



Ashwani Goel

Sitara MPU

**摘要**

本文档用作 MCU+SDK 9.2.1 或更高版本 PCIe EP 驱动器程序扩展的应用报告，适用于德州仪器 (TI) 基于 ARM 的 SOC AM64x 和 AM243x。

**内容**

<b>1 缩写</b> .....	2
<b>2 引言</b> .....	2
2.1 外设组件快速互连.....	2
2.2 AM64x 和 AM243x 上的 PCIe 特性.....	5
<b>3 x86 作为 RC , AM64x 作为 EP</b> .....	6
3.1 硬件环境.....	6
3.2 软件环境.....	7
<b>4 测试设置</b> .....	9
4.1 适用于 LINUX 和 WIN 的通用设置.....	9
4.2 Linux 驱动程序(VFIO).....	10
4.3 测试应用程序使用情况.....	10
4.4 LINUX PC 的设置步骤.....	12
4.5 MSI 示例.....	14
4.6 Windows PC 的设置步骤.....	15
<b>5 PCIe 测试规范</b> .....	19
5.1 标识和配置功能.....	19
5.2 基准时钟功能.....	22
5.3 进站 ATU 和 BAR 功能.....	25
5.4 出站 ATU 功能.....	30
5.5 MSI 功能.....	31
5.6 下行中断功能.....	32
5.7 器件电源管理状态功能.....	33
5.8 功能级复位机制.....	35
5.9 传统中断机制.....	35
5.10 MSI-X 功能.....	36
5.11 热复位机制.....	36
<b>6 Windows 示例驱动程序验证</b> .....	38
<b>7 参考资料</b> .....	40

**商标**

所有商标均为其各自所有者的财产。

## 1 缩写

<b>PCIe</b>	外设组件快速互连
<b>PCI-SIG</b>	PCI 特别兴趣小组
<b>EP</b>	端点
<b>RC</b>	根复合体
<b>SSC</b>	展频
<b>BIOS</b>	基本输入输出软件
<b>CCS</b>	Code Composer Studio
<b>TI</b>	德州仪器 (TI)
<b>BAR</b>	基址寄存器
<b>MSI</b>	报文信号中断
<b>MSI-X</b>	报文信号中断 X
<b>SBL</b>	辅助引导加载程序
<b>VFIO</b>	虚拟功能 I/O
<b>IOMMU</b>	I/O 存储器管理单元
<b>ATU</b>	地址转换单元
<b>FLR</b>	功能级复位

## 2 引言

### 2.1 外设组件快速互连

外设组件快速互连 (PCIe) 是 2003 年推出的一种主板扩展总线标准，用于实现中央处理单元 (CPU) 与外设组件之间的高速串行通信。如今，PCIe 是主要的主板扩展总线标准，也是许多其他板载应用常用的通信方法。PCIe 通常用于图形处理单元 (GPU) 和固态硬盘 (SSD)，以便通过 CPU 发送和接收数据。

- 德州仪器 (TI) TIPL 视频：[什么是 PCIe？](#)

#### 2.1.1 PCIe 通信的元件

PCIe 通信包含三个主要元件：根复合体、中继器和 PCIe 端点。PCIe 通信是分层的，因此有一个单一来源，即根复合体，所有数据都通过该单一来源进行传递。数据从多个 PCIe 端点传输到根复合体，反之亦然。

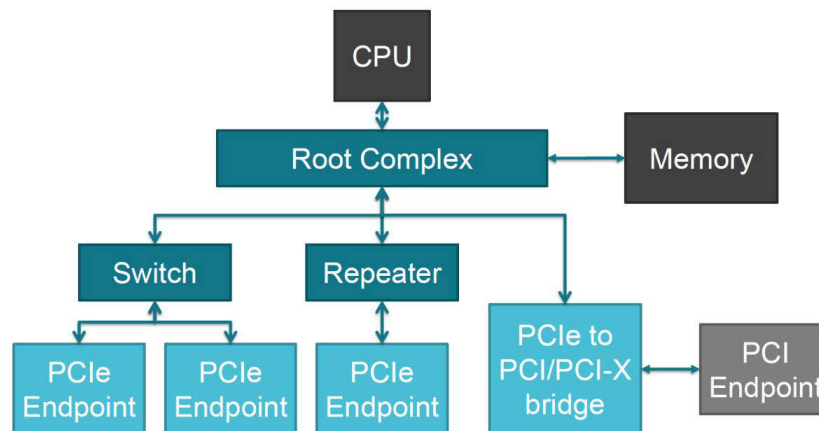


图 2-1. PCIe 拓扑

### 2.1.1.1 根复合体

根复合体是系统 CPU、存储器和其他 PCIe 接口之间的接口。根复合体要么直接集成到 CPU 中，要么作为分立式元件在 CPU 外部。此接口还用作单一源，来自各个 PCIe 端点的所有数据均通过。图 2-1 将根复合体显示为连接 CPU、存储器和 PCIe 元件的深蓝色框，并被称为根复合体。

有关更多详细信息，请参阅 [AM64x 作为 RC](#)

### 2.1.1.2 中继器

中继器是一种信号调节设备，可确保进出根复合体和 PCIe 端点的信号良好。中继器可分为两类：重定时器 and 转接驱动器。这两种常见的 PCIe 元件均用于保持高速链路的信号质量并补偿布线上的信号质量损失。图 2-1 将中继器显示为连接根复合体和 PCIe 端点的深蓝色框，称为中继器。

### 2.1.1.3 端点

端点是 PCIe 终端元件的通用术语。这可以表示许多不同类型的 PCIe 设备，例如 M.2 固态硬盘 (SSD) 或图形处理单元 (GPU)。端点可以是 PCIe 元件或带有 PCIe 到 PCI/PCI-X 桥接器的 PCI 元件。图 2-1 将 PCI 端点显示为连接到桥接器、交换机或中继器的浅蓝色框和灰色框，称为 PCIe 端点或 PCI 端点。

有关更多详细信息，请参阅 [AM64x 作为 EP](#)

## 2.1.2 信号

PCIe 通信的每个元件（转接驱动器除外）都具有以下控制信号：PERST、WAKE、CLKREQ、和 REFCLK。这些信号用于生成高速信号并与其他 PCIe 设备进行通信。图 2-2 展示了 PCIe 设备及控制信号的示意图。该图显示，除 REFCLK 外的所有控制信号均为低电平有效信号。

在图 2-2 中，中继器称为重定时器

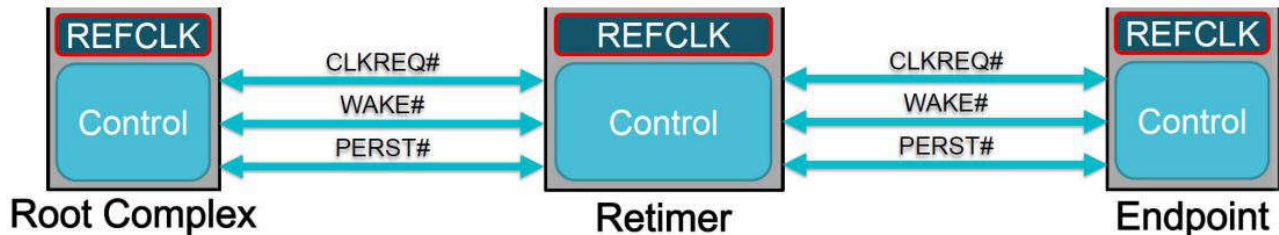


图 2-2. PCIe 信号

### 2.1.2.1 PERST

PERST 被称为基础复位。在系统中的所有电源轨和基准时钟稳定之前，PERST 可以保持低电平。该信号从低电平转换为高电平通常表示链路初始化开始。在图 2-2 中，PERST 称为 PERST#。

### 2.1.2.2 WAKE 和 CLKREQ

WAKE 和 CLKREQ 信号都用于在低功耗状态之间转换。WAKE 信号是一个低电平有效信号，用于在处于低功耗状态时将 PCIe 接口返回到工作状态。CLKREQ 信号也是低电平有效信号，用于请求基准时钟。在图 2-2 中，它们分别称为 WAKE# 和 CLKREQ#。

### 2.1.2.3 REFCLK

REFCLK (基准时钟信号) 是 PCIe 设备开始数据传输的先决条件。PCIe 器件使用该 100MHz 基准时钟信号在链路中生成高速 PCIe 数据，并由链路中的 PCIe 器件共享。在图 2-2 中，REFCLK 称为 REFCLK。

## 2.1.3 PCIe 常见用途

- 计算机硬件
  - 显卡  
PCIe 是显卡的常用主板接口。PCIe 允许 GPU 与系统其余部分之间进行高速通信。
  - 声卡  
声卡使用 PCIe 插槽进行音频处理和输出。

- **存储设备**  
SSD (固态硬盘) 通过 PCIe 连接, 可实现快速数据传输。
- **网络接口卡 (NIC)**  
PCIe 可实现高速网络连接。
- **工业系统**
  - 在工业自动化和控制系统中, PCIe 用于传感器、传动器和控制器之间的高速通信。
  - 工业 PC 的扩展卡和外设通常依赖 PCIe。
- **数据中心**
  - 数据中心中的服务器和存储系统使用 PCIe 连接存储设备、网络适配器和加速器 (如 GPU 或 FPGA)。
  - PCIe 提供低延迟通信, 这对于数据中心工作负载至关重要。
- **汽车技术**
  - **信息娱乐系统**  
PCIe 接口用于连接多媒体元件, 例如显示屏、音频系统和导航单元。
  - **高级驾驶辅助系统 (ADAS)**  
PCIe 连接传感器、摄像头和处理单元, 以进行实时数据处理。
- **笔记本电脑和微型 PC**
  - PCIe 用于连接内置外设和插卡。
  - **Mini PCIe** 使用与常规 PCIe 相同的拓扑和规格, 而且具有电气兼容性。
  - 现在通用的 M.2 SSD 接口也使用 PCIe 拓扑。

### 2.1.4 PCIe 聚合吞吐量

PCI Express 互连称为链路, 用于连接两个器件。链路由每个方向的 1、2、4、8、12、16 或 32 个信号组成 (请注意, 系统使用全差分信号, 每个信号实际上需要两根导线)。这些信号被称为通道。设计人员根据给定链路上所需的目标性能基准确定要实施的通道数。在命名法中, 链路的宽度用一个数字的前面加上 **x** 显示, 其中 **x** 的发音为 *by*, 因此比如链路在每个方向上有 4 个信号, 称为 *by four* 链路。

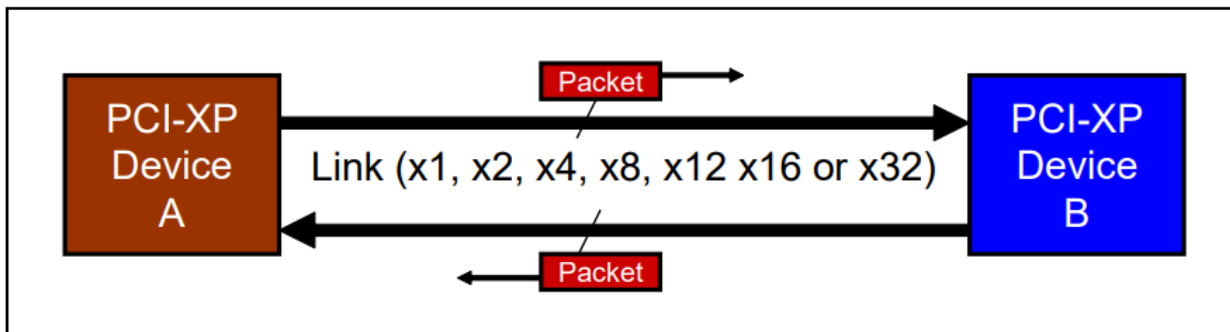


图 2-3. PCIe Express 链路

表 2-1 展示了各种链路宽度实现的聚合带宽数。从此表中可以明显看出, PCI Express 可实现的峰值带宽明显高于当今大多数现有总线。

考虑一下这些带宽数是如何计算的。目前, 每个方向的发送/接收速率为每通道 2.5Gbps。为了在数据传输和接收期间支持更高层次的稳健性, 将要发送的每个数据字节都转换为一个 10 位代码 (通过发送器器件中的 8b/10b 编码器)。换句话说, 每发送一个字节的数据实际上会传输 10 位的编码数据。结果是传输一个数据字节会产生 25% 的开销。PCI Express 实现了双工/单工链路, 这表示数据是同时发送和接收的。

聚合带宽假定两个方向同时传输流量。要获得表 2-1 中的聚合带宽数, 请将每秒 2.5Gbits 乘以 2 (以考虑两个方向), 然后乘以通道数, 最后除以每字节 10 位 (以考虑 8 至 10 位编码), 以得到每秒字节数。

**表 2-1. PCIe 链路速度**

PCIExpress 链路宽度	x1	x2	x4	x8	x12	x16	x32
AggregateBand - 宽度 (GB/s)	0.5	1	2	4	6	8	16

## 2.2 AM64x 和 AM243x 上的 PCIe 特性

这里有一个 PCIe 子系统实例。以下是一些主要特性：

- EP 和 RC 运行
- 第 1 代和第 2 代运行速度
- x1 通道支持
- 传统中断
- MSI (消息信号中断)

### EVM

该 EVM 上有一个 PCIe 子系统实例。以下是该实例的一些详细信息：

**表 2-2. AM64x 和 AM243x EVM 上的 PCIe**

实例	支持的通道	支持的连接器
PCIE0	1 通道	标准母连接器

有关更多信息：[PCIe](#)

