浮点指令同时处于浮点乘法和/或加法单元的不同阶段中。

**管道停滞[*pipeline stall*]** — 正常操作获取的临时停止。可导致管道停滞的事件包括：高速缓存未命中、非法操作检测、本地端口存取转换为全局端口、直接外部存取 (DEA) 和交叉开关争用。

**管道** — 一种设计技术，通过将操作分为一系列阶段（每个阶段执行一部分操作）来减少每个操作的有效传播延迟。通常情况下，一串数据以连续的方式，在一定时间内通过管道，每个时钟周期前进一个阶段。

**视频帧宽[*pitch*]** — 视频帧中一条线开始部分与下一条线开始部分之间的字节数。

**像素** — 一个图像元素 (pel)。

**像素块传输 [*pixel-block transfer , PIXBLT*]** — 每个像素由一个或多个位表示的像素阵列操作。PIXBLT 是 bitBLT 的扩展集，包括常用的布尔函数，以及整数算术和多位运算。另请参见 bitBLT。

**像素削减[*pixel dropping*]** — 从一条视频线移除像素，以垂直向下调节该视频线的过程。

**回放模式[*playback mode*]** — 音频子系统的模式，在该模式下，直接存储器存取 (DMA) 传输提供音频数据进行回放。

**指向[*point*]** — 移动鼠标光标，直至其覆盖屏幕上的目标对象。

**轮询[*poll*]** — 程序使用的连续测试，直至达到所需条件。

**弹出[*pop*]** — 从堆栈移除某字的操作。

**沿** — 对应水平或垂直同步脉冲任意一端消隐间隔的部分视频显示信号。前沿指先于同步脉冲的消隐间隔，而后沿则指紧跟同步脉冲的消隐间隔。另请参见后沿、前沿。

**端口** — 包含消息队列的一类共享数据对象。

**端口地址** — 调试器用来与仿真器通信的 PC I/O 存储器空间。端口地址通过仿真器电路板上的开关进行选择，并使用 -p 调试器选项与调试器通信。

**端口 ID** — 内核用来标识已打开端口的 32 位值。另请参见 ID。

**端口表** — 固定大小的指针表，内核使用其来跟踪已打开的端口。

**后分频器[*postscaler*]** — 后分频移位器。用于后分频累加器 (ACC) 数据的 0-7 位左桶式移位器。

**省电模式** — 处理器进入休眠状态，耗电量较正常运行时少得多的模式。执行 IDLE（闲置）指令后，即启动此模式。省电模式下，所有内部内容维持不变，这样在该模式结束时，操作仍能继续。所有片上 RAM 的内容也保持不变。

**电源管理单元 [*power management unit , PMU*]** — 具有两个或更多集成直流到直流转换器的集成电路 (IC)。

上电复位 [**power-on reset , POR**] — 器件产生一个内部复位时的电压。

电源发送和电源接收 [**Tx**], [**Rx**] — 对于无线电池充电系统, Tx 是电源无线发送端, 而 Rx 是电源接收端。

**pragma** — 指示编译器如何处理特殊语句的预处理器指令。

抢占[**preempt**] — 中断一项任务的处理, 允许开始执行更紧急的任务。

预取计数器 [**prefetch counter , PFC**] — 预取程序指令的寄存器。PFC 包含当前预取的指令地址, 并在新的预取启动后进行更新。

预处理器[**preprocessor**] — 解释宏定义、扩展宏、解释头文件和有条件编译, 并在收到预处理器指令后立即执行的软件工具。

预分频器[**prescaler**] — 将高频压控振荡器 (VCO) 信号分频为较低频率的 N 分频器内部的分频器部件。

预分频器计数器 [**prescaler counter, PSC**] — 定时器控制寄存器 (TCR) 中的位, 此位为片上定时器指定预分频计数。另请参见定时器预分频器计数器。

当前标志[**present flag**] — 与高速缓存子块关联的高速缓存标签寄存器中的位, 指示子块中的信息是否出现在高速缓存中。

印刷电路组装 [**printed-circuit assembly , PCA**] — 一种印刷电路板, 在其上的电路中已安装另外制造的组件部件, 以执行已定义的功能。

专用上下文[**private context**] — 某一特定软件库或相关函数组的专用任务上下文部分, 对主任务程序和程序中使用其它库保密。

处理器 — 计算机的中央处理单元 (CPU) 或微处理器。大多数情况下, 它是指构成计算机内核大脑的一个或多个芯片。

处理器模式状态寄存器 [**processor mode status register , PMST**] — 包含状态和控制位的存储器映射寄存器。

处理器节点[**processor node**] — 多处理器系统中, 执行多媒体视频处理器 (MVP) 指令, 并能通过该指令的节点内消息传递机制与其它处理器通信的处理器。

乘积寄存器 [**product register, PREG**] — 保存乘法器输出的寄存器。PREG 的高字和低字能单独存取。另请参见 乘法器 (*MULT*)。

乘积定标移位器[**product-scaling shifter**] — 执行乘积 0、1 或 4 位左移或 6 位右移的 32 位移位器。左移选项用于管理二进制补码乘法的其他符号位。右移选项用于减小管理中央算术逻辑单元 (CALU) 乘积累加溢出的数。(The right-shift option is used to scale down the number to manage the overflow of product accumulation in the central arithmetic logic unit (CALU).)

乘积移位模式 [**product shift mode, PM**] 位 — 定义乘积移位器 (P-SCALER) 模式的字段。这两个位决定乘积寄存器 (PREG) 输出的移位值 (0、1、4 位左移位器, 6 位右移位器)。它们存储在状态寄存器 1 (ST1) 中。

系统配置窗口 — 显示代码执行统计数据的窗口。此窗口只有处于系统配置环境下才会显示。(此窗口是所有 TI 调试器的图形用户界面的组成部分。)

系统配置 — 决定处理器在程序各个部分所花费时间的技术。

程序地址总线 [**program address bus , PAB**] — 为程序存储器读取和写入提供地址的内部总线。

- 程序地址寄存器 [program address register, PAR]** — 存储程序地址总线当前驱动地址的寄存器，持续时间为完成当前机器周期内计划的全部存储器操作所需的所有周期。
- 程序地址生成逻辑[program-address generation logic]** — 为程序存储器的读写生成地址，以及在要求两个寄存器来寻址操作数的指令中生成操作数地址的逻辑电路。此电路可以在每个机器周期中生成一个地址。另请参见 *数据地址生成逻辑*。
- 程序控制逻辑[program control logic]** — 解码指令、管理管道、存储操作状态及解码有条件操作的逻辑电路。
- 程序控制器[program controller]** — 解码指令、管理管道、存储中央处理单元 (CPU) 状态及解码有条件操作的逻辑电路。
- 程序计数器 [program counter, PC]** — 标识程序中当前语句的寄存器。PC 持续对程序存储器进行寻址，并始终包含下一要获取指令的地址。每次指令解码操作完成后，PC 内容将进行更新。
- 程序计数器 [PC] 字段** — 32 位 PC 寄存器中包含下一指令的地址的 29 位 并行处理器或 30 位主处理器计数器字段。
- 程序/数据等待状态寄存器 [program/data wait-state register , PDWSR]** — 指定程序、数据和输入/输出 (I/O) 空间的等待状态的存储器映射寄存器。PDWSR 的高位字节指定数据空间等待状态，而低位字节则指定程序空间等待状态。复位时，PDWSR = FFFF。
- 程序流程控制单元[program flow control unit]** — 管理从并行处理器指令高速缓存获取操作码的单元。
- 程序级优化[program-level optimization]** — 将所有源文件编译进一个中间文件的主动优化方式。因为编译器能够看到整个程序，所以执行多个优化时可以采用在文件级优化中很少应用的程序级优化。
- 程序存储器地址 [program memory address, PMA]** — 为存储在程序存储器中的乘法器操作数提供地址的寄存器。
- 程序存储器[program memory]** — 用于存储和执行程序的存储器区域。
- 程序读取总线 [program read bus, PRDB]** — 从程序存储器向 CPU 传输指令代码、立即操作数及表信息的内部总线。
- 程序选择引脚 [program select pin , PS]** — DSP 将 PS 置为有效，指示对外部程序存储器的存取。
- 程序空间等待状态 [program-space wait-state, PSWS] 位** — 等待状态发生器控制寄存器 (WSGR) 的位 0。PSWS 决定应用到片外程序存储器空间读取操作的等待状态数。
- 程序空间上部等待状态 [program-space upper wait-state, PSUWS] 位** — 上部程序空间等待状态位。等待状态生成器控制寄存器 (WSGR) 中的值，决定片外上部程序空间（地址：8000h–FFFFh）的读写等待状态数。'C209 上没有 PSUWS；相反，请参见程序空间等待状态位 (PSWS)。
- 可编程逻辑阵列 [programmable logic array , PLA]** — 由互连逻辑门组成的集成电路。
- 可编程只读存储器 [programmable read-only memory , PROM]** — 存储数字数据的大规模集成电路。该存储器可以通过紫外线灯擦除和重新编程，或者只可在工厂或现场编程一次。
- 编程 I/O [PIO] 模式** — 编程 I/O 模式。不使用直接存储器存取 (DMA) 的音频编解码模式；而是对 PIO 端口直接进行样本读写操作。
- 逻辑程序** — 函数中设置堆栈的代码部分。另请参见管道循环逻辑程序。

受保护模式 — 32 位扩展 DOS 模式。这些程序需要扩展存储器管理器，且只能在较大的处理器上运行。它们使用计算机上所有可用的 RAM（高达 64M 字节）。

下拉菜单 — 通过名称或使用鼠标从调试器显示屏顶部的菜单栏访问的命令菜单。

脉冲代码调制 [**pulse code modulation , PCM**] — 通过对声波进行取样，然后将每个样本转换为二进制数的一种数字化语音技术。

脉冲编码调制模式 [**PCM**] 位 — 启用/禁用缓冲串行端口 (BSP) 发送器的 1 位字段。此位存储在 BSP 控制扩展寄存器 (SPCE) 中。

脉冲频率调制（或模式） [**PFM**] — 为了调制输出信号，在不同频率时采用固定持续时间脉冲。

脉宽调制 [**pulse width modulation , PWM**] — 用来发送信息的占空比。

推入 [**push**] — 将字放入堆栈的操作。

## Q

**Qi** — 无线充电联盟 (WPC) 符号。

无声运行 [**quiet run**] — 抑制正常横幅和进程信息的运行方式。

量化错误 [**quantization error**] — 由于数字信号只能具有离散值，而模拟信号则可以具有信号动态范围内的任意值，从而导致在模数信号转换过程中发生的错误。

量级 [**quantum levels**] — 当信号只能具有一定离散值时，这些值被称为量级。

快速输出放电 [**quick output discharge, QOD**] — 将节点放电的电流路径。

无声非数 [**quiet not-a-number, QNaN**] — 无数字值的浮点数，某些特定例外情况下被用作发送给主处理器的信号。

品质因素 [**quality factor, Q**] — 指定频率下，电感器虚数电抗与实际电阻间的真实比率。

**R**

**R 分频器[R Divider]** — 用因子 R 将晶振基准频率（和相位）分频的分频器。

**射频前端 [radio frequency front end, RFFE]** — 射频系统中，天线和数字基带系统之间的全部组件。

**RAM 使能引脚 [RAM enable pin, RAMEN]** — 这个引脚启用或禁用片上单存取 RAM。

**ROM 模型** — 链接器链接 C 代码时使用的自动初始化模型。链接器在您使用 `-cr` 选项调用链接器时使用此模型。RAM 模型允许在加载时而不是在运行时初始化变量。

**RAM 覆盖 (OVLY) 位** — 决定是否可在数据存储空间中寻址片上单存取 RAM 的 1 位字段。复位时，OVLY = 0 此位将存储在处理器模式状态寄存器 (PMST) 中。

**随机存取存储器 [random-access memory , RAM]** — 可读写的存储器元素。

**随机存取存储器数模转换器 [random-access memory digital-to-analog converter, RAMDAC]** — 用于将数字 RGB（红-绿-蓝）信息转换为驱动显示器的模拟信号。

**光栅[raster]** — 组成电视画面或计算机显示的一系列扫描线。光栅线就是扫描线，即生成画面的电子波束对整个显示表面的单次扫描。

**光栅操作[raster-op]** — 将源数据和目标数据进行算术或逻辑组合的光栅操作，该操作在像素阵列从一个位置往另一个位置传输的过程中发生。

**原始数据** — 输出段中的可执行代码或初始化的数据。

**RCA 连接器[RCA connector]** — 将诸如立体声设备或复合视频显示器等音频和视频器件连接至计算机视频适配器的连接器。

**RCA 插孔[RCA jack]** — RCA 连接器的插座，通常位于音频或视频设备上。

**只读存储器 [read-only memory, ROM]** — 包含不能改变的永久数据的半导体存储元件。

**读取选择引脚 [ $\overline{RD}$ ]** — DSP 将  $\overline{RD}$  置为有效，以请求对外部程序、数据或者 I/O 空间的读取。 $\overline{RD}$  能与外部器件的输出启用引脚直接连接。

**读取/写入 [ $R/\overline{W}$ ] 引脚** — 这个存储器控制信号表示与外部器件通信时的传输方向。

**就绪** — 一种任务状态，表示任务正在执行或一旦获得处理器就能立即执行。

**就绪** — **外部器件就绪引脚**。用于从外部创建等待状态。此引脚驱动为低电平时，器件等待一个 CPU 周期，然后再测试 READY。READY 驱动为低电平后，器件不再继续处理，直到 READY 驱动为高电平。

**就绪队列** — 一个链接列表，包含所有处于 READY 状态的任务的描述符。

**实模式[real mode]** — 16 位本机 MS-DOS 模式。该模式将可用存储器限制为 640K 字节。调用 DOS 可能需要从受保护模式切换到实模式。TMS320C3x/C4x 代码生成工具不再支持 DOS 实模式工具。

**实时** — 计算物理过程发生的实际时间，在此期间，计算结果能与该物理过程进行交互。

**实时处理[real-time processing]** — 确保计算机系统的操作按照严格的时间间隔完成所需的机制。

**实时系统[real-time system]** — 该系统中，每项处理工作都在指定期限内完成。实时系统的特征更多地是由最坏情况下的延迟，而不是其最佳情况下的速度决定。其最坏情况延迟必须设定，以确保作业能够在特定时间内完成。

接收 **[receive , Rx]** — 无线电池供电系统的电源接收端。

接收缓冲器半接收 **[RH]** 位 — 表示接收缓冲区哪半部分已经被接收的 1 位缓冲串行端口 (BSP) 标志。复位时, RH = 0 此位存储在 BSP 控制扩展寄存器 (SPCE) 中。

接收时钟输入 **(CLKR)** 引脚 — 接收外部时钟信号, 以采用时钟信号形式将数据从串行数据接收 (DR) 引脚发送到同步串行端口接收移位寄存器 (RSR) 的引脚。

接收 (同步串行端口) **[DR]** 引脚 — 接收串行数据的同步串行端口引脚。由于每个位都在 DR 接收, 该位以串行的方式传输到接收移位寄存器 (RSR) 中。

接收 FIFO 缓冲器非空 **[receive FIFO buffer not empty , RFNE]** 位 — 同步串行端口控制寄存器 (SSPCR) 中的位, 指示同步串行端口的接收 FIFO 缓冲器中是否包含要读取的数据。

接收帧同步 **[ frame synchronization , FSR]** 引脚 — 接收启动同步串行端口接收过程的帧同步脉冲的输入引脚。

接收中断 (异步串行端口) — 在接收过程中由下列事件之一引发的中断 (TXRXINT): 异步数据发送和接收寄存器 (ADTR) 包含一个新字符; 发生溢出; 发生组帧错误; RX 引脚上检测到中断; 自动波特率检测逻辑在 ADTR 上检测到字符 A 或 a。

接收中断屏蔽位 **[receive interrupt mask bit , RIM]** — 异步串行端口控制寄存器 (ASPCR) 中的位, 用于启用或禁用异步串行端口的接收中断。