### 3 x86 作为 RC , AM64x 作为 EP

在这里，我们将介绍在作为 RC 的通用 x86 PC 和作为 EP 的 AM64x 之间建立 PCIe 连接的步骤。

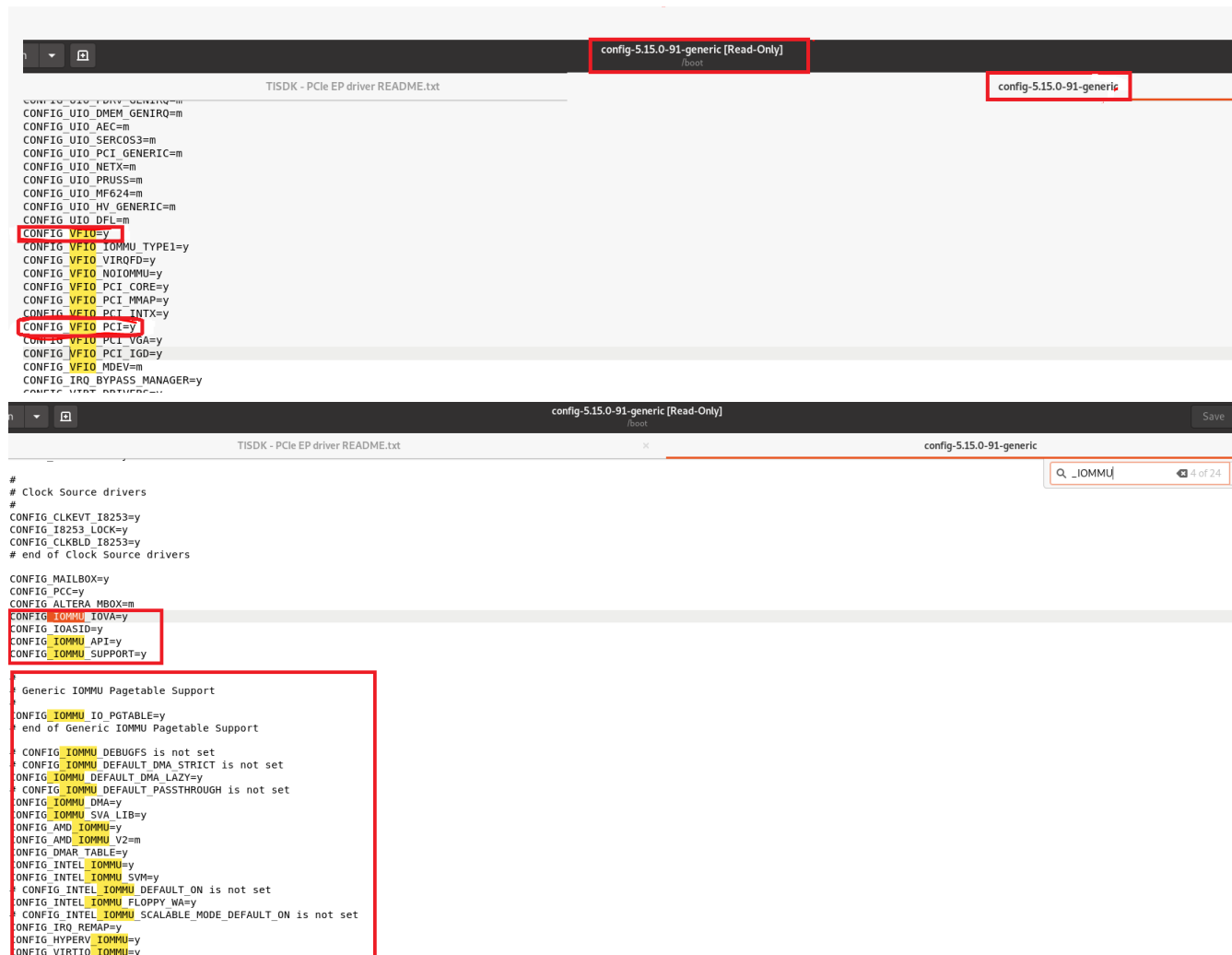
#### 3.1 硬件环境

以下硬件用于执行指定的功能测试：

- 具有可用 PCIe 插槽的 x86 系统
  - PCIe 插槽不需要连接到 PCIe 交换机，而是需要直接连接到 x86 CPU 或 PCH。
  - 为了进行开发和测试，使用了以下系统：

```
root@sitarampuapps-ThinkStation-P620: /home/sitara_mcu_apps# uname -a
Linux sitarampuapps-ThinkStation-P620 5.15.0-91-generic #101-20.04.1-Ubuntu SMP Thu Nov 16 14:22:28 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

- x86 需要安装支持 VFIO、VFIO-PCI 和 IOMMU 的最新 Linux 版本。



- [TMDS243EVM](#) 或 [TMDS64EVM](#) : 日后称为一个 EVM
- [PCIE\\_FLEX\\_NOCLK](#) : [Adex Electronics PCIe 柔性扩展器电缆 PE-FLEX1-G2.MMCX-12-TI1](#)
- [PCIE\\_FLEX\\_CLK](#) : 经修改的 [Adex Electronics PCIe 柔性扩展器电缆 PE-FLEX1-G2.MMCX-12-TI1](#) , 具有连接的基准时钟 REFCLK+/-
- [SPEC\\_ANA](#) : 频谱分析仪
- 需要修改 AM24x EVM 以支持 PCIe 通用基准时钟 : - 移除电阻 R661、R662、R667 和 R668 - 组装电阻 R665、R666、R679 和 R680 ( 全部为 0 欧姆 )

## 3.2 软件环境

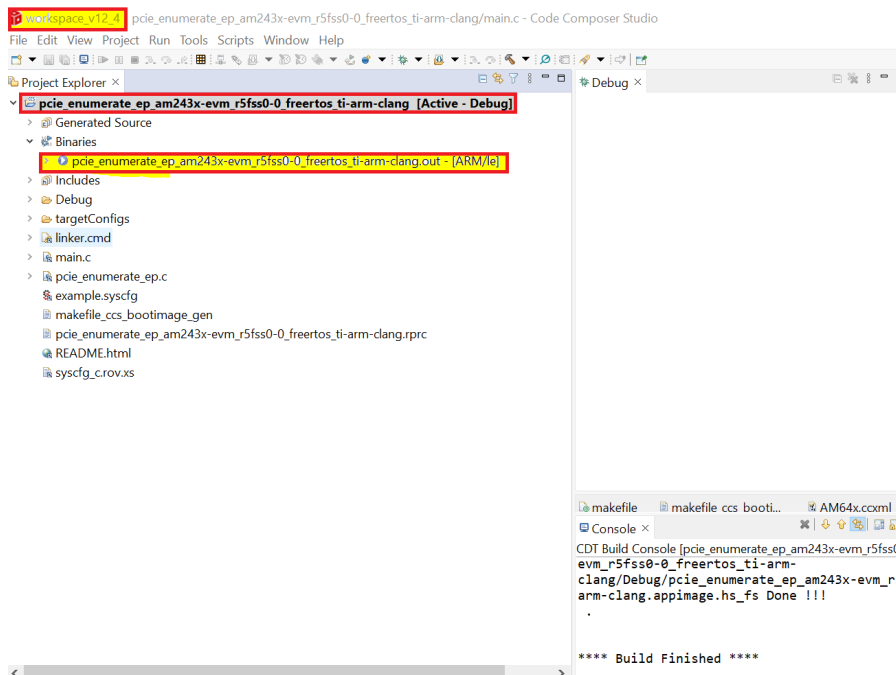
以下软件用于执行指定的功能测试：

1. [CCS12](#) : TI Code Composer Studio , 版本 : 12.4.0.00007 2。
2. [SysConfig](#) : 版本 : 1.17.0 3。
3. SER\_TER : 串行终端仿真器程序 , 例如 Tera Term 或 Putty
4. LIN: Lenovo ThinkStation-P620 具有 Ubuntu 20.04
5. WIN : Windows 10 22H2
6. MCU+ SDK
7. EP 示例应用程序
  - a. “PCIe 枚举 (EP)” 示例演示了一个支持通过运行 Windows 或 Linux 的 RC 进行枚举的 EP。

### 3.2.1 构建应用程序

位置 : examples/drivers/pcie/pcie\_enumerate\_ep

该示例可以导入到 CCS 12.4 中 , 并构建为常规 CCS 工程。



### 3.2.2 用法

- 您可以参阅 [MCUSDK 文档](#) , 使用部分将生成的二进制文件刷写到 OSPI 中
- 如果在未连接到 x86 RC 的 AM24x EVM 上启动 `pcie_enumerate_ep` 示例 , 或者 x86 未通电 , 则唯一的输出将是 :
  - 关闭 AM64x 和 Linux-PC 电源
  - 为 AM64x 上电 , 刷写 `pcie_enumerate_ep`

```

y
DMSC Firmware Version 9.0.7--v09.00.07 (Kool Koala)
DMSC Firmware revision 0x9
DMSC ABI revision 3.1

[BOOTLOADER_PROFILE] Boot Media      : NOR SPI FLASH
[BOOTLOADER_PROFILE] Boot Media Clock : 166.667 MHz
[BOOTLOADER_PROFILE] Boot Image Size : 0 KB
[BOOTLOADER_PROFILE] Cores present   :
r5f0-0
[BOOTLOADER PROFILE] SYSFW init      :      12192us
[BOOTLOADER PROFILE] System_init     :       548us
[BOOTLOADER PROFILE] Drivers_open    :       285us
[BOOTLOADER PROFILE] Board_driversOpen :    21966us
[BOOTLOADER PROFILE] Sciclient Get Version :     9840us
[BOOTLOADER PROFILE] CPU Load        :    186155us
[BOOTLOADER_PROFILE] SBL Total Time Taken :   230991us

Image loading done, switching to application ...
PCIe: initialized and waiting for link

```

之后，示例应用程序等待建立 PCIe 链路，这需要 RC。连接电缆并对 RC 通电后，应用程序输出状态将发生变化：

- 现在打开 Linux PC 电源

```

y
DMSC Firmware Version 9.0.7--v09.00.07 (Kool Koala)
DMSC Firmware revision 0x9
DMSC ABI revision 3.1

[BOOTLOADER_PROFILE] Boot Media      : NOR SPI FLASH
[BOOTLOADER_PROFILE] Boot Media Clock : 166.667 MHz
[BOOTLOADER_PROFILE] Boot Image Size : 0 KB
[BOOTLOADER_PROFILE] Cores present   :
r5f0-0
[BOOTLOADER PROFILE] SYSFW init      :      12192us
[BOOTLOADER PROFILE] System_init     :       548us
[BOOTLOADER PROFILE] Drivers_open    :       285us
[BOOTLOADER PROFILE] Board_driversOpen :    21966us
[BOOTLOADER PROFILE] Sciclient Get Version :     9840us
[BOOTLOADER PROFILE] CPU Load        :    186155us
[BOOTLOADER_PROFILE] SBL Total Time Taken :   230991us

Image loading done, switching to application ...
PCIe: initialized and waiting for link
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
■

```



## 4 测试设置

### 4.1 适用于 LINUX 和 WIN 的通用设置

要将 TMDS243EVM/TMDS64EVM 用作 PCIe EP 以及基于 Linux 的 PCIe RC 或基于 Windows 的 RC 执行功能测试，需要执行以下测试设置：

1. 在 TMDS243EVM 上执行硬件修改：
  - a. 移除电阻器 R661、R662、R667 和 R668
  - b. 组装 0Ω 电阻器 R665、R666、R679 和 R680
2. 移除 TMDS243EVM 上的跳线 J34，因为我们既不希望 AM24x 驱动 PERST 信号（EP，这是输入），也不希望 x86 的 PERST 信号复位我们的处理器，因为我们希望在 x86 之前引导 AM24x，以确保满足启动和复位时序要求。
3. 在基于 X86 Linux 的 RC 硬件的 BIOS 设置中启用 Intel 虚拟化技术以使用 IOMMU。
4. 使用修改后的电缆 PCIE\_FLEX\_CLK 将 TMDS243EVM 与 X86\_10TH 或 X86\_ADLN 连接。
5. 请注意，下图中用蓝色圆圈标明的零欧姆电阻器：

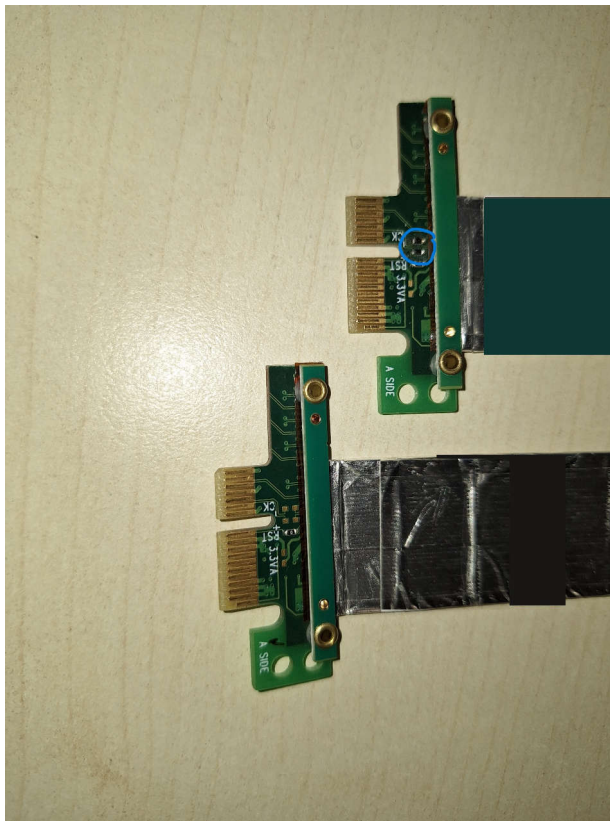


图 4-1. 修改的 PCIE\_FLEX\_CLK 电缆

6. 通过 SD 卡使用 NULL SBL 来引导 TMDS243EVM。
7. 使用串行终端仿真器程序 SER\_TER 通过适用于 UART 的 FTDI USB 端口连接到 TMDS243EVM。

#### 备注

串行连接旨在提供 PCIe EP 的各种状态消息，可用于如节 5 和节 6 中所述的测试验证

节 5 中所述的每项功能测试都需要修改 `pcie_enumerate_ep` 示例应用程序。这些修改在 Sysconfig 文件上或 CCS 上的源代码内执行。执行这些修改时，请继续执行以下测试设置：

1. 更改 Sysconfig 文件或源代码时，保存文件并构建工程。
2. 打开为特定目标设计的配置并与目标连接。在 TMDS243EVM 上执行系统复位，加载并运行 `pcie_enumerate_ep` 示例应用程序。

### 3. 基于 X86 的引导 PCIe RC。

备注

PCIe EP 需要始终在 PCIe RC 硬件引导之前运行。

### 4. 通过终端命令成功启动时执行测试验证，如节 5 中所述。

## 4.2 Linux 驱动程序(VFIO)

### 4.2.1 先决条件

要使用 ti-sample-vfio 示例驱动程序，目标系统的 Linux 需要支持以下功能：

- VFIO、VFIO\_PCI
  - CONFIG\_VFIO、CONFIG\_VFIO\_PCI

```

TISDK - PCIe EP driver README.txt
CONFIG_UIO_DMEN_GENERIC=m
CONFIG_UIO_AEC=m
CONFIG_UIO_SERCOS3=m
CONFIG_UIO_PCI_GENERIC=m
CONFIG_UIO_NETX=m
CONFIG_UIO_PRUSS=m
CONFIG_UIO_MF624=m
CONFIG_UIO_HV_GENERIC=m
CONFIG_UIO_DFL=m
CONFIG_VFIO=y
CONFIG_VFIO_IOMMU_TYPE1=y
CONFIG_VFIO_VIOPFD=y
CONFIG_VFIO_NOIOMMU=y
CONFIG_VFIO_PCI_CORE=y
CONFIG_VFIO_PCI_MMAP=y
CONFIG_VFIO_PCI_INTX=y
CONFIG_VFIO_PCI=y
CONFIG_VFIO_PCI_VGA=y
CONFIG_VFIO_PCI_IGD=y
CONFIG_VFIO_MDEV=m
CONFIG_IRQ_BYPASS_MANAGER=y
    
```

- IOMMU
  - CONFIG\_IOMMU、CONFIG\_INTEL\_IOMMU
  - 通过内核命令行参数 intel\_iommu=on 启用
  - 需要在 BIOS 中启用 Intel VT-d 支持

```

TISDK - PCIe EP driver README.txt
# Clock source drivers
CONFIG_CLKSRC_I8253=y
CONFIG_CLKSRC_LIC8253=y
CONFIG_CLKSRC_I8253=y
# end of clock source drivers
CONFIG_HAILBOOM=y
CONFIG_PCC=y
CONFIG_ALTERA_N10K=m
CONFIG_CXL=y
CONFIG_CXL_IOMMU=y
CONFIG_CXL_APM=y
CONFIG_CXL_SUPPORT=y
Generic IOMMU PageTable Support
CONFIG_IOMMU_ID_POTABLE=y
# end of Generic IOMMU PageTable Support
CONFIG_IOMMU_DEBUGFS is not set
CONFIG_IOMMU_DEFAULT_DMA_STRICT is not set
CONFIG_IOMMU_DEFAULT_DMA_LAZY=y
CONFIG_IOMMU_DEFAULT_PASSTHROUGH is not set
CONFIG_IOMMU_SVA=y
CONFIG_IOMMU_SVA_LIB=y
CONFIG_IOMMU=y
CONFIG_IOMMU_V2=m
CONFIG_IOMMU_STABLE=y
CONFIG_INTEL_IOMMU=y
CONFIG_INTEL_IOMMU_Snp=y
CONFIG_INTEL_IOMMU_DEFAULT_ON is not set
CONFIG_INTEL_IOMMU_FLOPPY_MAP=y
CONFIG_INTEL_IOMMU_SCALABLE_MODE_DEFAULT_ON is not set
CONFIG_IRQ_REMAP=y
CONFIG_HYPERV_IOMMU=y
CONFIG_VFIO_IOMMU=y
    
```

### 4.2.2 构建

ti-sample-vfio 示例驱动程序包含一个可使用简单 GCC 命令在目标上编译的 C 语言文件：

```
gcc ti-sample-vfio.c -o ti-sample-vfio -g -O2
```

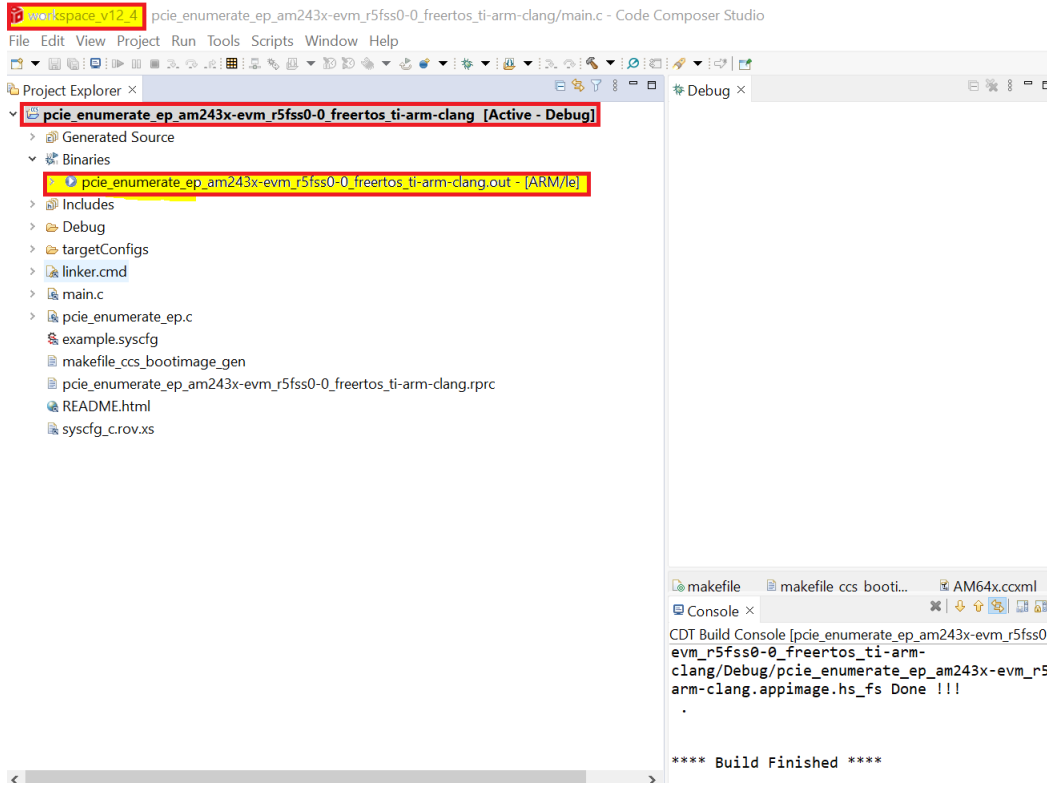
### 4.2.3 部署

由于该驱动程序可以在 Linux 上自行编译，因此没有单独的部署步骤。

## 4.3 测试应用程序使用情况

- 若要运行 pcie\_enumerate\_ep 示例应用程序，需要使用 NULL 引导加载程序 (SOC 初始化二进制) 来引导 AM24x EVM。

- 然后，可通过 CCS 12.4 和板载 XDS110 来载入 `pcie_enumerate_ep` 示例。



- `pcie_enumerate_ep` 示例将输出显示在 EVM 的调试 UART 上。

```

j
DMSC Firmware Version 9.0.7--v09.00.07 (Kool Koala)
DMSC Firmware revision 0x9
DMSC ABI revision 3.1

[BOOTLOADER_PROFILE] Boot Media      : NOR SPI FLASH
[BOOTLOADER_PROFILE] Boot Media Clock : 166.667 MHz
[BOOTLOADER_PROFILE] Boot Image Size : 0 KB
[BOOTLOADER_PROFILE] Cores present   :
r5f0-0
[BOOTLOADER_PROFILE] SYSFW init       :      12192us
[BOOTLOADER_PROFILE] System_init     :       548us
[BOOTLOADER_PROFILE] Drivers_open    :       285us
[BOOTLOADER_PROFILE] Board_driversOpen :    21966us
[BOOTLOADER_PROFILE] Sciclient Get Version :    9840us
[BOOTLOADER_PROFILE] CPU Load        :    186155us
[BOOTLOADER_PROFILE] SBL Total Time Taken :   230991us

Image loading done, switching to application ...
PCIe: initialized and waiting for link
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
    
```

- 如果在未连接到 x86 RC 的 AM24x EVM 上启动 `pcie_enumerate_ep` 示例，或者 x86 未通电，则唯一的输出将是：

```
PCIe: initialized and waiting for link
```

- 之后，示例应用程序等待建立 PCIe 链路，这需要 RC。连接电缆并对 RC 通电后，应用程序输出状态将发生变化：

```

PCIe: link detected
PCIe Link Parameter: PCIe Gen1 with 2.5 GT/s speed, Number of Lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
    
```

- 此时，应用程序已准备好通过 RC 驱动程序（可以是 `ti-sample-vfio` 或 `ti-sample-kmdf`）进行配置。

## 4.4 LINUX PC 的设置步骤

由于实现了基于 Linux VFIO 驱动程序的 RC 示例应用程序 *ti-sample-vfio* 以用于测试和验证，因此节 5 中所述的某些测试需要使用该应用程序。要使用 *ti-sample-vfio*，需要实现以下设置：

1. PCIe 成功引导后，打开 Linux 终端并获得 root 权限：

```
sudo su
```

2. 在 Linux 终端中使用 `lspci` 命令确定 TMD5243EVM PCIe EP 器件的总线、器件和功能编号。使用 Sysconfig 中设置的供应商和器件 ID。使用以下命令进行搜索，该命令会以数字形式显示所有 PCIe 器件的相关信息：

```
lspci -vtn
```

```

\-[0000:00]--00.0 Advanced Micro Devices, Inc. [AMD] Starship/Matisse Root Complex
+--00.2 Advanced Micro Devices, Inc. [AMD] Device 164f
+--01.0 Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
+--01.1-[01]---00.0 Aquantia Corp. AQ107 NBase-T/IEEE 802.3bz Ethernet Controller [AQtion]
+--02.0 Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
+--03.0 Advanced Micro Devices, Inc. [AMD] Starship/Matisse PCIe Dummy Host Bridge
+--03.1-[02-07]---00.0-[03-07]---00.0-[04]---00.0 Cadence Design Systems, Inc. Device 0100
|
|   +-08.0-[05]---00.0 Advanced Micro Devices, Inc. [AMD] Starship/Matisse Reserved SPP
|   |
|   |   +-00.1 Advanced Micro Devices, Inc. [AMD] Matisse USB 3.0 Host Controller
|   |   \--00.3 Advanced Micro Devices, Inc. [AMD] Matisse USB 3.0 Host Controller
+--09.0-[06]---00.0 Advanced Micro Devices, Inc. [AMD] FCH SATA Controller [AHCI mode]
\--0a.0-[07]---00.0 Advanced Micro Devices, Inc. [AMD] FCH SATA Controller [AHCI mode]
    
```

3. 下图展示了上一命令的输出。在本例中，PCIe EP 被分配了总线 4、器件 00 和功能 0。
4. 使用 `modprobe` 加载 VFIO-PCI 驱动程序：

```
modprobe vfio-pci
```

5. 将 TMD5243EVM PCIe EP 供应商和器件 ID 分配给 VFIO 驱动程序。

```
echo "17cd 0100" > /sys/bus/pci/drivers/vfio-pci/new_id
```

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# modprobe vfio-pci
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# echo "17cd 0100" > /sys/bus/pci
pci/      pci-epf/   pci-express/
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# echo "17cd 0100" > /sys/bus/pci/drivers/vfio-pci/new_id
    
```

6. 检查 PCIe EP 被分配到的 IOMMU 组：

```
readlink /sys/bus/pci/devices/0000:04:00.0/iommu_group
```

a. 上一命令的输出可提供：分配了 IOMMU 组的 PCIe EP。

7. 确保 EP 是此 IOMMU 组中的唯一器件：

```
ls -l /sys/bus/pci/devices/0000:04:00.0/iommu_group/devices
```

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# ls -l /sys/bus/pci/devices/0000:04:00.0/iommu_group/devices/
total 0
lrwxrwxrwx 1 root root 0 Jan 15 14:39 0000:04:00.0 -> ../../../../devices/pci0000:00/0000:00:03.1/0000:02:00.0/0000:03:00.0/0000:04:00.0
    
```

可以看出，TMD5243EVM PCIe EP 是 IOMMU 组 60 中的唯一器件。如果同一 IOMMU 组内有其他 PCIe 器件，则这些 PCIe 器件也必须绑定到 VFIO 驱动程序。

8. 编译 *ti-sample-vfio* (后续步骤不再需要 root 权限)：

```
gcc ti-sample-vfio.c -o ti-sample-vfio -g -O2
```

9. 使用先前确定的参数执行 *ti-sample-vfio* 应用程序：

```
sudo ./ti-sample-vfio 40 0 60 1 wait
```

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIE/TISDK - PCIE EP Driver - 1.0protol/TISDK - PCIE EP Driver - 1.0protol/PCIE EP Driver - 1.0protol/src/ti-sample-vfio# ./ti-sample-vfio
4 0 0 60
Using PCI device 0000:04:00.0 in IOMMU group 60
pre-SET CONTAINER:
VFIO_CHECK_EXTENSION VFIO_TYPE1_IOMMU: Present
VFIO_CHECK_EXTENSION VFIO_NOIOMMU_IOMMU: Not Present
post-SET CONTAINER:
VFIO_CHECK_EXTENSION VFIO_TYPE1_IOMMU: Present
VFIO_CHECK_EXTENSION VFIO_NOIOMMU_IOMMU: Not Present
Config Region Info: region index 0x7, size 0x1000, offset 0x700000000000, cap_offset 0x0, flags 0x3
BAR0 Info: size 0x8000, offset 0x0, flags 0xf
MSI IRQ Info: Index: 1, Count: 1, Flags: 9
RC completed EP initialization
RC resets EP
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIE/TISDK - PCIE EP Driver - 1.0protol/TISDK - PCIE EP Driver - 1.0protol/PCIE EP Driver - 1.0protol/src/ti-sample-vfio#

```

**备注**

*ti-sample-vfio* 应用程序需要以下参数才能执行：

1. [bus] : PCIe EP 总线编号
2. [device] : PCIe EP 器件型号
3. [function] : PCIe EP 功能编号
4. [IOMMU group] : PCIe EP IOMMU 组
5. [test\_mode] 仅测试用例 4.3.2 需要此参数，它指的是扩展入站 ATU/BAR 配置 ( 请参阅相应的说明 )。要启动测试用例，必须传递参数 *testbars*。否则，可以省略此参数，输入将解释为后续参数 [Number of MSI IRQs]。
6. [Number of MSI IRQs] : 测试 4.5.2 中描述的要测试的 MSI IRQ 的数量。如果未执行测试 4.5.2，则该参数需要设置为 1。
7. [Number of loops] : 测试程序可以执行的环路数量。此参数是可选的，可以留空。默认值是 10。
8. [ 'wait' ] 此参数用于指示测试程序在测试应用程序执行期间等待用户输入。

**4.4.1 UART 控制台输出**

运行示例应用程序会使器件从 D3hot 进入 D0 状态。

在执行采样直至 EP 最终返回到 D3hot 状态时，应用程序输出进一步的状态变化：

```

EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: MSI enabled with 1 vector(s) using address fee00538 and data 0
APPL: EP configured
APPL: EP unconfigured
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state
PCIe: signaling APPL halt
APPL: pcie not ready

```

```

r5f0-0
[BOOTLOADER PROFILE] SYSFW init           : 12192us
[BOOTLOADER PROFILE] System_init         : 547us
[BOOTLOADER PROFILE] Drivers_open        : 285us
[BOOTLOADER PROFILE] Board_driversOpen   : 22008us
[BOOTLOADER PROFILE] Sciclient_Get_Version : 9845us
[BOOTLOADER PROFILE] CPU_Load            : 184217us
[BOOTLOADER_PROFILE] SBL_Total_Time_Taken : 229098us

Image loading done, switching to application ...
PCIe: initialized and waiting for link
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state
PCIe: signaling APPL halt
APPL: pcie not ready
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: MSI enabled with 1 vector(s) using address fee00000 and data 0
APPL: EP configured
APPL: EP unconfigured
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe Link Parameter: PCIe Gen2 with 5.0 GT/s speed, Number of Lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state
PCIe: signaling APPL halt
APPL: pcie not ready
■

```

#### 4.5 MSI 示例

该测试设置用于测试 PCIe MSI RC 工程 `pcie_msi_irq_rc_am243x-evm_r5fss0-0_nortos_ti-arm-clang` 以及 PCIe EP 示例工程 `pcie_msi_irq_rc_am243x-evm_r5fss0-0_nortos_ti-arm-clang`。