接收中断 (同步串行端口) **[RINT]** — 根据接收期间接收 FIFO 缓冲器中的字数生成的中断 (RINT)。触发条件 (缓冲器中所需的字数) 由同步串行端口控制寄存器 (SSPCR) 中的接收中断位 (FR1 和 FR0) 的值决定。

接收 **[RX]** 引脚 (异步串行端口) — 异步串行端口接收数据期间, 此引脚每次接收一个字符的一个位, 并发送给异步串行端口接收移位寄存器 (ARSR)。

接收 **[DR]** 引脚 (同步串行端口) — 接收串行数据的同步串行端口。由于每个位都在 DR 接收, 该位以串行的方式传输到接收移位寄存器 (RSR) 中。

接收就绪 **[receive ready ,RRDY]** 位 — 从 0 到 1 转换的 1 位标志, 指示数据接收移位寄存器 (RSR) 中的内容已复制到数据接收寄存器 (DRR) 中, 并且可以读取。接收中断 (RINT) 在转换时生成。RRDY 位可以在软件中轮询, 而不用使用串行端口中断。此位存储在串行端口控制寄存器 (SPC) 和 TDM 串行端口控制寄存器 (TSPC) 中。

接收寄存器 (异步串行端口) **[receive register (asynchronous serial port), ADTR]** — 异步串行端口中的寄存器, 写入数据进行发送, 并读取接收到的数据。另请参见 *异步串行端口接收移位寄存器*。

接收寄存器 (同步串行端口) **[receive register (synchronous serial port), SDTR]** — 一种 I/O 映射读/写寄存器, 用于向发送 FIFO 缓冲器发送数据, 从接收 FIFO 缓冲器提取数据。

接收复位 **[receive reset, RRST]** 位 — 串行端口控制寄存器中的位, 用于复位串行端口的接收器部分。

接收寄存器溢出指示符位 **[receiver register overrun indicator bit, OE]** — I/O 状态寄存器 (IOSR) 中的位, 指示异步串行端口的接收器是否发生了溢出; 即, 异步数据发送和接收寄存器 (ADTR) 中是否有未读字符被新字符覆盖。

接收器复位 **[RRST]** 位 — 将串行端口接收器复位的 1 位标志。复位时, RRST = 0 此位存储在串行端口控制寄存器 (SPC) 和 TDM 串行端口控制寄存器 (TSPC) 中。

**接收移位寄存器已满 [receive shift register full, RSRFULL] 位** — 表示串行端口接收器是否已经发生溢出的 1 位标志。此位存储在串行端口控制寄存器 (SPC) 中。

**接收移位寄存器 (同步串行端口) [RSR]** — 接收移位寄存器已满 [receive shift register full, RSRFULL] 位

**回收端口 [reclamation port]** — 消息被丢弃时, 消息缓冲器返回到的消息端口。然后, 包含此消息的缓冲器可发送另一条消息。另请参见 *端口 ID*。

**重新连接** — 断开连接后, 目标用来选择启动器以恢复处理的函数。

**精简指令集计算机 [reduced instruction set computer, RISC]** — 在指令集与相关解码机制方面, 比微编程复杂指令集计算机简单很多的计算机。其结果是, 在更小、更低价的芯片上, 实现了更高的指令吞吐量和更快的实时中断处理响应。

**冗余循环 [redundant loops]** — 同一循环的两个版本, 其中一个为软件管道化循环, 另一个则为未管道化循环。当工具无法保证总数足够大以管道化循环并达到最佳性能时, 就会产生冗余循环。

**可重入代码 [reentrant code]** — 可由多个任务同时执行的程序代码。可重入代码不包含任何任务专用的数据。

**基准寄生信号 [reference spurs]** — 由泄露电流和电荷泵不匹配导致的 PLL 输出上的有害频率尖峰, 此电荷泵 FM 调制压控振荡器 (VCO) 调节电压。(Undesired frequency spikes on the output of the PLL caused by leakage currents and mismatch of the charge pump that FM modulate the VCO tuning voltage.)

**刷新** — 恢复存储器件 (比如动态 RAM (DRAM) 或视频 RAM (VRAM)) 的电荷电容或恢复存储器内容的方法。

**刷新率 [refresh rate]** — 视频源在屏幕上重现显示的速度。

**寄存器 [register]** — 位于处理器或电子器件内的一小块高速存储器区域, 用来临时存储数据或指令。每个寄存器都有一个名称, 包含一些字节的信息, 并由程序引用。

**寄存器文件** — 一组寄存器。

**重定位 [relocation]** — 当某符号地址发生改变时, 连接器调整所有引用该符号的过程。另外也指连接器将每个输出段放入存储器的过程。

**远程进程调用 [remote procedure call, RPC]** — 处理器上执行的进程 (或函数) 调用, 而不是发出调用的过程。另请参见 *存根例程*。

**重复计数器 [RC] 寄存器** — CPU 寄存器文件中的寄存器, 用于指定当块重复执行完毕后, 代码块要重复的次数减一。

**重复模式** — 重复执行代码块的零开销方法。使用重复模式可在尽可能短的时间内执行代码的时间关键段。

**回复消息 [reply message]** — 服务器回复客户端服务请求的消息。

**回复端口 [reply port]** — 接收回复请求消息的消息端口。另请参见 *端口 ID*。

**请求消息 [request message]** — 客户端向服务器请求服务而发送的消息。

**保留 [reserved, R]** — 为未来小型计算机系统接口 (SCSI) 标准化预留的位、字节、字段或编码值的术语。

**复位 [reset]** — 通过将寄存器和控制位设置为预定值并发送在指定地址开始执行的信号, 使处理器进入已知状态的方法。

- 复位异步串行端口 [reset asynchronous serial port, URST] 位** — 异步串行端口控制寄存器 (ASPCR) 内将异步串行端口复位的位。
- 复位引脚 ( $\overline{RS}$ )** — 更准确的说，是这个引脚上的信号导致了复位。
- 复位向量[reset vector]** — 用于复位的中断向量。
- 分辨率带宽 [resolution bandwidth , RBW]** — 显示输入信号功率与频率之间关系的 RF 设备。这款设备的工作方式为，获取频率斜升函数，并将其与输入频率信号混合在一起。混频器的输出由一个带通滤波器滤波，此滤波器的带宽等于分辨率带宽。这款滤波器的带宽越窄，允许通过的噪声越少。
- 资源 ID** — 内核用来标识其已打开或已创建的端口、信标或任务的 32 位值。另请参见 *ID*。
- 资源表** — 固定大小的指针表，内核使用它来跟踪其已打开或已创建的内部资源。另请参见 *内核资源*。
- 恢复[resume]** — 使已挂起的任务处于就绪状况，以再次执行。
- 回扫[retrace]** — 当显示屏的扫描光束从一条水平线（垂直场）的结束位置行进到下一水平线（垂直场）的开始位置时，此扫描光束绘出的线。
- 返回地址[return address]** — CPU 从子例程或中断处理例程返回时要执行的指令的地址。
- 反向汇编[reverse assembly]** — 将存储器的内容从机器语言转换成汇编语言的过程（也称反汇编）。
- 反向电流阻断 [reverse current blocking, RCB]** — 阻断流经 FET 体二极管的电流。
- 反向电流保护 [reverse current protection, RCP]** — 在不损坏 FET 的情况下，允许一定量的反向电流流经 FET 的体二极管。
- 最右 1 [rmo]** — 返回最右 1 位位置的操作。主处理器中，这是一个实际指令，而在并行处理器中，却是一个位检测函数。
- 最右位改变 [rmbc]** — 并行处理器的操作，旨在返回值不同于位 0 的最右位的位置。如果源中的所有位相同，则设置溢出状态位。
- 纹波进位输出信号** — 计数器的输出信号，指示该计数器已达到其最大值。
- ROM 启用 [ROM enable, ROMEN]** — 决定是否启用片上 ROM 的外部引脚。
- ROM 模型** — 链接器链接 C 代码时使用的自动初始化模型。链接器在您使用 `-c` 选项调用链接器时使用此模型。在 ROM 模型中，链接器将数据表的 `.cinit` 段加载到存储器中，并在运行时初始化该变量。
- ROM 宽度** — 每个输出文件的宽度（以位为单位），或更具体的说，文件中单个数据值的宽度。ROM 宽度决定 TI 十六进制转换实用工具如何将数据分入输出文件。目标字映射到存储器字后，存储器字将分解为一个或多个输出文件。输出文件的数目由 ROM 宽度决定。
- 轮询调度[round-robin scheduling]** — 调度一组相同优先级任务的执行的方法。这些任务轮流使用处理器。每个任务轮流执行，所有任务的时间间隔基本相同。
- 路由端口[routing port]** — 多处理器系统中的本地端口，通过此端口，多媒体视频处理器 (MVP) 执行将消息发送给特定外部处理器节点。另请参见消息路由表。
- 行地址选通 [ $\overline{RAS}$ ]** — 驱动动态 RAM (DRAM) 和/或视频 RAM (VRAM) 的行地址选通输入的存储器接口信号。
- 运行地址[run address]** — 段运行的地址。