1. 通过未修改的电缆 PCIE\_FLEX\_NOCLK 连接两个 TMDS243EVM 或两个 TMDS64EVM。



图 4-2. 通过未修改的 PCIe 电缆连接两个 AM64x

2. 对于第一个 TMDS243EVM，打开 CCS，导入并编译 `pcie_msi_irq_rc_am243x-evm_r5fss0-0_nortos_ti-arm-clang`。打开为特定目标设计的配置并与目标连接。执行系统复位并加载示例应用程序。
3. 使用串行终端仿真器程序 `SER_TER` 通过适用于 UART 的 FTDI USB 端口连接到第一个 TMDS243EVM。



- 对于第二个 TMD5243EVM，打开第二个 CCS 应用程序（这可能需要额外的工作区），导入并编译 `pcie_msi_irq_ep_am243x-evm_r5fss0-0_nortos_ti-arm-clang`。打开为特定目标设计的配置并与目标连接。执行系统复位并加载示例应用程序。
- 使用串行终端仿真器程序 SER\_TER 通过适用于 UART 的 FTDI USB 端口连接到第二个 TMD5243EVM。

## 4.6 Windows PC 的设置步骤

要使用该驱动程序，可以运行 `ti-sample-console` 应用程序。此应用程序将打开驱动程序，将 IOCTL 发送至驱动程序，并等待 IOCTL 返回。

KMDF 驱动程序使用随 IOCTL 一起发送的 *模式* 来填充 EP 的 Bar0 数据区域，然后在 EP 中触发下游中断。EP 将数据区域从 Bar0 复制到 RC 驱动程序的 DMA 缓冲区，并在 RC 中触发 MSI。RC 处理该中断，并使用通过 DMA 发回的数据来回复 IOCTL

此测试设置适用于：

- Windows-PC 作为 PCIe Rc
- AM243x 作为 PCIe EP

### 4.6.1 先决条件

要将 TI TMD5243EVM 用作 PCIe EP 和基于 Windows 的 RC 来执行功能测试，可以执行以下测试设置：

构建 Windows 驱动程序需要一个具有以下软件包的 Windows 主机：

- 适用于 Windows 10 的 WDK，2004 版
- Windows SDK 10.0.19041.685

Visual Studio 2019（专业版或社区版）

- 需要使用 Visual Studio 的安装程序将 `spectre` 缓解库添加为单个元件
- `ti-sample-kmdf` 和 `ti-sample-console` 源代码

目标机器需要安装以下软件：

- Windows 10 22H2
- 适用于 Windows 10 的 WDK，2004 版
- [Microsoft Visual C++ Redistributable](#)

### 4.6.2 构建

`ti-sample-kmdf` 解决方案包含两个工程：内核模式驱动程序 `ti-sample-kmdf` 和控制台应用程序 `ti-sample-console`。

通过在 Visual Studio 2019 中打开 `ti-sample-kmdf` 解决方案并构建整个解决方案，可以构建这两个工程。

- 该驱动程序当前仅在 x86 计算机上支持 64 位模式，因此需要选择 x64 平台和 `Release` 配置。
- 在解决方案资源管理器中，选择解决方案 `ti-sample-kmdf` 并生成解决方案。

编译输出位于解决方案目录 (`ti-sample-kmdf\x64\Release`) 下的新文件夹 `x64\Release` 中。

### 4.6.3 部署

目标机器上需要以下来自解决方案构建输出的文件：

- `ti-sample-kmdf.inf`
- `ti-sample-kmdf.sys`
- `ti-sample-kmdf.cat`
- `ti-sample-console.exe`

默认情况下，Windows 仅接受经签名的驱动程序。可以修改安装以接受所谓的经测试签名的驱动程序。Windows KMDF 示例驱动程序 `ti-sample-kmdf` 使用此方法，并构建为经测试签名的驱动程序。

要允许 Windows 使用经测试签名的驱动程序，请打开管理员提示符 (cmd，以管理员身份运行)，并输入以下命令：

```
Bcdedit.exe -set TESTSIGNING ON
```

启用测试签名需要重新引导系统。此时，需要如上所述启动带有 `pcie_enumerate_ep` 应用程序的 AM24x EVM。

然后，您需要在目标计算机上安装用于对驱动程序进行测试签名的证书。此证书与驱动程序一同放置在解决方案输出文件夹中，并命名为 `ti-sample-kmdf.cer`。

它可以通过管理员提示符使用 WDK 附带的 `CertMgr.exe` 工具进行安装：

- `cd C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64\`
- `CertMgr.exe /add ti-sample-kmdf.cer /s /r localMachine root /all`
- `CertMgr.exe /add ti-sample-kmdf.cer /s /r localMachine trustedpublisher`

### 备注

- 使用 `ti-sample-kmdf.cer` 的完整路径
- PCI 设备并验证硬件 ID 是否为 `PCI\VEN_17cd&DEV_0100`。右键点击设备

```
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

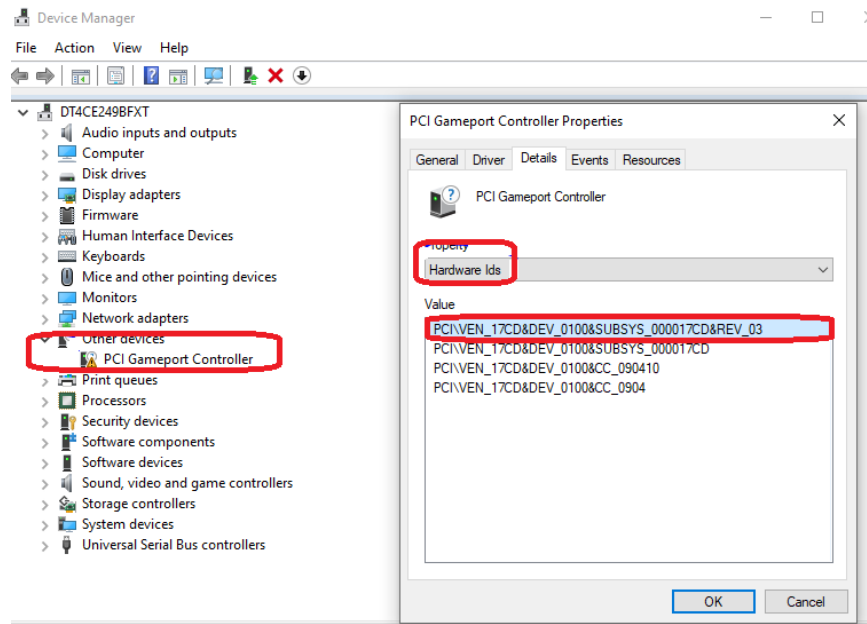
C:\windows\system32>cd C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64

C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64>CertMgr.exe /add ti-sample-kmdf.cer /s /r localMachine root /all
CertMgr.exe is not recognized as an internal or external command,
operable program or batch file.

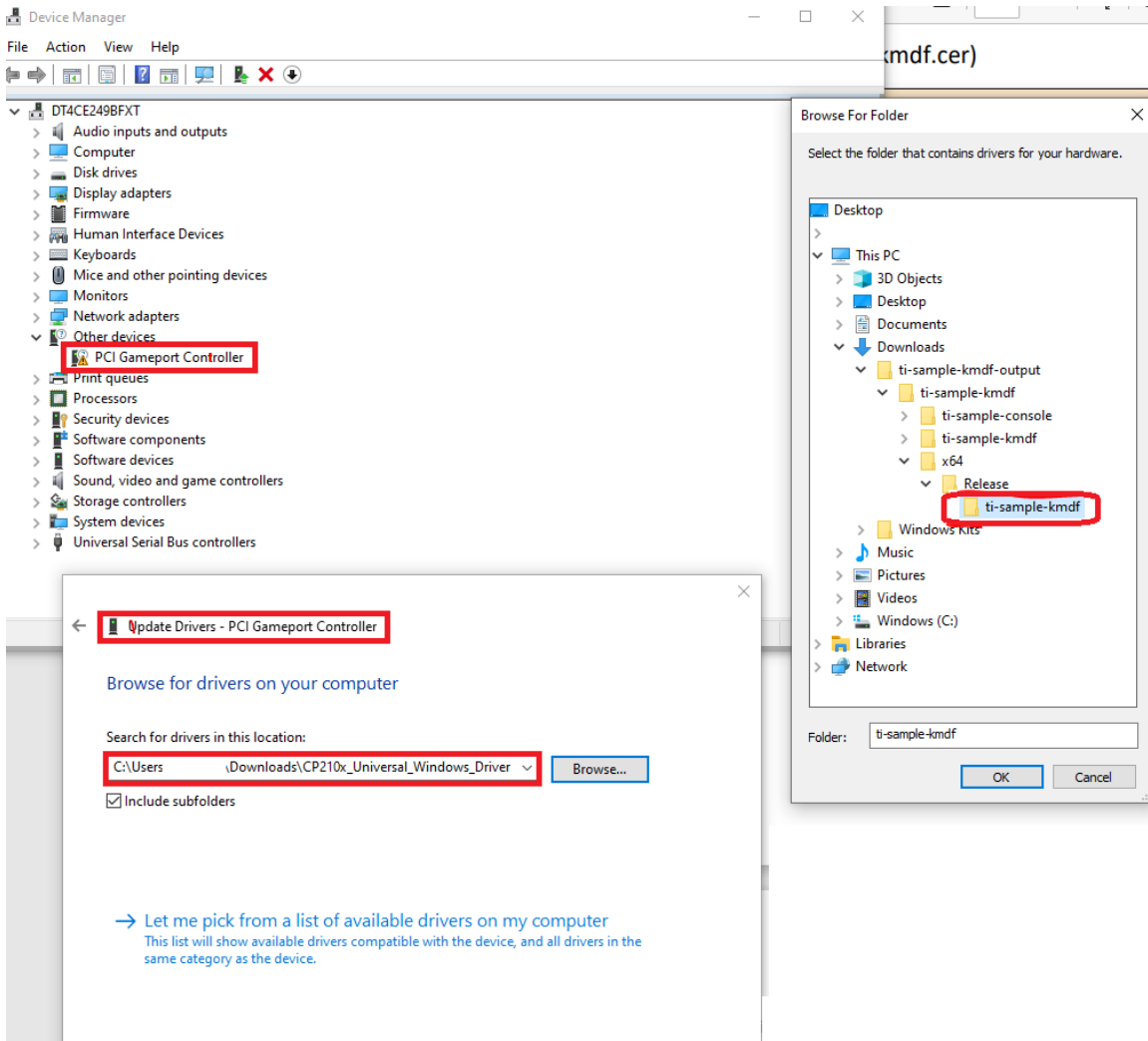
C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64>CertMgr.exe /add ti-sample-kmdf.cer /s /r localMachine root /all
CertMgr Succeeded

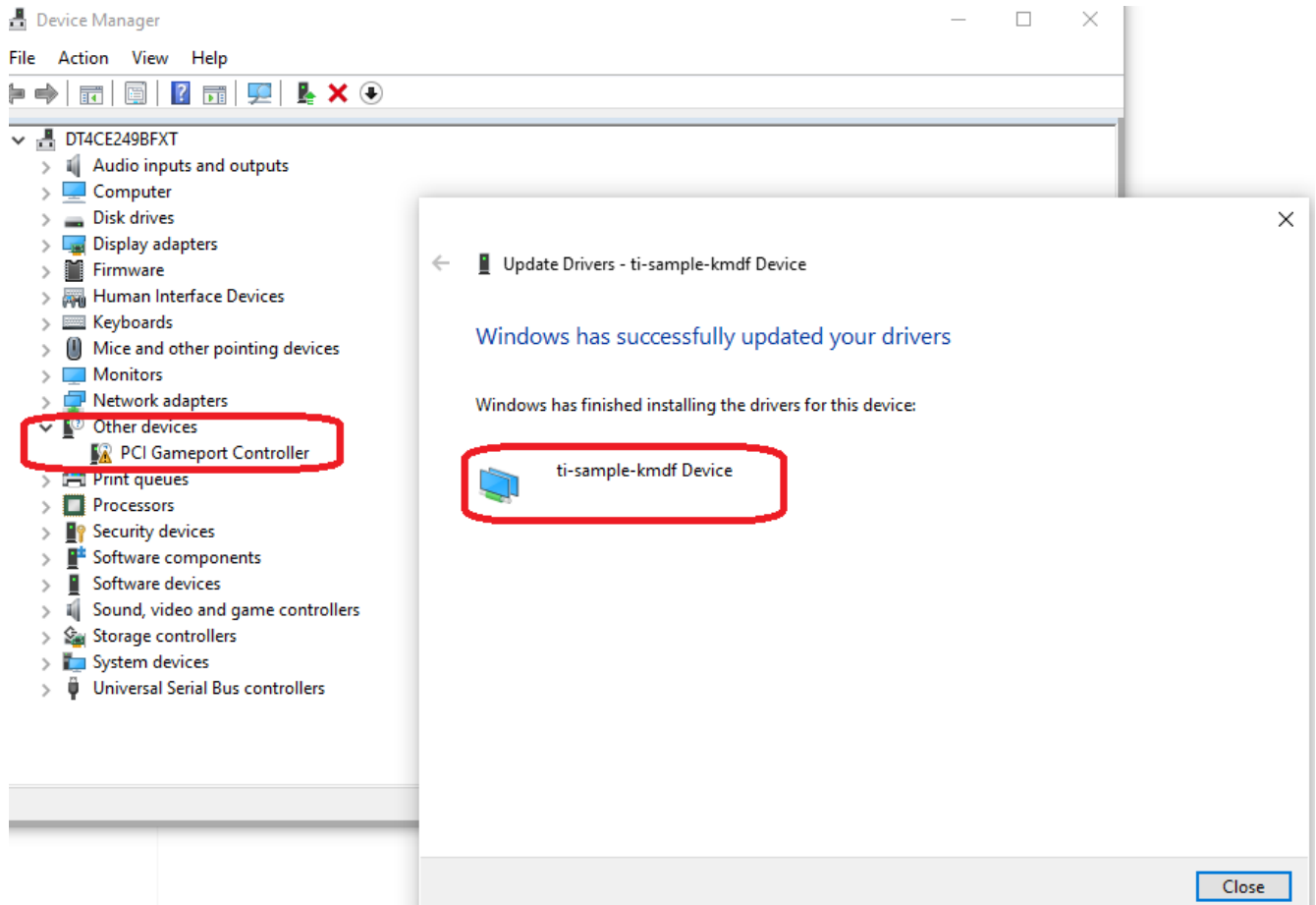
C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64>CertMgr.exe /add ti-sample-kmdf.cer /s /r localMachine trustedpublisher
CertMgr Succeeded

C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0\x64>
```









Windows 会安装驱动程序，然后在完成 ti-sample-kmdf 设备的驱动程序安装后通知您。

## 5 PCIe 测试规范

本章定义并指定了各种 PCIe 功能测试。根据测试说明，提供有关如何执行测试的详细说明以及所需结果的说明。节 4.4 中显示的以下测试规范是在假定环境下执行测试。