**运行时环境[[runtime environment](#)]** — 程序函数中的运行时参数。这些参数由存储器和寄存器约定、堆栈组织、函数调用约定及系统初始化定义。

**运行支持函数** — 标准的 ANSI 函数，执行的任务不属于 C 语言的一部分（比如存储器分配、字符串转换和搜索等）。

**运行支持库[[runtime-support library](#)]** — 包含运行支持函数来源的库文件。

**SAM 溢出事件[SAM overflow event]** — 在视频控制器 (VC) 中，由串行寄存器传输 (SRT) 控制器生成的串行存取存储器溢出事件决定何时执行循环，以处理拆分串行存取存储器 (SAM) 未激活的一半。

**采样率[sample rate]** — 音频编解码器采样音频数据的速率。通常以赫兹为单位（每秒采样数）。

**标量类型[scalar type]** — 一种 C 编程类型，其中变量为单变量，不是由其它多个变量组成。

**扫描链[scan chain]** — 由多个逻辑存储元素（独立位存储器件）组成的移位寄存器。在测试模式中，存储元素在移位寄存器中连接。在正常操作中，它们执行正常的系统功能。扫描路径用于将测试数据移入逻辑存储元素中，以便控制，并移出测试响应数据，以便观察。（也称扫描路径。）

**调度器[scheduler]** — 也称为任务调度器。负责决定就绪队列中任务执行顺序的内核部分。

**调度[scheduling]** — 操作系统中用于共享资源（比如处理器或存储器）的策略。

**暂存式 RAM[scratch-pad RAM]** — 双存取 RAM (DARAM) 的另外一个名称。

**脚本[script]** — 包含一组外壳命令的文件。

**滚动[scrolling]** — 上下左右移动窗口内容，以查看非原始显示内容的方法。

**SCSI** — 小型计算机系统接口。

**SCSI 地址** — 分配给小型计算机系统接口 (SCSI) 器件的唯一地址 (0-7)。

**SCSI 器件** — 连接到小型计算机系统接口 (SCSI) 总线的主机计算机、外设控制器或智能外设单元。

**SCSI ID** — 小型计算机系统接口 (SCSI) 地址的有效位表示（此位地址与数据总线的位数关联）。

**段[section]** — 代码或数据的浮动块，最终在存储器映射中占据与其它代码块相邻的空间。

**段标头[section header]** — 常用对象文件格式 (COFF) 对象文件的一部分，包含文件中关于某段的信息。每段都有其自己的标头。标头指向段的起始地址，包含该段的大小等信息。

**段程序计数器 [section program counter, SPC]** — 汇编器中的元素，可持续跟踪段中的当前位置；每个段都有其自己的 SPC。

**信标[semaphore]** — 多处理环境中，限制共享资源（例如，存储）访问的经典方法。信标是受保护的变量（或抽象数据类型），只有特定的测试或递增变量值的操作才能访问它。

**信标 ID** — 内核用来标识已打开信标的 32 位值。另请参见 *ID*。

**半普遍存在像素[semi-omnipresent pixel]** — 视频显示中，一次在屏幕上的两个位置同时出现的像素。

**信标表[semaphore table]** — 固定大小的指针表，内核使用它来跟踪已打开的信标。

**灵敏度[sensitivity]** — 对锁相环 (PLL) 芯片（来自压控振荡器 (VCO)）高频输入的功率限制。在这些限值出现时，计数器开始错误计算频率，并且分频出现错误。

**顺序模式[sequential mode]** — 一种主处理器 (MP) 浮点单元模式，在该模式下，MP 完成执行一个单浮点指令后，才开始执行下一个浮点指令。

**串行存取存储器 [serial access memory, SAM]** — 视频 RAM (VRAM) 中的一种存储阵列，可以通过串行寄存器传输周期进行存取。另请参见 *串行寄存器传输 [SRT] 控制器；视频随机存取存储器 (VRAM)*。

- 串行存取存储器递增 [serial access memory increment, SAMINC]** — 串行寄存器传输 (SRT) 控制器用于地址计算的 32 位控制掩码。此掩码在串行存取存储器掩码 (SAMMASK) 寄存器中的最高有效位 1 以上的位位置包含一个 1。
- 串行存取存储器掩码 [serial access memory mask, SAMMASK]** — 包含一连串 1 的串行寄存器传输 (SRT) 控制器寄存器。这些 1 的数量和位置取决于视频 RAM (VRAM) 串行存取存储器 (SAM) 的宽度和连接至 VRAM 的地址线路。
- 串行时钟 [serial clock, SCLK]** — 输入时钟信号，由串行寄存器传输 (SRT) 控制器用来跟踪视频 RAM (VRAM) 的接触点。
- 串行数据接收 [DR] 引脚** — 接收串行数据的同步串行端口引脚。由于每个位都在 DR 接收，该位以串行的方式传输到接收移位寄存器 (RSR) 中。
- 串行数据发送 [DX] 引脚** — 在该引脚上，数据从同步串行端口串行传输；从发送移位寄存器 (XSR) 按一次一位的速度接收数据字
- 串行端口 [serial port]** — 一种 DSP 外设，用于按顺序发送和接收数据样本，以和串行器件（比如模数和数模转换器和编解码器、微处理器和 DSP）进行通信。另请参见 *串行端口接口*。
- 串行端口接口 [serial-port interface]** — 一种片上全双工串行端口接口，可为带具有外部硬件的串行器件（比如编解码器和串行模数转换器）提供通信服务。串行端口的状态和控制由串行端口控制寄存器 (SPC) 中指定。
- 串行端口接收中断 [serial port receive interrupt, RINT] 位** — 1 位标志，指示数据接收移位寄存器 (RSR) 的内容已复制到数据接收寄存器 (DRR) 中。此位存储在中断标志寄存器 (IFR) 中。
- 串行端口发送中断 [XINT] 位** — 1 位标志，指示数据发送寄存器 (DXR) 的内容已复制到数据发送移位寄存器 (XSR) 中。此位存储在中断标志寄存器 (IFR) 中。
- 串行寄存器传输 [serial register transfer, SRT] 控制器** — 调度发送至传输控制器的请求，以将数据输入或者输出视频 RAM (VRAM) 帧存储器的硬件。
- 服务器** — 为客户端程序或任务提供服务的程序或任务。
- SETBRK 位** — 异步串行端口控制寄存器 (ASPCR) 中的位，当端口不发送数据时，在异步发送 (TX) 引脚上选择输出电平（高或低）。
- 共享存取模式 [shared-access mode, SAM]** — 此模式允许 DSP 和主机访问主机端口接口 (HPI) 存储器。此模式下，异步主机访问在内部被同步；如果发生冲突，主机将具有访问优先权，而 DSP 则等待一个周期。
- 共享存取模式 [shared-access mode, SMOD] 位** — 启用或禁用共用存取模式 (SAM) 的 1 位字段。此位存储在主机端口接口 (HPI) 控制寄存器 (HPIC) 中。另请参见 *共享访问模式 (SAM)* 和 *仅主机模式 (HOM)*。
- 共享 RAM** — 可由多个处理器共用的存储器。由并行处理器数据 RAM 和并行处理器参数 RAM 组成。
- 外壳程序 [shell program]** — 一个步骤就可以编辑、汇编和选择性链接的实用工具。通过编译器（包括解析器、优化器和代码发生器）、汇编器和链接器，外壳程序可以运行一个或者多个源模块。
- 移位器 [shifter]** — 字中可将位向左或向右移位的单元。另请参见 *P-SCALER*。
- 短浮点格式 [short-floating-point format]** — 浮点数的 16 位表示法，带 12 位尾数和 4 位指数。