### 5.1 标识和配置功能

表 5-1. 标识和配置功能

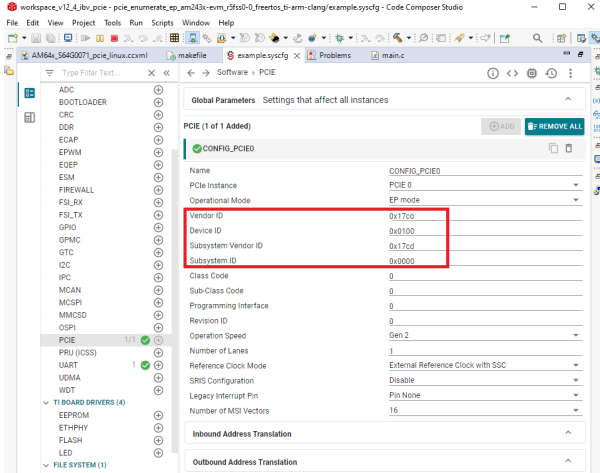
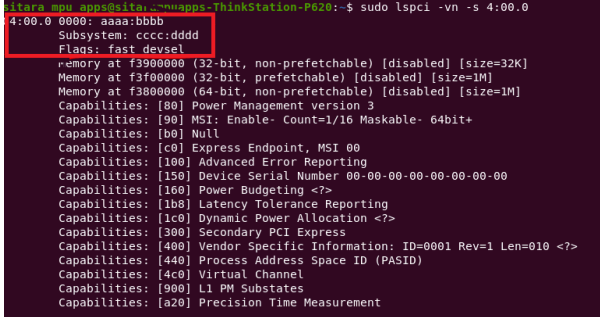
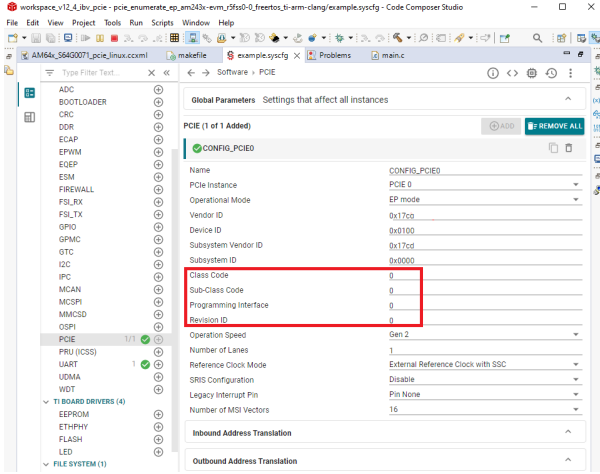
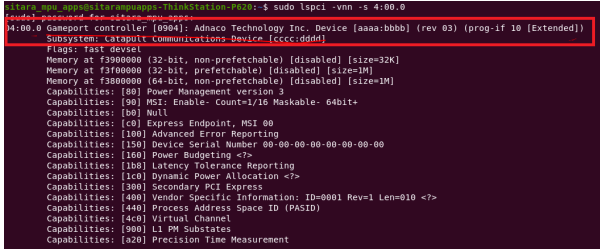
序号	测试规范
1	<p><b>说明：</b> 进行测试以验证是否可以在 TMD5243EVM PCIe EP 中正确设置和配置所需的 PCIe 供应商 ID、器件 ID、子系统 ID 和子系统供应商 ID。</p> <p><b>执行：</b></p> <ol style="list-style-type: none"> <li>通过 SysConfig 配置所需的 ID 等等： <ol style="list-style-type: none"> <li>供应商 ID : 0xAAAA</li> <li>器件 ID : 0xB BBB</li> <li>子系统供应商 ID : 0xC CCC</li> <li>子系统 ID : 0xD DDD</li> </ol> </li> </ol>  <ol style="list-style-type: none"> <li>确定 PCIe EP 总线、器件和功能编号，并验证 Linux 终端中配置的 ID，等等。 <ol style="list-style-type: none"> <li><code>sudo lspci -vn -s 4:00.0</code></li> <li>所需的 ID 按照 SysConfig 中的配置显示。下图展示了预期结果。</li> </ol> </li> </ol> 

表 5-1. 标识和配置功能 (续)

序号	测试规范
2	<p><b>说明：</b></p> <p>进行测试以验证是否可以在 PCIe EP 中正确设置和配置所需的 PCIe 类代码、子类代码、编程接口和修订 ID。仅出于测试目的，基类代码需要设置为“Input device (09h)”，并将子类代码设置为“Gameport controller (10h)”，将编程接口设置为“10h”。修订 ID 设置为“03h”。</p> <p>有关上述参数编码的更多信息，请参阅 PCI-SIG 发布的“PCI 代码和 ID 分配规范”。</p> <p><b>执行：</b></p> <ol style="list-style-type: none"> <li>通过 SysConfig 配置所需的设置。 <ol style="list-style-type: none"> <li>类代码：0x09</li> <li>子类代码：0x04</li> <li>编程接口：0x10</li> <li>修订 ID：0x03</li> </ol> </li> </ol>  <ol style="list-style-type: none"> <li>确定 PCIe EP 总线、器件和功能编号，并验证 Linux 终端中配置的设置，等等。 <ol style="list-style-type: none"> <li><code>sudo lspci -vnn -s 4:00.0</code></li> <li>所需的参数按照 SysConfig 中的配置显示。下图展示了预期结果。</li> </ol> </li> </ol> 

5.1.1 测试用例

测试

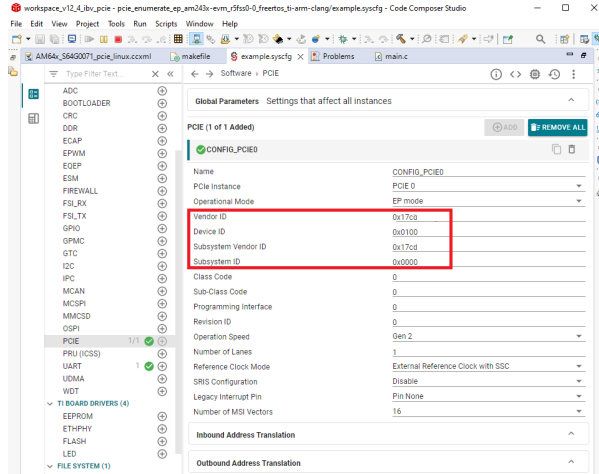
说明：

进行测试以验证是否可以在 TMD5243EVM PCIe EP 中正确设置和配置所需的 PCIe 供应商 ID、器件 ID、子系统 ID 和子系统供应商 ID。

执行：

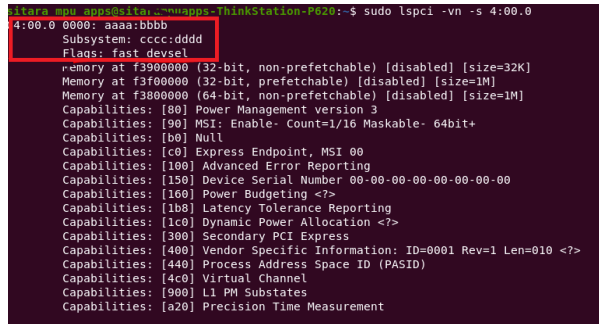
- 通过 SysConfig 配置所需的 ID 等等。
  - 供应商 ID：0xAAAA
  - 器件 ID：0xBBBB
  - 子系统供应商 ID：0xCCCC

d. 子系统 ID : 0xDDDD



2. 确定 PCIe EP 总线、器件和功能编号，并验证 Linux 终端中配置的 ID，等等。

```
sudo lspci -vn -s 4:00.0
```



所需的 ID 按照 SysConfig 中的配置显示。

测试

说明：

进行测试以验证是否可以在 PCIe EP 中正确设置和配置所需的 PCIe 类代码、子类代码、编程接口和修订 ID。

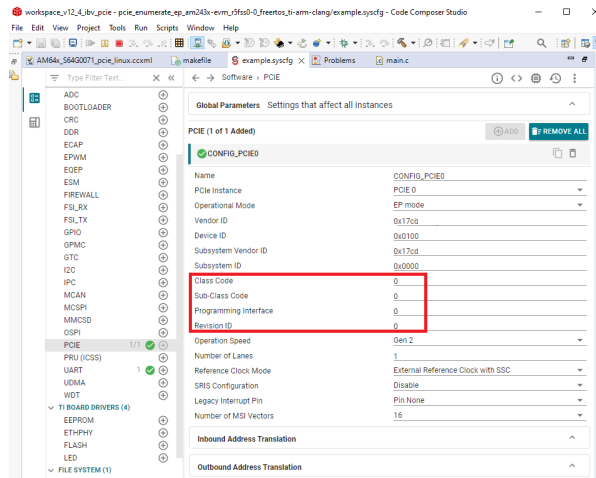
仅出于测试目的，基类代码需要设置为 *Input device (09h)*，并将子类代码设置为 *Gameport controller (10h)*，将编程接口设置为 *10h*。修订 ID 需要设置为 *03h*。

有关上述参数编码的更多信息，请参阅 PCI-SIG 发布的 *PCI 代码和 ID 分配规范*。

执行：

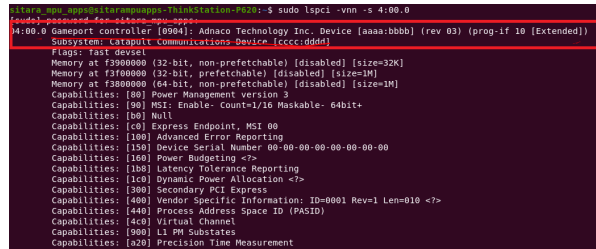
1. 通过 SysConfig 配置所需的设置。

- a. 类代码：0x09
- b. 子类代码：0x04
- c. 编程接口：0x10
- d. 修订 ID：0x03



2. 确定 PCIe EP 总线、器件和功能编号，并验证 Linux 终端中配置的设置，等等：

```
sudo lspci -vnn -s 4:00.0
```



所需的参数按照 SysConfig 中的配置显示。

## 5.2 基准时钟功能

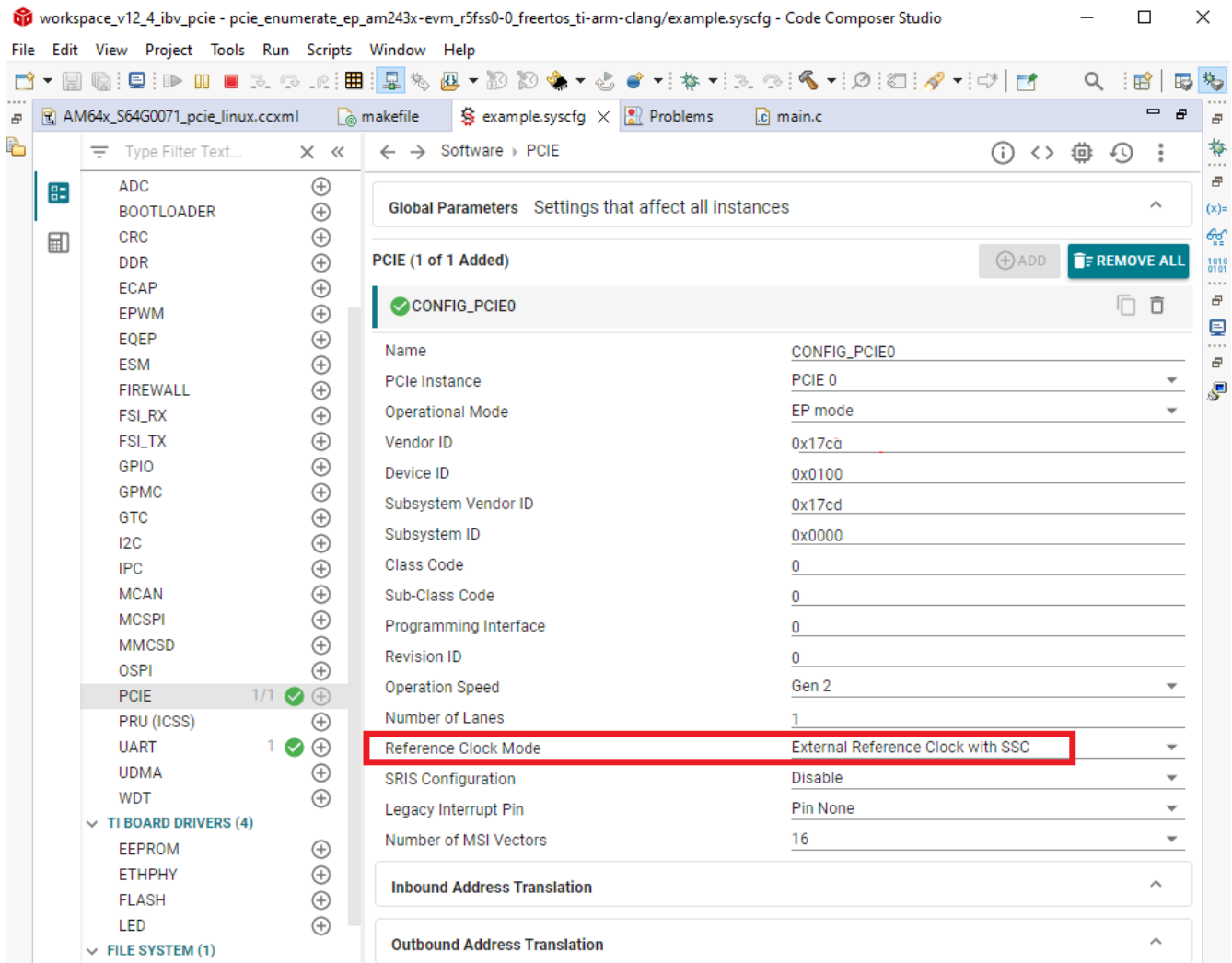
### 测试

#### 说明：

测试以验证是否可以在 TMD5243EVM PCIe EP 上配置外部基准时钟，以及 PCIe EP 是否可以在外部基准时钟下正常工作。

#### 执行：

1. 检查 Sysconfig 上是否配置了以下设置：
  - a. 基准时钟模式：外部基准时钟，无 SSC



2. 验证是否启用了通用时钟和插槽时钟机制，如下图所示。

```
DevSta: CorrErr- NonFatalErr- FatalErr- UnsupReq- AuxPwr- TransPend-
LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L1, Exit Latency L1 <8us
ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp+
LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommCLK+
ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
LnkSta: Speed 5GT/s (ok), Width x1 (ok)
TrErr- Train- SlotCLK+ DActive- BWMgmt- ABWMgmt-
```