- 短立即数[short immediate]** — 主处理器 (MP) 上, 由 MP 指令提供的 15 位有符号或无符号的整数, 作为 32 位 指令格式内的操作数之一。
- 短立即值[short-immediate value]** — 8、9 或 13 位常数, 作为使用立即寻址的指令的操作数。
- 短偏移[short offset]** — 半字或字传输的无符号 3 位立即指数; 字节传输的无符号 4 位立即指数。短偏移可以在寻址模式下用于指令中并行指定的其它操作。
- 短整数格式[short integer format]** — 用于整数数据的二进制补码, 16 位格式。
- 短无符号整数格式** — 用于整数数据的 16 位无符号格式。
- 副作用** — C 表达式的一种功能, 在表达式中使用赋值运算符来影响表达式中所使用其中一个要素的值。
- 信号** — 用于在信标处递增计数。另请参见 信标。
- 信号事件** — 因信号到达信标处而导致的事件。如果任务正在等待事件, 则该任务已计划开始执行。
- 符号位** — 一个值的最高有效位, 用来向 CPU 表示该值的符号 (正或负)。
- 符号扩展** — 用值的符号位填充该值未使用的最高有效位 (MSB)。
- 符号扩展** — 用符号位填充数字的高序位的操作。
- 符号扩展模式位 [sign-extension mode bit , SXM]** — 启用或禁用算术运算符扩展的 1 位字段。此位存储在状态寄存器 1 (ST1) 中。
- 信令非数 [signaling not-a-number, SNaN]** — 没有数值的浮点数, 但在某些特殊异常情况下可以用来向主处理器发送信号。
- 信噪比 [signal-to-noise ratio , SNR]** — 音频质量的度量标准。
- 模拟器[simulator]** — 一种开发工具, 可以模拟器件的运行, 以通过使用器件调试器来执行和调试应用程序。
- 单存取 RAM [single-access RAM , SARAM]** — 仅能在单个时钟周期内进行读写操作的存储器空间; 可在单个 CPU 周期内访问 (读取或写入) 一次的 RAM。
- 单端 [single ended, SE]** — 驱动一端接地的扬声器。
- 单精度浮点** — 32 位浮点数
- 单精度浮点格式** — 浮点数的 32 位表示法, 带 24 位尾数和 8 位指数。
- 单精度整数格式** — 用于整数数据的二进制补码 32 位格式
- 单精度无符号整数格式** — 用于整数数据的 32 位无符号格式
- 单步[single step]** — 使您可以查看每条语句效果的程序执行形式。程序逐条执行语句; 每条语句执行完后, 调试器暂停, 以更新数据显示窗口。
- 单线程[single threaded]** — 每次在一个处理器上只允许执行一个单线程或程序元素的编程方法。多任务系统刚好相反, 它允许在一个处理器上同时运行多个任务。
- 偏移** — 基于信号来源与器目的之间物理距离的多个时钟信号的时间差。开关延迟由逻辑门所导致。
- 小外形尺寸 (无引线) [small outline (no lead), SON]** — 无外露引线的塑料封装。

**史密斯圆图[smith chart]** — 显示器件阻抗随频率变化的图。

**soft 位** — 当在高级语言调试器中遇到断点时，与 **Free** 位配合使用来决定串行端口或定时器状态的 1 位字段。复位时，**Soft = 0**。

**软件开发系统 [software development system, SWDS]** — 为程序评估和开发提供低成本方法的 PC 兼容插件板。

**软件管道操作[software pipelining]** — C 优化器和汇编优化器使用的一种技术，用来从一个循环中调度指令，从而可以并行执行该循环中的多个迭代。

**软件堆栈[software stack]** — 使您可以使用 **PSHD** 和 **POPD** 指令将硬件堆栈扩展到数据存储器中的程序控制功能。堆栈可以从数据存储器直接存储和恢复，每次一个字。此功能对深度子例程嵌套，或防止堆栈溢出很有用。

**软件写入[software write]** — 向主处理器或并行处理器指令中指定的目标寄存器的写入。软件写入比硬件自动执行的写入（比如程序计数器递增或算术逻辑单元 (ALU) 状态设置）具有更高的优先级。

**源文件[source file]** — 包含经编译或汇编后形成对象文件的 C 代码或汇编语言代码的文件。

**频谱分析仪[spectrum analyzer]** — 显示输入信号功率与频率之间关系的 RF 设备。这款设备的工作方式为，获得频率斜升函数，并将其与输入频率信号混频。此混频器的输出经带通滤波器滤波，此滤波器的带宽等于分辨率带宽。这款滤波器的带宽越窄，其上允许通过的噪声越少。

**轮转[spinning]** — 对寄存器或存储器位置内的状态标志进行持续监控的一种技术，其中处理器不断轮询该标志，直到等待的状态发生改变。

**拆分模式[split mode]** — SRAM 内存组

直接存储器存取 (DMA) 协处理器的一种操作模式，使一个 DMA 通道能同时处理一个通信端口的接收和发送部分。

**拆分乘法[split multiply]** — 同时执行两个 8 位乘 8 位运算的并行处理器运算。并行处理器乘法器可执行两个无符号 8 位乘以无符号 8 位，或两个无符号 8 位乘以有符号 8 位的乘法运算。

**寄生信号衰减[spurious attenuation]** — 这是指环路滤波器减少基准寄生信号的程度。这可在闭环传输函数中看到。

**寄生信号增益 [spur gain, SG]** — 这是指在比较频率上进行评估的开环传输函数的幅度。这很好地表示出两个环路滤波器基准寄生信号的比较方式。

**SRAM 内存段[SRAM banks]** — 静态随机存取存储器组 这些包括参数和数据 RAM 以及指令和数据高速缓存。

**独立预处理器[standalone preprocessor]** — 能作为独立程序扩展宏、**#include** 文件和有条件编译的软件工具。它也可以执行包括指令解析在内的集成预处理。

**独立模拟器[standalone simulator]** — 可加载和运行可执行 **COFF .out** 文件的软件工具。当和 C I/O 库配合使用时，独立模拟器支持全部具有屏幕标准输出的 C I/O 函数。

**堆栈[stack]** — 保留用于为子例程和中断处理例程存储返回地址的存储器块。

**堆栈指针[stack pointer]** — 包含（指向）系统堆栈顶部地址的特殊用途 32 位寄存器。

**开始位[start bit]** — 异步串行端口发送或接收的每个 8 位数据值前必须要有一个开始位，即一个逻辑 0 脉冲。

**状态[state]** — 多媒体视频处理器 (MVP) 执行程序内核控制运行的主处理器 (MP) 驻留任务的状态。任务有以下四种状态：READY（就绪），WAITING（等待），SUSPENDED（挂起）和 WAITSUSPEND（等待挂起）。

**静态随机存取存储器 [static random-access memory, SRAM]** — 无需刷新的快速存储器，DRAM 就是如此。然而，其价格较 DRAM 贵，且达不到 DRAM 一样的高密度。

**静态变量[static variable]** — 仅仅局限在一个函数或程序内的一种变量。当退出函数或程序时，静态变量的值不会被丢弃；当重新进入函数或程序时，将恢复其之前的值。

**状态[status]** — 执行完一条命令后，将从目标发送一个字节信息至启动器。

**状态位[status bit]** — 状态字或者寄存器中的位，包含单条状态信息。

**停止位[stop bit]** — 异步串行端口发送或接收的每个 8 位数据值必须跟随一个或两个停止位，每一个均是一个逻辑 1 脉冲。所需停止位数量取决于异步串行端口控制寄存器 (ASPCR) 的停止位选择器 (STB)。

**存储类[storage class]** — 符号表的任何条目，指示应如何存取表中的符号。

**字符串表[string table]** — 存储名称长度超过 8 个字符的符号的表。长度为 8 个或以上字符的符号名称不能存储在符号表中；而是存储在字符串表中。指向字符串表中字符串位置的符号进入点的名称部分。

**结构[structure]** — 拥有统一名称的一个或者多个变量的集合。

**子块[sub-block]** — 高速缓存子块

高速缓存块的四个分区之一。高速缓存子块是在子块未命中时送入高速缓存的存储器单元。每个子块在该块的标志寄存器中有一个存在位（以及一个脏位，仅限主处理器 (MP) 数据高速缓存）。