## 测试

### 说明：

测试以验证 AM243X/AM64X 的内部基准时钟配置是否可以与启用的输出和 SSC 配置一同使用。

### 执行：

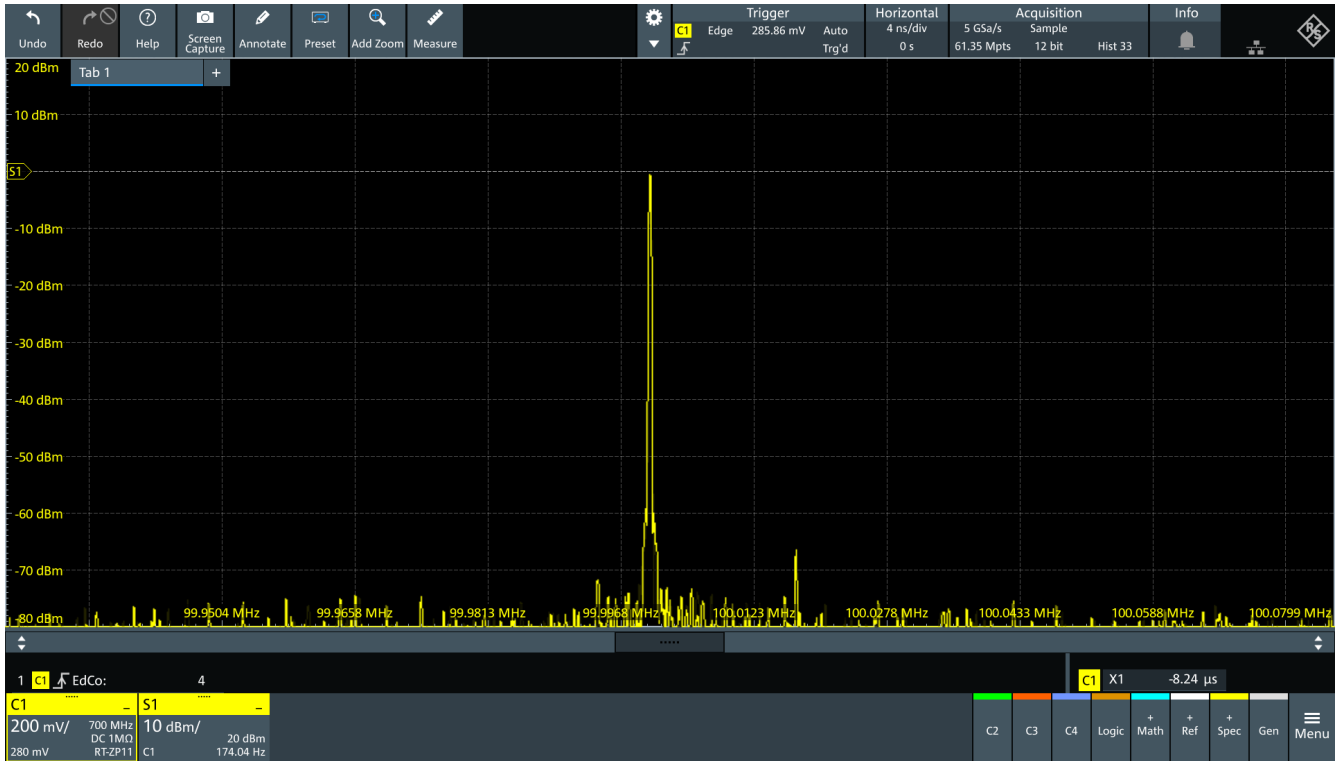
1. 从 TMD5243EVM PCIe EP 中移除 PCIe 电缆 PCIE\_FLEX\_CLK

### 备注

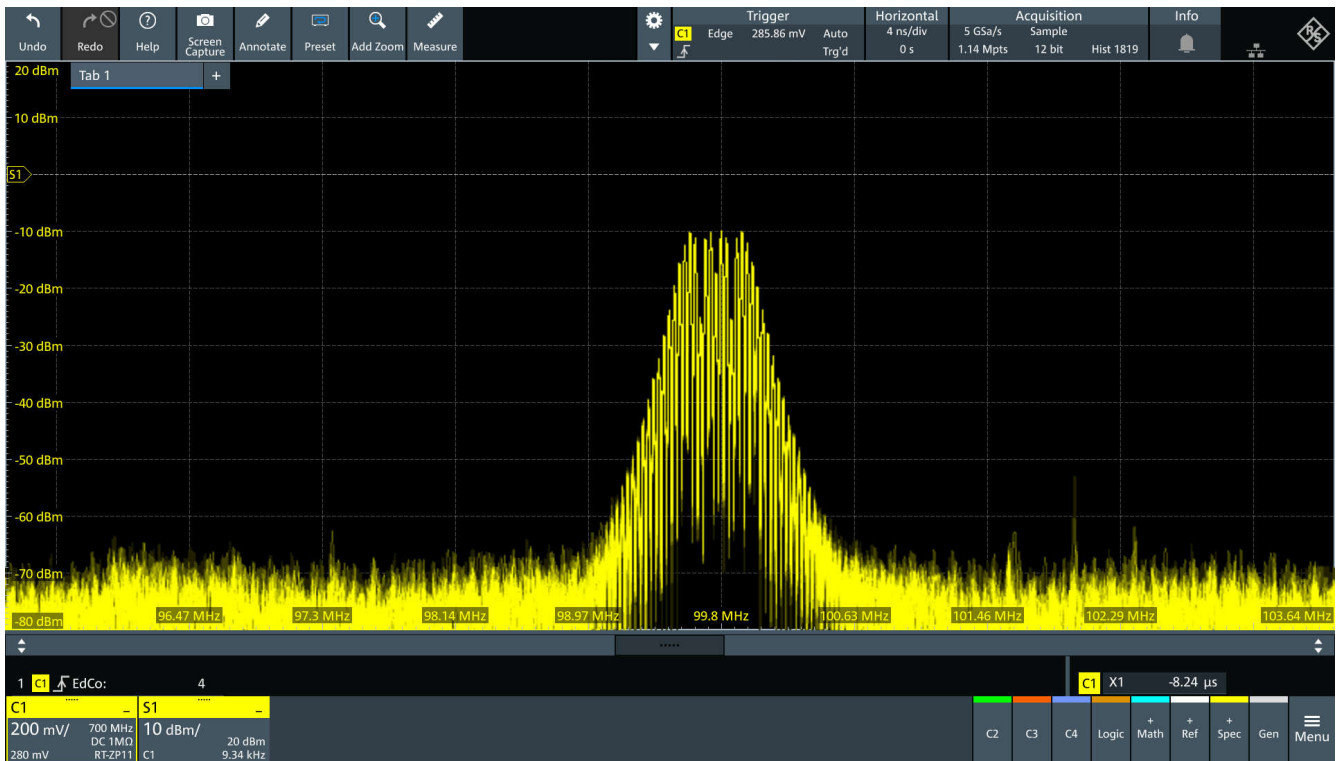
这很重要，因为有两个来自 EP 和 RC 的驱动 PCIe 基准时钟源会损坏硬件。

2. 通过 Sysconfig 配置以下设置：

- a. 内部基准时钟，无 SSC，输出使能。
3. 测量 PCIe 连接器上的 PCIe 基准时钟频谱。由于 SCC 被禁用，测得的频谱显示一个频率高于 100MHz 的噪声水平，如下图所示。



4. 通过 Sysconfig 配置以下设置：
  - a. 内部基准时钟，有 SSC，输出使能。
5. 由于启用了 SSC，因此测得的频谱可以分布在 100MHz 周围，如下图所示。



我们会探测 REFCLK+ 和各自的 GROUND 引脚，以在示波器上获取波形



[https://en.wikipedia.org/wiki/PCI\\_Express](https://en.wikipedia.org/wiki/PCI_Express)

PCI Express connector pinout (x1, x4, x8 and x16 variants)

Pin	Side B	Side A	Description	Pin	Side B	Side A	Description
1	+12 V	PRSENT1#	Must connect to farthest PRSENT2# pin	50	HSOp(8)	Reserved	Lane 8 transmit data, + and -
2	+12 V	+12 V	Main power pins	51	HSOn(8)	Ground	Lane 8 receive data, + and -
3	+12 V	+12 V		52	Ground	HSIp(8)	
4	Ground	Ground		53	Ground	HSIn(8)	
5	SMCLK	TCK		SMBus and JTAG port pins	54	HSOp(9)	Ground
6	SMDAT	TDI	55		HSOn(9)	Ground	
7	Ground	TDO	56		Ground	HSIp(9)	Lane 9 receive data, + and -
8	+3.3 V	TMS	57		Ground	HSIn(9)	
9	TRST#	+3.3 V	Aux power & Standby power	58	HSOp(10)	Ground	Lane 10 transmit data, + and -
10	+3.3 V aux	+3.3 V		59	HSOn(10)	Ground	
11	WAKE#	PERST#	Link reactivation; fundamental reset [23]	60	Ground	HSIp(10)	Lane 10 receive data, + and -
<b>Key notch</b>				61	Ground	HSIn(10)	
12	CLKREQ#[24]	Ground	Clock Request Signal	62	HSOp(11)	Ground	Lane 11 transmit data, + and -
13	Ground	REFCLK+	Reference clock differential pair	63	HSOn(11)	Ground	
14	HSOp(0)	REFCLK-	Lane 0 transmit data, + and -	64	Ground	HSIp(11)	Lane 11 receive data, + and -
15	HSOn(0)	Ground		65	Ground	HSIn(11)	
16	Ground	HSIp(0)	Lane 0 receive data, + and -	66	HSOp(12)	Ground	Lane 12 transmit data, + and -
17	PRSENT2#	HSIn(0)		67	HSOn(12)	Ground	
18	Ground	Ground	PCI Express x1 cards end at pin 18	68	Ground	HSIp(12)	Lane 12 receive data, + and -
				69	Ground	HSIn(12)	

### 5.3 进站 ATU 和 BAR 功能

#### 测试

#### 说明：

进行测试以验证 TMD5243EVM/TMD564EVM PCIe EP 的 PCIe 进站 ATU 和 BAR 配置是否正常工作。

默认情况下，在 SysConfig 中为 `pcie_enumerate_ep` 示例应用程序设置了以下 BAR 配置：

1. 进站地址转换 0：  
此 ATU 配置使用区域索引 0，该区域索引具有一个链接到外部结构 `bar0_mem` 的 32Kb 不可预取 32 位存储器 BAR。无法为该测试修改此进站 ATU 配置，因为它是确保基于 RC VFIO 的示例应用程序 `ti-sample-vfio` 正常运行所必需的。
2. 进站地址转换 1：  
此 ATU 配置使用区域索引 1，该区域索引具有一个链接到外部数据缓冲区 `bar1_data` 的 64MB 可预取 32 位存储器 BAR。可以为此测试修改进站 ATU，因为它专门用于测试各种 BAR 配置。
3. 进站地址转换 2：  
此 ATU 配置使用区域索引 2，该区域索引具有一个链接到外部数据缓冲区 `bar2_data` 的 1Gb 不可预取 64 位存储器 BAR。可以为此测试修改进站 ATU，因为它专门用于测试各种 BAR 配置。

#### 执行：

1. 在 SysConfig 文件中为 `pcie_enumerate_ep` 应用程序的进站地址转换 1 和 2 设置所需的 BAR 配置。
2. 检查所需的 PCIe EP BAR 是否在基于 Linux 的 RC 硬件上正确配置。启动时，配置的 BAR 可在 PCIe 配置空间中显示为禁用，如下图所示：

```

root@sitarampuapps-ThinkStation-P620: /home/sitara_mpu_apps
sitara_mpu_apps@sitarampuapps-ThinkStation-P620:~$ sudo su
[sudo] password for sitara_mpu_apps:
Sorry, try again.
[sudo] password for sitara_mpu_apps:
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 4:00.0
04:00.0 Gameport controller: Adnaco Technology Inc. Device bbbb (rev 03) (prog-if 10 [Extended])
Subsystem: Catapult Communications Device dddd
Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
Region 0: Memory at f2c00000 (32-bit, non-prefetchable) [disabled] [size=32K]
Region 1: Memory at f3e00000 (32-bit, prefetchable) [disabled] [size=2M]
Region 2: Memory at f2800000 (64-bit, non-prefetchable) [disabled] [size=4M]
Capabilities: [80] Power Management version 3
Flags: PMEclk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
Status: D0 NoSoftRst+ PME-Enable- DSEL=0 DScale=0 PME-
Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
Address: 0000000000000000 Data: 0000
    
```

- 运行 RC 示例应用程序 *ti-sample-vfio*。打开第二个 Linux 终端并检查 PCIe EP 配置空间。当程序在 EP 初始化和 BAR 映射后停止时，相应的 BAR 现在可被启用（未显示为禁用）。

```

Terminal ×
COM180 ×
INFO: Bootloader_runCpu:155: CPU r5f1-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU m4f0-0 is initialized to 400000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_loadSelfCpu:207: CPU r5f0-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_loadSelfCpu:207: CPU r5f0-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runSelfCpu:217: All done, resetting self ...

PCIe: EP initialized and waiting for link
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state
PCIe: signaling APPL halt
APPL: pcie not ready
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: MSI enabled with 1 vector(s) using address fee00000 and data 0
Mapping MSI target at 0xfee00000 - size 0xff...
Mapping DMA buffer at 0x0 - size 0xffff...
APPL: EP configured
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
DMA test done
Send MSI IRQ nr. 0
MSI test done
BAR test done
Disabling MSI mapping...
Disabling DMA buffer mapping...
APPL: EP unconfigured
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state

```

4. 继续执行 RC 示例应用程序 *ti-sample-vfio*。程序可以正常继续并在没有任何故障的情况下结束。

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps
sitara_mpu_apps@sitarampuapps-ThinkStation-P620:~$ sudo su
[sudo] password for sitara_mpu_apps:
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
    Subsystem: Catapult Communications Device dddd
    Control: I/O- Mem+ BusMaster- SpecCycle- MemWInv- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
    Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERERR- INTx-
    Region 0: Memory at f2c00000 (32-bit, non-prefetchable) [size=32K]
    Region 1: Memory at f3e00000 (32-bit, prefetchable) [size=2M]
    Region 2: Memory at f2800000 (64-bit, non-prefetchable) [size=4M]
    Capabilities: [80] Power Management version 3
        Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
        Status: D3 NoSoftRst+ PME-Enable- DSet=0 DScale=0 PME-
    Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
        Address: 0000000000000000 Data: 0000
    Capabilities: [b0] Null
    Capabilities: [c0] Express (v2) Endpoint, MSI 00
        DevCap: MaxPayload 128 bytes, PhantFunc 0, Latency L0s <1us, L1 <1us
            ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset- SlotPowerLimit 0.000W
        DevCtl: CorrErr+ NonFatalErr+ FatalErr+ UnsupReq-
            RlxDrd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+
            MaxPayload 128 bytes, MaxReadReq 512 bytes
        DevSta: CorrErr+ NonFatalErr+ FatalErr+ UnsupReq- AuxPwr- TransPend-
        LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L1, Exit Latency L1 <8us
            ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp+
        LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommClk+
            ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
        LnkSta: Speed 5GT/s (ok), Width x1 (ok)
            TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
        DevCap2: Completion Timeout: Range B, TimeoutDis+, NoOPPrP-, LTR+
            10BitTagCom-, 10BitTagReq-, OBFF Via message, ExtFmt+, EETLPPrefix+,
        MaxEETLPPrefixes 1
        EmergencyPowerReduction Not Supported, EmergencyPowerReductionInit-