**子块未命中[sub-block miss]** — 高速缓存未命中，其中存在所需要的块，但是没有需要的子块；结果导致管道停滞，直至所需子块进入高速缓存。

**子例程[subroutine]** — 由标准函数从应用程序调用的例程，器本身并不执行具体函数。另请参见 *远程过程调用 (RPC)*。

**子段[subsection]** — 最终将占据存储器映射中连续空间的浮动代码或数据块。子段为较大段中的较小段。子段让您可以更严格地控制存储器映射。

**替代符号表[substitution symbol table]** — 汇编器执行期间进行维护以跟踪与给定符号关联的文本的表。

**超级 VHS（垂直螺旋扫描）[S-VHS]** — 除了色度和亮度数据被当作能提供更高视频品质的分量外，S-VHS 与 VHS 视频录制标准相类似。

**管理者模式[supervisor mode]** — 主处理器 (MP) 对所有控制寄存器具有写访问权，并能对 MP 参数 RAM 进行写操作的模式。

**挂起[suspend]** — 无限期停止任务的继续执行，或直到另一任务指示内核恢复已挂起的任务。

**S-VHS 至 RCA 适配器线缆** — 线缆，用来连接视频源和软件开发板 (SDB)。

**交换文件[swap file]** — 虚拟存储器（辅存）在硬盘上分配的文件。

**符号** —

1. 由程序员定义的，表示特定数据项、指令、例程、值或地址的位置的字母、数字、符号或者其它标记。
2. 表示地址或值的字母数字字符串。

**符号调试** — 软件工具保留符号信息的能力，可供模拟器或仿真器等调试工具使用。

**符号表** — 常用对象文件格式 (COFF) 对象文件的一部分，包含有关该文件定义和使用的符号的信息。

**符号调试** — 软件工具保留符号信息的能力，可供模拟器或仿真器等调试工具使用。

**同步[*sync*]** — 指示显示器在何处放置图片的同步信号。另请参见 *水平同步* 和 *垂直同步*。

**同步串行端口接收中断[synchronous serial port receive interrupt]** — 中断屏蔽寄存器中的位，与同步串行端口的接收中断绑定。该位由串行端口内部生成。

**语法** — 语言的语法和结构规则。所有较高级编程语言都具有正式语法。

**系统外壳[*system shell*]** — 与 SYSTEM 命令一起执行的方法，通过此方法，调试器可以清空调试器显示器并临时退出至 DOS 提示符。这使您可以输入 DOS 命令或允许调试器显示执行 DOS 命令后的信息。

13 T

**T31 比率** — 三阶环路滤波器极的比率。如果比率为 0，那么这实际上是一个二阶滤波器。如果比率为 1，那么这个值是产生最低基准寄生信号的参数值。

**T41 比率** — 四阶滤波器中，极 T4 与极 T1 的比率。如果这个比率为 0，那么此环路滤波器为三阶或更低阶滤波器。

**T43 比率** — 极 T4 与极 T3 的比率。根据经验，选择的这个值不大于 T31 比率。

**标签** —

1. 可分配给结构、联合或枚举的可选类型名称。
2. 保存高速缓存块地址的寄存器。

**接触点[tap point]** — 数据被移入或从视频 RAM (VRAM) 串行 I/O 端口输出的点地址。串行寄存器传输 (SRT) 控制器使用串行时钟 (SCLK) 输入跟踪 VRAM 接触点。

**目标** — 执行启动器请求的小型计算机系统接口 (SCSI) 器件（通常为控制器）。

**目标存储器[target memory]** — 器件中加载可执行对象代码的物理存储器。

**目标系统[target system]** — 执行开发目标代码的系统。

**任务** — 多任务系统中和其它程序元素同时执行的程序元素。

**任务自变量[task argument]** — 内核开始执行任务时，作为自变量传递给该任务的指针值。

**任务描述符[task descriptor]** — 指定任务状态、优先级、函数指针、自变量和事件标志的数据结构。每项任务在创建时都会分配一个任务描述符。

**任务错误[task error]** — 任务中调用内核函数时发生的错误。与影响整个系统的系统错误相反，除发生错误的任务外，任务错误对其它任务并没有直接的影响。

**任务函数[task function]** — 以标准 C 函数形式编写的任务程序代码。执行程序内核中的任务调度器通过标准的 C 任务函数调用开始执行该任务。

**任务 ID** — 内核用来标识已创建的任务的 32 位值。

**任务中断[task interrupt]** — 多媒体视频处理器 (MP) 硬件机制，通过此机制，主处理器 (MP) cmdnd 指令能够在中断被启用时将一个任务中断发送给一个或多个并行处理器。MP 使用此机制来在多个并行处理器上创建任务。

**任务优先级[task priority]** — 分配给任务以标识其相对紧急程度的数字。多媒体视频处理器 (MVP) 执行程序内核使您可以给任务分配 0-31 之间的优先级数字。数字越大，表示优先级越高（越紧急）。

**任务调度器[task scheduler]** — 负责决定就绪队列中任务执行顺序的内核部分。

**任务状态[task state]** — 多媒体视频处理器 (MVP) 执行程序内核控制中运行的主处理器 (MP) 驻留任务的状态。任务有以下四种状态: READY (就绪), WAITING (等待), SUSPENDED (挂起) 和 WAITSUSPEND (等待挂起)。

**任务表** — 内核用来跟踪其已创建的任务的固定大小指针表。

**TDM 地址 [TDM address, TADD]** — 确定四线制串行总线上哪个器件应读取 TDM 数据 (TDAT) 线中的数据的单一、双向地址线。

**TDM 时钟 [TDM clock, TCLK]** — 用于时分复用 (TDM) 操作的单一、双向时钟线。TDM 接收时钟 (TCLKR) 和 TDM 发送时钟 (TCLKX) 引脚通过外部连接来构成 TCLK 线。

**TDM 数据 [TDM data, TDAT]** — 发送所有时分复用 (TDM) 数据的单一、双向线。TDM 串行数据接收 (TDR) 和 TDM 串行数据发送 (TDX) 引脚通过外部连接来构成 TDAT 线。

**TDM 接收中断 (TDM receive interrupt, TRNT) 位** — 指示时分复用 (TDM) 数据接收移位寄存器 (TRSR) 中的内容已复制到 TDM 数据接收寄存器 (TRCV) 中的 1 位标志。此位存储在中断标志寄存器 (IFR) 中。

**TDM 发送中断 (TDM transmit interrupt, TXNT) 位** — 指示时分复用 (TDM) 数据发送寄存器 (TDXR) 中的内容已复制到数据发送移位寄存器 (XSR) 中的 1 位标志。此位存储在中断标志寄存器 (IFR) 中。

**温度补偿晶振振荡器 [temperature-compensated crystal oscillator, TCXO]** — 用温度补偿来提高频率准确度的晶振。

**测试存取端口 [test access port, TAP]** — 由包括在器件中的 IEEE 标准 1149.1–1990 定义的, 以执行边界扫描函数, 并提供 DSP 和仿真器之间通信的标准通信端口。

**测试和控制 (TC) 标志位** — 存储算术逻辑单元 (ALU) 或并行逻辑单元 (PLU) 测试位操作的结果的 1 位标志。TC 位受 APL, BIT, BITT, CMPR, CPL, LST #1, NORM, OPL 和 XPL 指令影响。TC 位的状态影响有条件分支、调用和返回指令的执行。此位存储在状态寄存器 1 (ST1) 中。