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIe/IBV-PCI-EP-Driver-1.0/src/ti-sample-vfio# ./ti-sample-vfio 4 0 0 60
Starting PCIe RC VFIO test application with parameters:
-----
PCIe Bus Number: 4
PCIe Device Number: 0
PCIe Function Number: 0
PCIe IOMMU Number: 60
Test mode: default test
MSI IRQ number: 1
Iteration number: 10
-----
Using PCI device 0000:04:00.0 in IOMMU group 60
VFIO CHECK EXTENSION VFIO TYPE1 IOMMU: Present
VFIO CHECK EXTENSION VFIO NOIOMMU IOMMU: Not Present
Config region info: region index 0x7, size 0x1000, offset 0x7000000000, cap_offset 0x0, flags 0x3
BAR0 Info: size 0x8000, offset 0x0, flags 0x7
MSI IRQ Info: index: 1, count: 16, flags: 9
RC completed EP initialization
-----
Start COPY test
COPY test passed with 10 loops
-----
Initialize MSI test. Expect 1 distinct MSI IRQs
Expect MSI IRQ nr. 0
MSI test passed
-----
Initialize BARS test
BAR1 Info: size 0x20000, offset 0x1000000000, flags 0x7
BAR2 Info: size 0x40000, offset 0x2000000000, flags 0x7
BAR test passed
-----
RC resets EP
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIe/IBV-PCI-EP-Driver-1.0/src/ti-sample-vfio#

```

## 测试

### 说明

进行测试以验证 TMD5243EVM/TMD564EVM PCIe EP 的 PCIe 入站 ATU 和扩展 BAR 配置是否正常工作。为此，为 PCIe EP 定义了多达 6 种不同的 BAR 配置：

1. 入站地址转换 0：  
此 ATU 配置使用具有 32Kb 不可预取 32 位存储器 BAR 的区域索引 0。
2. 入站地址转换 1：  
此 ATU 配置使用具有 32Mb 可预取 32 位存储器 BAR 的区域索引 1。
3. 入站地址转换 2：  
此 ATU 配置使用具有 512Mb 不可预取 32 位存储器 BAR 的区域索引 2。
4. 入站地址转换 3：  
此 ATU 配置使用具有 128 字节 32 位 I/O BAR 的区域索引 3。
5. 入站地址转换 4：  
此 ATU 配置使用具有 1Kb 32 位 I/O BAR 的区域索引 4。
6. 入站地址转换 5：  
此 ATU 配置使用具有 8Kb 32 位的区域索引 5。I/O BAR。

### 执行：

1. 在 Sysconfig 文件中为 `pcie_enumerate_ep` 应用程序设置所述的 BAR 配置。
2. 检查所需的 PCIe EP BAR 是否基于 Linux 的 RC 硬件上正确配置。启动时，配置的 BAR 可在 PCIe 配置空间中显示为禁用，如下图所示：

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
Subsystem: Cadence Design Systems, Inc. Device 0000
Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAhort- <TAhort- <MAhort- >SERR- <PERR- INTx-
Region 0: Memory at f3800000 (32-bit, non-prefetchable) [disabled] [size=32K]
Region 1: Memory at f3f00000 (32-bit, prefetchable) [disabled] [size=1M]
Region 2: Memory at f3600000 (32-bit, non-prefetchable) [disabled] [size=2M]
Region 3: I/O ports at 3400 [disabled] [size=128]
Region 4: I/O ports at 3000 [disabled] [size=1K]
Region 5: I/O ports at 2000 [disabled] [size=4K]
Capabilities: [80] Power Management version 3
Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
Status: D3 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
Address: 0000000000000000 Data: 0000
Capabilities: [b0] Null
Capabilities: [c0] Express (v2) Endpoint, MSI 00
DevCap: MaxPayload 128 bytes, PhantFunc 0, Latency L0s <1us, L1 <1us
ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset- SlotPowerLimit 0.000W
DevCtl: CorrErr+ NonFatalErr+ FatalErr+ UnsupReq-
Rlxd0rd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+
MaxPayload 128 bytes, MaxReadReq 512 bytes
DevSta: CorrErr- NonFatalErr- FatalErr- UnsupReq- AuxPwr- TransPend-
LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L1, Exit Latency L1 <8us
ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp+
LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommClk+
ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
LnkSta: Speed 5GT/s (ok), Width x1 (ok)
TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
DevCap2: Completion Timeout: Range B, TimeoutDis+, NROPrPrP-, LTR+
10BitTagComp-, 10BitTagReq-, OBFF Via message, ExtFmt+, EETLPPrefix+, MaxEETLPPrefixes 1
EmergencyPowerReduction Not Supported, EmergencyPowerReductionInit-
FRS-, TPHComp-, ExtTPHComp-
AtomicOpsCap: 32bit- 64bit- 128bitCAS-
DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-, LTR+, OBFF Disabled
AtomicOpsCtl: ReqEn-
LnkCtl2: Target Link Speed: 5GT/s, EnterCompliance- SpeedDis-
Transmit Margin: Normal Operating Range, EnterModifiedCompliance- ComplianceSOS-
Compliance De-emphasis: -6dB
LnkSta2: Current De-emphasis Level: -3.5dB, EqualizationComplete-, EqualizationPhase1-
EqualizationPhase2-, EqualizationPhase3-, LinkEqualizationRequest-
  
```

- 带 `testbars` 参数运行 RC 示例应用程序 `ti-sample-vfio`。打开第二个 Linux 终端并检查 PCIe EP 配置空间。当程序在 VFIO 初始化后停止时，相应的 BAR 现在可以启用（未显示为已禁用），如下图所示。



```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# ./ti-sample-vfio 4 0 0 60
bash: ./ti-sample-vfio: No such file or directory
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
Subsystem: Cadence Design Systems, Inc. Device 0000
Control: I/O+ Mem+ BusMaster- SpecCycle- MemWInV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >Abort- <Abort- <MAbort- >SERR- <PERR- INTx-
Region 0: Memory at f3800000 (32-bit, non-prefetchable) [size=32K]
Region 1: Memory at f3f00000 (32-bit, prefetchable) [size=1M]
Region 2: Memory at f3600000 (32-bit, non-prefetchable) [size=2M]
Region 3: I/O ports at 3400 [size=128]
Region 4: I/O ports at 3000 [size=1K]
Region 5: I/O ports at 2000 [size=4K]
Capabilities: [80] Power Management version 3
Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
Status: D3 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
Address: 0000000000000000 Data: 0000
Capabilities: [b0] Null
Capabilities: [c0] Express (v2) Endpoint, MSI 00
DevCap: MaxPayload 128 bytes, PhantFunc 0, Latency L0s <1us, L1 <1us
ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset- SlotPowerLimit 0.000W
DevCtl: CorrErr+ NonFatalErr+ FatalErr+ UnsupReq-
RlxdOrd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+
MaxPayload 128 bytes, MaxReadReq 512 bytes
DevSta: CorrErr- NonFatalErr- FatalErr- UnsupReq- AuxPwr- TransPend-
LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L1, Exit Latency L1 <8us
ClockPM- Surprise- LLActRep- BwNot- ASPMOptComp+
LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommClk+
ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
LnkSta: Speed 5GT/s (ok), Width x1 (ok)
TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
DevCap2: Completion Timeout: Range B, TimeoutDis+, NROPrPrP-, LTR+
10BitTagComp-, 10BitTagReq-, OBFF Via message, ExtFmt+, EETLPPrefix+, MaxEETLPPrefixes 1
EmergencyPowerReduction Not Supported, EmergencyPowerReductionInit-
FRS-, TPHComp-, ExtTPHComp-
AtomicOpsCap: 32bit- 64bit- 128bitCAS-
DevCtl2: Completion Timeout: 50us to 50ms, TimeoutDis-, LTR+, OBFF Disabled
AtomicOpsCtl: ReqEn-
LnkCtl2: Target Link Speed: 5GT/s, EnterCompliance- SpeedDis-
Transmit Margin: Normal Operating Range, EnterModifiedCompliance- ComplianceSOS-
Compliance De-emphasis: -6dB
LnkSta2: Current De-emphasis Level: -3.5dB, EqualizationComplete-, EqualizationPhase1-
EqualizationPhase2-, EqualizationPhase3-, LinkEqualizationRequest-

```

4. 继续执行 RC 示例应用程序 *ti-sample-vfio*。该程序可执行扩展 BAR 测试和输出 BAR 信息，如下图所示。

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIE/IBV-PCIE-1.0/PCIE EP Driver - 1.0/src/ti-sample-vfio# ./ti-sample-vfio 4 0 0 60 testbars
-----
Starting PCIe RC VFIO test application with parameters:
PCIe Bus Number: 4
PCIe Device Number: 0
PCIe Function Number: 0
PCIe IOMMU Number: 60
Test mode: extended BAR test
-----
Using PCI device 0000:04:00.0 in IOMMU group 60
VFIO_CHECK_EXTENSION VFIO_TYPE1_IOMMU: Present
VFIO_CHECK_EXTENSION VFIO_NOIOMMU_IOMMU: Not Present
Config region info: region index 0x7, size 0x1000, offset 0x700000000000, cap_offset 0x0, flags 0x3
BAR0 Info: size 0x8000, offset 0x0, flags 0x7
BAR1 Info: size 0x100000, offset 0x100000000000, flags 0x7
BAR2 Info: size 0x200000, offset 0x200000000000, flags 0x7
BAR3 Info: size 0x80, offset 0x300000000000, flags 0x3
BAR4 Info: size 0x400, offset 0x400000000000, flags 0x3
BAR5 Info: size 0x1000, offset 0x500000000000, flags 0x3
Extended BAR test passed
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIE/IBV-PCIE-1.0/PCIE EP Driver - 1.0/src/ti-sample-vfio#

```

## 5.4 出站 ATU 功能

### 测试

#### 说明：

进行测试以验证 TMD5243EVM/TMD564EVM PCIe EP 的多个 PCIe 出站 ATU 配置是否正常工作。

*pcie\_enumerate\_ep* 应用程序实现了两个出站映射。第一个实现了从内部“PCIE0\_DAT0”窗口到相应 PCIe 地址的 DMA 映射。这个 DMA 出站映射被用于写回从 RC 中接收到的数据。第二个提供 MSI 机制，并将由 RC 配置的本地 MSI 地址映射到相应的 PCIe 地址。

#### 执行：

1. 运行 PCIe EP 应用程序 `pcie_enumerate_ep`。
2. 运行基于 Linux 的 RC 测试应用程序 `ti-sample-vfio`。
3. 检查两个程序的状态。当 `ti-sample-vfio` 等待 PCIe EP 生成的 MSI 中断并检查接收到的数据时，它可以终止而不会出现任何故障，确保 PCIe EP 出站映射的功能正确。

## 5.5 MSI 功能

### 测试

#### 说明：

进行测试以验证 MSI IRQ 是否从 PCIe EP 正确发送到 RC 配置的地址。

#### 执行：

1. 运行 PCIe EP 应用程序 `pcie_enumerate_ep`。
2. 运行基于 Linux 的 RC 测试应用程序 `ti-sample-vfio`。
3. 检查两个程序的状态。当 `ti-sample-vfio` 等待 PCIe EP 在指定地址发送 MSI IRQ 时，它可以终止而不会出现任何故障，从而确保 PCIe EP MSI 机制功能正常。

### 测试

#### 说明：

进行测试以验证是否在 PCIe RC 中正确确定了 PCIe EP 中可用的不同 MSI IRQ (支持多消息) 数量上限，以及是否可以从 RC 请求更少数量的所需 MSI IRQ (启用多消息)。

#### 执行：

1. 运行 PCIe EP 应用程序 `pcie_enumerate_ep`，将 MSI IRQ 的默认数量设置为 16。
2. 在基于 Linux 的 RC 硬件上，检查偏移为 90 时的 PCIe EP MSI 功能。MSI 功能可以通过将计数设置为 1 (总共为 16) 来禁用，如下图所示。
3. 运行基于 Linux 的 RC 测试应用程序 `ti-sample-vfio`。将所配置的 MSI IRQ 的默认数量作为要作为第五个参数进行测试的所需 MSI IRQ 数量传递：

```
sudo ./ti-sample-vfio 9 0 0 19 16
```

4. 按 Enter 继续运行 `ti-sample-vfio`，直到程序在以下状态输出时停止：*Initialize MSI test.Expect 16 distinct MSI IRQs*。
5. 打开第二个 Linux 终端并检查偏移为 90 时的 PCIe EP MSI 功能。MSI 机制可以通过计数 16 来启用，如下图所示。
6. 继续运行 `ti-sample-vfio`。程序可以正常继续，使用 16 个 MSI IRQ 执行扩展 MSI 测试，并在不出现任何故障的情况下结束，如下图所示。
7. 运行基于 Linux 的 RC 测试应用程序 `ti-sample-vfio`。将要测试的 MSI IRQ 数量配置为小于默认数量，例如 8：

```
sudo ./ti-sample-vfio 9 0 0 19 8
```

8. 按 Enter 继续运行 `ti-sample-vfio`，直到程序在以下状态输出时停止：
  - a. *Initialize MSI test. Expect 8 distinct MSI IRQs*。
9. 检查第二个 Linux 终端上偏移为 90 时的 PCIe EP MSI 功能。MSI 功能可以通过计数 8 (总共为 16) 来启用，如下图所示。
10. 继续运行 `ti-sample-vfio`。程序可以正常继续，使用 8 个 MSI IRQ 执行扩展 MSI 测试，并在不出现任何故障的情况下结束，如下图所示。
11. 在基于 Linux 的 RC 硬件上，检查偏移为 90 时的 PCIe EP MSI 功能。MSI 功能可以通过将计数设置为 1 (总共为 16) 来禁用，如下图所示。

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps/Downloads/IBV-PCIE/IBV-PCIE-1.0/PCIE EP Driver - 1.0/src/ti-sample-vfio# ./ti-sample-vfio 4 0 0 60 16
-----
Starting PCIe_RC_VFIO_test application with parameters:
PCIe Bus Number: 4
PCIe Device Number: 0
PCIe Function Number: 0
PCIe IOMMU Number: 60
Test mode: default test
MSI IRQ number: 16
Iteration number: 10
-----
Using PCI device 0000:04:00.0 in IOMMU group 60
VFIO_CHECK_EXTENSION VFIO_TYPE1_IOMMU: Present
VFIO_CHECK_EXTENSION VFIO_NOIOMMU_IOMMU: Not Present
Config region info: region index 0x7, size 0x1000, offset 0x70000000000, cap_offset 0x0, flags 0x3
BAR0 Info: size 0x8000, offset 0x0, flags 0x7
MSI IRQ Info: index: 1, count: 16, flags: 9
RC completed EP initialization
-----
Start COPY test
COPY test passed with 10 loops
-----
Initialize MSI test. Expect 16 distinct MSI IRQs
Expect MSI IRQ nr. 0
Expect MSI IRQ nr. 1
Expect MSI IRQ nr. 2
Expect MSI IRQ nr. 3
Expect MSI IRQ nr. 4
Expect MSI IRQ nr. 5
Expect MSI IRQ nr. 6
Expect MSI IRQ nr. 7
Expect MSI IRQ nr. 8
Expect MSI IRQ nr. 9
Expect MSI IRQ nr. 10
Expect MSI IRQ nr. 11
Expect MSI IRQ nr. 12
Expect MSI IRQ nr. 13
Expect MSI IRQ nr. 14
Expect MSI IRQ nr. 15
MSI test passed
-----
Initialize BARS test
BAR1 Info: size 0x100000, offset 0x10000000000, flags 0x7
BAR2 Info: size 0x200000, offset 0x20000000000, flags 0x7
BAR test passed
  