**.text 段** — 缺省常用对象文件格式 (COFF) 的段之一。 .text 段是包含可执行代码的初始化段。 .text 指令将代码汇编到 .text 段中。

**执行线程[thread of execution]** — 可在多任务系统中调度的执行单元。该术语特指程序元素的逐步执行; 不包括其它属性, 比如分配给任务或进程的系统资源。

**紧密耦合多处理器[tightly-coupled multiprocessor]** — 相对于处理器在松散型耦合多处理器系统中通过网络通信, 多处理器系统中的处理器则通过共享存储器互相通信。

- 时间分片 [time slicing]** — 允许下一任务运行前，分配给每项任务最大执行时间（称为时间片或量子）的一种多任务方法。
- 时分复用 [time-division multiplexing, TDM]** — 数字音频的通信方法，此方法可承载很多通道。
- 时分复用 [TDM]** — 多个器件共享单条串行总线，并轮流在该总线上通信的过程。时隙（通道）的总数取决于已连接器件的数目。在一个时隙内，给定器件可以与总线上的任意器件组合进行通信。
- 时分复用 (TDM) 位** — 启用或禁用 TDM 串行端口的 1 位字段。此位存储在 TDM 串行端口控制寄存器 (TSPC) 中。
- 定时器 [timer]** — 用于产生脉冲或事件计时的可编程外设。
- 定时器分频寄存器 [timer divide-down register, TDDR] 位** — 为片上定时器指定定时器分频比（周期）的字段。当定时器预分频器计数器 (PSC) 递减到 0 以下时，PSC 将加载 TDDR 的内容。复位时，TDDR = 0000。这些位存储在定时器控制寄存器 (TCR) 中。
- 定时器中断 [timer interrupt, TINT]** — 主计数器（TIM 寄存器）递减为 0 后，定时器在下一个 CLKOUT1 周期内产生的中断。
- 定时器中断 [TINT] 位** — 指示定时器计数器寄存器 (TIM) 已递减到 0 以下的 1 位标志。此位存储在中断标志寄存器 (IFR) 中。
- 定时器输出 [timer output, TOUT] 引脚** — 根据片上定时器的速率提供到输出信号的访问。主计数器（TIM 寄存器）递减为 0 后的下一个 CLKOUT1 周期内，一个信号被发送至 TOUT。
- 定时器预分频计数器 [timer prescaler counter, PSC] 位** — 为片上定时器指定计数的字段。当 PSC 递减到 0 以下或定时器复位时，PSC 将加载定时器分频寄存器 (TDDR) 的内容，并且定时器计数器寄存器 (TIM) 将递减。这些位存储在定时器控制寄存器 (TCR) 中。
- 定时器重载 [timer reload, TRB] 位** — 将片上定时器复位的 1 位标志。当 TRB = 1 时，定时器计数器寄存器 (TIM) 将加载定时器周期寄存器 (PRD) 内的值，并且定时器预分频器计数器 (PSC) 将加载定时器分频寄存器 (TDDR) 位的值。此位存储在定时器控制寄存器 (TCR) 中。
- 定时器停止状态 [timer stop status, TSS] 位** — 停止并重新启动片上定时器的 1 位标志。复位时，TSS = 0，定时器立即开始定时。此位存储在定时器控制寄存器 (TCR) 中。
- 栈顶 [top of stack, TOS]** — 8 层后进、先出硬件堆栈的顶层。
- 总谐波失真和噪声 [total harmonic distortion and noise, THD+N]** — 音频质量的度量值。
- 推挽式输出 [totem-pole output]** — 同时积极驱动高和低逻辑电平的输出电路。

**传输控制器 [transfer controller, TC]** — 多媒体视频处理器 (MVP) 的片上直接存储器存取 (DMA) 控制器，用于处理高速缓存以及在 MVP 的每个处理器及其外部存储器之间传输一维、二维和三维数据块。

**传输控制 [transmission control, TCOMP] 位** — 同步串行端口控制寄存器中 (SSPCR) 中的位；指示同步串行端口的发送 FIFO 缓冲器中的所有数据均已发送。

**发送 [transmit, Tx]** — 无线电池充电系统的电源发射端。

**发送缓冲器半发送 [transmit buffer half transmitted, XH] 位** — 表示发送缓冲器的哪半部分被发送的 1 位标志。XINT 中断发生时（中断程序或 IFR 轮询），可以读取 XH 位。复位时，XH = 0。此位存储在 BSP 控制扩展寄存器 (SPCE) 中。

**发送时钟 I/O [CLKX] 引脚** — 为同步串行端口发送移位寄存器到 DX 引脚间的数据传输计时的引脚。如果串行端口配置为可接收外部时钟，则该引脚可接收时钟信号。如果端口配置为可生成内部时钟，则此引脚发送时钟信号。

**发送为空指示符 [transmit empty indicator, TEMT] 位** — I/O 状态寄存器 (IOSR) 中的位，表示异步数据发送和接收寄存器 (ADTR) 和异步串行端口发送移位寄存器 (AXSR) 为满或为空。

**发送帧同步 [transmit frame synchronization, FSX] 引脚** — 这个 I/O 引脚接受并生成一个启动同步串行端口传输过程的帧同步脉冲。如果端口配置为可接收外部帧同步脉冲，则 FSX 引脚接收该脉冲。如果端口配置为可生成内部帧同步脉冲，则 FSX 引脚发送该脉冲。

**发送中断屏蔽 [transmit interrupt mask, TIM] 位** — 异步串行端口控制寄存器 (ASPCR) 中的位，可启用或禁用异步串行端口的发送中断。

**发送中断串行端口 [transmit interrupt serial port]** — 传输时，发送寄存器为空时产生的中断。此情况表示发送寄存器已准备好接收新的发送字符。

**发送中断 [XINT] 同步串行端口** — 传输期间，根据发送 FIFO 缓冲器中字的数量生成的中断。触发条件（缓冲器中所需字数）由同步串行端口控制寄存器 (SSPCR) 的发送中断位 (FT1 和 FT0) 的值决定。

**发送模式 [transmit mode, TXM] 位** —

1. 指定帧同步发送 (FSX) 脉冲源的 1 位字段。复位时，TXM = 0 此位存储在串行端口控制寄存器 (SPC) 和 TDM 串行端口控制寄存器 (TSPC) 中。
2. 同步串行端口控制寄存器 (SSPCR) 中的位 3；决定帧同步源信号为外部信号还是内部信号。

**发送就绪 [transmit ready, XRDY] 位** — 用 0 到 1 转换来表示数据发送寄存器 (DXR) 的内容已复制到数据发送移位寄存器 (XSR) 中，且数据已准备好与新数据字一同加载的 1 位标志。发送中断 (XINT) 是在转换中生成的。XRDY 位可以在软件中轮询，而不使用串行端口中断。此位存储在串行端口控制寄存器 (SPC) 和 TDM 串行端口控制寄存器 (TSPC) 中。

**发送和接收中断 [transmit and receive interrupt]** — 用于对来自串行端口的  $\Delta$  中断、接收中断或发送中断作出反应的 CPU 中断。另请参见  $\Delta$  中断。

- 发送寄存器为空指示符 [transmit register empty indicator, THRE] 位** — 位于 I/O 状态寄存器 (IOSR) 中的位，指示异步数据发送和接收寄存器 (ADTR) 中的内容已传输至异步串行端口发送移位寄存器 (AXSR)。
- 发送复位 [transmit reset, XRST] 位** — 同步串行端口控制寄存器 (SSPCR) 中的位，用于复位同步串行端口的发送器部分。
- 发送移位寄存器为空 [XSREMPY] 位** — 表示串行端口发送器是否已经出现下溢的 1 位标志。此位存储在串行端口控制寄存器 (SPC) 中。
- 发送器复位 [XRST] 位** — 将串行端口发送器复位的 1 位标志。复位时， $\overline{XRST} = 0$ 。此位存储在串行端口控制寄存器 (SPC) 和时分复用 (TDM) 串行端口控制寄存器 (TSPC) 中。
- 透明度 [transparency]** — 将源像素呈现为不可见的像素属性，从而使部分目的地阵列可以透过部分源阵列显示出来。
- 源操作的透明度 [transparency on source operation]** — 传输控制器 (TC) 进行的传输，其中源数据逐字节与一个透明度值进行比较；这些比较根据透明度数据大小进行分组。如果比较的字节位于组匹配中，则 TC 禁用相应字节选通，以防止对该组内的任何字节进行写入操作。
- 陷阱 [trap]** — 下一条指令开始执行前，强迫程序中断的当前执行指令造成的特殊情况。在处理器对陷阱进行处理后，一般会在造成陷阱的指令后随即的指令中恢复执行中断的程序。
- 陷阱向量表 [trap vector table, TVT]** — 每个地址对应一个中断的地址排序列表；当执行陷阱后，处理器对存储在陷阱向量表中相应位置的地址执行分支。
- 陷阱向量表指针 [trap vector table pointer, TVTP]** — CPU 扩展寄存器文件中的寄存器，包含陷阱向量表的起始地址。
- 涓流刷新周期 (trickle refresh cycles)** — 低优先级刷新周期。只有在总线空闲时，这些刷新周期才会发生。
- 三联符序列 [tri-graph sequence]** — 由 ISO 646-1983 不变量代码集定义的具有含义的三字符序列。这些字符不能在 C 字符集中表现出来，但可以扩展为一个字符。例如，三联符 '??' 扩展为 '^'。
- 循环次数 [trip count]** — 循环结束前执行的次数。
- 三重行 (triple)** — 表中由三列组成的一行。例如，RGB 三重行包含定义某种颜色的红、绿和蓝值。

**14 U**

**未配置的存储器** — 未定义为存储器映射的一部分，且无法加载代码或数据的存储器。

**欠压闭锁 [under-voltage lockout, UVLO]** — 防止器件运行的电压。

**统一模式 [unified mode]** — 直接存储器存取 (DMA) 协处理器的一种操作模式。该模式主要用于存储器到存储器之间的传输。这是 DMA 通道的缺省操作模式。另请参见 *拆分模式*。

**未初始化段 [uninitialized section]** — 在存储器映射中保留空间但不包含实际内容的常用对象文件格式 (COFF) 段。这些段由 .bss 和 .usect 指令创建。

**联合 [union]** — 可以保存不同类型和大小的对象的变量。

**UNION** — 能使链接器将同一地址分配给多个段的 SECTIONS 指令选项。

**通用异步接收器和传输器 [universal asynchronous receiver and transmitter, UART]** — 异步串行端口的另外一个名称。

**无符号值 [unsigned value]** — 无论其实际符号是什么，都当作正值的值。

**紧急刷新周期 [urgent refresh cycles]** — 当积压的刷新请求超过 16 个时出现的高优先级刷新周期。同时执行四个刷新周期，而余下的刷新请求需等待更高优先级周期完成后再执行。

**用户模式 [user mode]** — 在该模式下，主处理器 (MP) 无法写入编号低于 0x4000 的控制寄存器且无法写入 MP 参数 RAM。

## 15 V

**变容二极管 [varactor diode]** — 这是 VCO 内被反向偏置的二极管。当 VCO 的调节电压变化时，这个二极管的结电容发生变化，反过来改变 VCO 电压。

**变量 [variable]** — 表示可假设为任何一组值的数量的符号。

**变量插入码引导传输 [variable-patch guided transfer]** — 一类引导传输，所有的插入码大小信息均在引导表中指定，而不是在数据包传输参数中指定，从而允许每个传输中的每个插入码具有不同的大小。另请参见  $\Delta$  引导传输、偏移引导传输和引导传输。

**压控振荡器增益 [voltage-controlled oscillator (VCO) gain] [ $K_{VCO}$ ]** — VCO 的增益单位为 MHz/V。

**向量点积 [vector dot product]** — 表示来自两个不同向量 **a** 和 **b** 的单独元素乘积的数学术语。

**向量指令 [vector instruction]** — 可以允许并行执行浮点操作和加载或存储指令的操作。

**供应商唯一性代码 [vendor unique, VU]** — 可以为每个供应商或器件单独定义的位值、字段值或编码值。

**垂直消隐 [vertical blanking,  $\overline{VBLNK}$ ]** — 每帧出现一次（隔行扫描系统则每场出现一次），且具有定义为整数条线的脉冲宽度（隔行扫描系统为半线）的双向垂直定时信号。垂直回扫时， $\overline{VBLNK}$  用于禁用像素捕获和显示。

**垂直同步 [vertical synchronization,  $\overline{VSYNC}$ ]** — 每帧出现一次，具有定义为整数条线的脉冲宽度（隔行模式为半线）的双向垂直定时信号。

**VGA 导通线缆 [VGA pass-through cable]** — 德州仪器 (TI) 提供的线缆，TMS320C8x 软件开发板 (SDB) 工具包或套件的一部分。该线缆用来连接标准视频图形阵列 (VGA) 图形板的输出端和 SDB 的 VGA 导通连接器。

**视频控制器 [video controller]** — 负责视频接口的多媒体视频处理器 (MVP) 部分。

**视频数字转换器 [video digitizer]** — 一种能将模拟视频信号转换为数字表示的器件。

**视频图形阵列 [video graphics array, VGA]** — 视频卡的行业标准。

**视频接口调色板 [video interface palette, VIP]** — 用于将数字 RGB（红-绿-蓝）信息转换为驱动显示器的模拟信号。

**视频随机存取存储器 [video random access memory, VRAM]** — 微处理器存储器地址空间的保留部分，用于在视频数据发送至显示器之前临时存储这些数据。使视频电路能逐位串行存取存储器的动态读取存取存储器类型。VRAM 具有针对处理器和视频电路的单独引脚。它用在高速视频应用程序中，可轻松连接到视频显示器上。

**虚拟存储器 [virtual memory]** — 程序使用比计算机实际提供的 RAM 更多的存储器的能力。这是通过使用磁盘上的交换文件增大 RAM 来实现。RAM 不足时，部分程序会在需要一个磁盘文件之前将此文件换出。交换文件与可用 RAM 的组合就是虚拟内存。

**压控振荡器 [voltage-controlled oscillator, VCO]** — 输出频率取决于输入（控制）电压的器件。

**16 W**

晶圆芯片级封装 [**wafer chip-scale package, WCSP**] — 没有任何传统封装的器件；具有焊球的芯片本身。

正在等待 [**waiting**] — 任务自动阻断，不再继续执行，直到所等待的事件发生的状态。内核允许一项任务等待消息到达端口，或信号到达信标。

等待队列 [**wait queue**] — 在端口等待消息或在信标等待信号的任务队列。

等待状态 [**wait state**] — 对外部存储器进行读写操作时，CPU 必须等待外部程序、数据或 I/O 存储器作出响应的一段时间。CPU 针对每个等待状态多等待一个周期。

等待状态发生器 [**wait-state generator**] — 为指定片外存储器空间（程序、数据或 I/O）生成有限数量等待状态的片上外设。等待状态在等待状态发生器控制寄存器 (**WSGR**) 中设置。

预热启动 [**warm boot**] — 处理器将控制权转给之前已加载的程序的应用起点地址时使用的方法。

观察窗口 [**WATCH window**] — 显示选定表达式、符号、地址和寄存器值的窗口。

精确定义的表达式 [**well-defined expression**] — 只包含预先定义的符号或汇编时间常数的表达式。

窗口 — 显示器上定义的矩形虚拟空间区域。

字 — 视为实体的字符或位字符串。

写入使能 [**WE**] 引脚 — DSP 将 **WE** 置为有效，以请求对外部程序、数据或 I/O 空间进行写入操作。

**17 X**

**XA0-XA13** 地址引脚 — 数据/程序存储器或 I/O 设备的外部地址引脚。这些引脚位于扩展总线上。另请参见 **A0-An**。

**XD0-XD31** 数据总线引脚 — 外部数据总线引脚，用于在处理器和外部数据或程序存储器或 I/O 器件之间传输数据。另请参见 **D0-D31**。

**18 Z**

零填充 [**zero fill**] — 向 32 位字段加载 16 位数时，使用零来填充低或高序位的方法。

## 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间接权利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, 但他们将独立负责满足与其产品及其应用中使用 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, 并且由客户独立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 © 2014, 德州仪器半导体技术(上海)有限公司

## 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 © 2014, 德州仪器半导体技术(上海)有限公司