```

## 测试

### 说明：

测试以验证是否在 TMDS243EVM/TMDS64EVM PCIe EP 上正确禁用了每个矢量屏蔽的 MSI。

### 执行：

1. 在基于 Linux 的 RC 硬件上，通过 Linux 终端检查 PCIe EP MIS 功能。可屏蔽字段可被禁用，如下图所示。

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
  Subsystem: Cadence Design Systems, Inc. Device 0000
  Control: I/O+ Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
  Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
  Region 0: Memory at f3800000 (32-bit, non-prefetchable) [size=32K]
  Region 1: Memory at f3f00000 (32-bit, prefetchable) [size=1M]
  Region 2: Memory at f3600000 (32-bit, non-prefetchable) [size=2M]
  Region 3: I/O ports at 3400 [size=128]
  Region 4: I/O ports at 3000 [size=1K]
  Region 5: I/O ports at 2000 [size=4K]
  Capabilities: [80] Power Management version 3
    Flags: PMEclk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
    Status: D3 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
  Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
    Address: 0000000000000000 Data: 0000
  Capabilities: [b0] Null
  Capabilities: [c0] Express (v2) Endpoint, MSI 00
    DevCap: MaxPayload 128 bytes, PhantFunc 0, Latency L0s <1us, L1 <1us
    ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset- SlotPowerLimit 0.000W
  
```

## 5.6 下行中断功能

### 测试

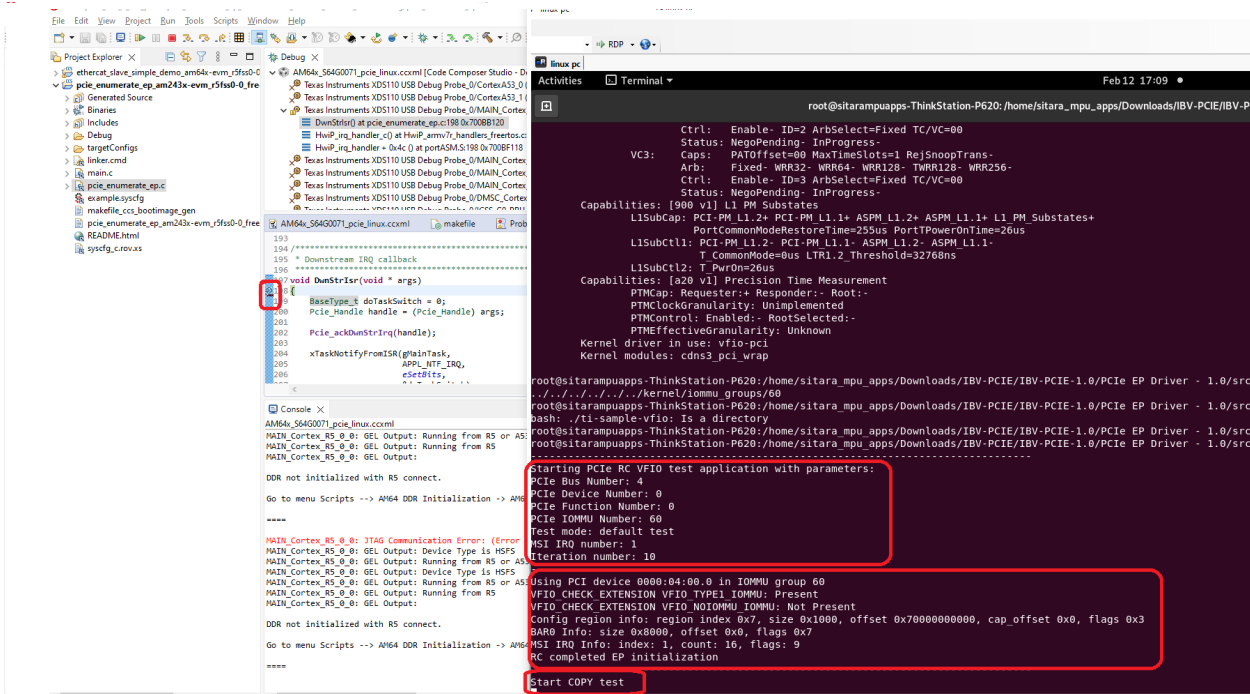
### 说明：

进行测试以验证是否可以在 TMDS243EVM/TMDS64EVM PCIe EP 上配置和触发下游中断功能。

### 执行：



1. 在 `pcie_enumerate_ep` 示例应用程序中，在下游中断服务例程 `void DwnStrlsr(void *args)` 等内 ( 第 196 行 ) 设置一个断点。
2. 在 TMD5243EVM PCIe EP 上执行系统复位、加载并运行 `pcie_enumerate_ep`。
3. 在基于 Linux 的 RC 硬件上使用 GNU 调试器启动 RC 示例应用程序 `ti-sample-vfio`。在第 623 行的 `sendDwnStrlrq` 函数等处设置断点。
4. 运行 `ti-sample-vfio`。当程序在 `sendDwnStrlrq` 处停止时，继续并检查是否在 `pcie_enumerate_ep` 上触发了下游中断
5. 如果测试结果为正，`pcie_enumerate_ep` 可以在 `void DwnStrlsr(void *args)` 处停止，指示成功触发下游中断。



## 5.7 器件电源管理状态功能

### 测试

#### 说明：

进行测试以验证电源管理状态在 TMD5243EVM/TMD564EVM PCIe EP 上是否正常工作。

#### 执行：

1. 在基于 Linux 的 RC 硬件上，使用 Linux 终端检查 PCIe EP 电源管理状态。启动时，PCIe EP 可处于电源管理状态 D0，如下图所示。

```

COM180 x |
DMSC ABI revision 3.1

INFO: Bootloader_runCpu:155: CPU r5f1-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU r5f1-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU m4f0-0 is initialized to 400000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runSelfCpu:217: All done, resetting self ...

PCIe: EP initialized and waiting for link
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
EP is in D0 state
PCIe signaling APPL ready
APPL: pcie ready

```

2. 将 VFIO 驱动程序绑定到 TMD5243EVM PCIe EP。
3. 由于 VFIO 作为内核驱动程序绑定到 PCIe EP，因此 PCIe EP 电源管理状态可能会更改为 D3hot。使用 Linux 终端验证电源管理状态，如下图所示：

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# modprobe vfio-nci
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# echo "17cd 0100" > /sys/bus/pci/drivers/vfio-pci/new_id
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
Subsystem: Cadence Design Systems, Inc. Device 0000
Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
Region 0: Memory at f3800000 (32-bit, non-prefetchable) [disabled] [size=32K]
Region 1: Memory at f3f00000 (32-bit, prefetchable) [disabled] [size=1M]
Region 2: Memory at f3600000 (32-bit, non-prefetchable) [disabled] [size=2M]
Region 3: I/O ports at 3400 [disabled] [size=128]
Region 4: I/O ports at 3000 [disabled] [size=1K]
Region 5: I/O ports at 2000 [disabled] [size=4K]
Capabilities: [80] Power Management version 3
Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
Status: D3 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
Address: 0000000000000000 Data: 0000
Capabilities: [b0] Null
Capabilities: [c0] Express (v2) Endpoint, MSI 00

```

```

sitara_mpu_apps@sitarampuapps-ThinkStation-P620:~$ sudo su
[sudo] password for sitara_mpu_apps:
root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
Subsystem: Cadence Design Systems, Inc. Device 0000
Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
Region 0: Memory at f3800000 (32-bit, non-prefetchable) [disabled] [size=32K]
Region 1: Memory at f3f00000 (32-bit, prefetchable) [disabled] [size=1M]
Region 2: Memory at f3600000 (32-bit, non-prefetchable) [disabled] [size=2M]
Region 3: I/O ports at 3400 [disabled] [size=128]
Region 4: I/O ports at 3000 [disabled] [size=1K]
Region 5: I/O ports at 2000 [disabled] [size=4K]
Capabilities: [80] Power Management version 3
Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
Address: 0000000000000000 Data: 0000
Capabilities: [b0] Null
Capabilities: [c0] Express (v2) Endpoint, MSI 00

```

4. 使用 GNU 调试器打开 *ti-sample-vfio*。在 *main* 内的 *initVFIO* 函数调用后（第 583 行）设置一个断点。运行 *ti-sample-vfio*。
5. 当 *ti-sample-vfio* 程序在 *initVFIO* 和 *initVFIO* 初始化 PCIe EP 器件后停止时，电源管理状态可更改为 D0。使用 Linux 终端验证电源管理状态，如下图所示：

```

COM180 x
INFO: Bootloader_runCpu:155: CPU m4f0-0 is initialized to 400000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runSelfCpu:217: All done, resetting self ...

PCIe: EP initialized and waiting for link
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state
PCIe: signaling APPL halt
APPL: pcie not ready
    
```

## 5.8 功能级复位机制

### 测试

#### 说明：

进行测试以验证是否在 TMD5243EVM/TMD564EVM PCIe EP 上正确禁用了功能级复位 (FLR) 机制。

#### 执行：

1. 在基于 Linux 的 RC 硬件上，检查是否在 PCIe EP 中禁用了 FLR 机制。这可通过下图所示的器件功能进行检查

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
  Subsystem: Cadence Design Systems, Inc. Device 0000
  Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
  Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
  Region 0: Memory at f3800000 (32-bit, non-prefetchable) [disabled] [size=32K]
  Region 1: Memory at f3f00000 (32-bit, prefetchable) [disabled] [size=1M]
  Region 2: Memory at f3600000 (32-bit, non-prefetchable) [disabled] [size=2M]
  Region 3: I/O ports at 3400 [disabled] [size=128]
  Region 4: I/O ports at 3000 [disabled] [size=1K]
  Region 5: I/O ports at 2000 [disabled] [size=4K]
  Capabilities: [80] Power Management version 3
    Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
    Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
  Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
    Address: 0000000000000000 Data: 0000
  Capabilities: [b0] Null
  Capabilities: [c0] Express (v2) Endpoint, MSI 00
    DevCap: MaxPayload 128 bytes, PhantFunc 0, Latency L0s <1us, L1 <1us
      ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset- SlotPowerLimit 0.000W
    DevCtl: CorrErr+ NonFatalErr+ FatalErr+ UnsupReq-
      RlxdOrd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+
      MaxPayload 128 bytes, MaxReadReq 512 bytes
    DevSta: CorrErr- NonFatalErr- FatalErr- UnsupReq- AuxPwr- TransPend-
    LnkCap: Port #0, Speed 5GT/s, Width x1, ASPM L1, Exit Latency L1 <8us
      ClockPM- Surprise- LLActRep- BwNot- ASPM0ptComp+
    LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- CommClk+
      ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
    LnkSta: Speed 5GT/s (ok), Width x1 (ok)
      TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
    
```

## 5.9 传统中断机制

### 测试

#### 说明：



进行测试以验证是否在 TMD5243EVM/TMD564EVM PCIe EP 上正确禁用了传统中断 (INTx) 机制。

执行：

1. 在基于 Linux 的 RC 硬件上，检查 Linux 终端中 PCIe EP 的器件状态。对于有效的测试用例，不能出现信息 *Interrupt: pin A routed to IRQ XXX*。

为了进行比较，下图展示了在基于 Linux 的同一 RC 硬件（其中传统中断机制处于活动状态）上不同 PCIe EP 的器件状态。

```

ibv@debian:~$ sudo lspci -vvv -s 00:02.0
00:02.0 VGA compatible controller: Intel Corporation CometLake-S GT2 [UHD Graphics 630] (rev 03) (prog-if 00 [VGA controller])
    DeviceName: onboard - Video
    Subsystem: Micro-Star International Co., Ltd. [MSI] CometLake-S GT2 [UHD Graphics 630]
    Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx+
    Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
    Latency: 0 Cache Line Size: 64 bytes
    Interrupt: pin A routed to IRQ 147
    I/O group: 0
    Region 0: Memory at 600000000 (64-bit, non-prefetchable) [size=16M]
    Region 2: Memory at 400000000 (64-bit, prefetchable) [size=256M]
    Region 4: I/O ports at 4000 [size=64]
    Expansion ROM at 000c0000 [virtual] [disabled] [size=128k]
    Capabilities: [40] Vendor Specific Information: Len=0c <?>
    Capabilities: [70] Express (v2) Root Complex Integrated Endpoint, MSI 00
        DevCap: MaxPayload 128 bytes, PhantFunc 0
            EXTTag- RBE+ FLReset+
        DevCtl: CorrErr- NonFatalErr- FatalErr- UnsupReq-
    
```

## 5.10 MSI-X 功能

测试

说明：

进行测试以验证是否在 TMD5243EVM/TMD564EVM PCIe EP 上正确禁用了 MSI-X 功能。

执行：

1. 在基于 Linux 的 RC 硬件上，检查是否在 PCIe EP 中禁用了 MSI 功能。由于 MSI-X 功能位于偏移 B0 处，如果禁用，此字段可以包含 *NULL*，如下图所示。

```

root@sitarampuapps-ThinkStation-P620:/home/sitara_mpu_apps# lspci -vvv -s 04:00.0
04:00.0 Gameport controller: Cadence Design Systems, Inc. Device 0100 (rev 03) (prog-if 10 [Extended])
    Subsystem: Cadence Design Systems, Inc. Device 0000
    Control: I/O- Mem- BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR- FastB2B- DisINTx-
    Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-
    Region 0: Memory at f3800000 (32-bit, non-prefetchable) [disabled] [size=32K]
    Region 1: Memory at f3f00000 (32-bit, prefetchable) [disabled] [size=1M]
    Region 2: Memory at f3600000 (32-bit, non-prefetchable) [disabled] [size=2M]
    Region 3: I/O ports at 3400 [disabled] [size=128]
    Region 4: I/O ports at 3000 [disabled] [size=1K]
    Region 5: I/O ports at 2000 [disabled] [size=4K]
    Capabilities: [80] Power Management version 3
        Flags: PMEclk- DSI- D1+ D2- AuxCurrent=0mA PME(D0+,D1+,D2-,D3hot+,D3cold-)
        Status: D0 NoSoftRst+ PME-Enable- DSel=0 DScale=0 PME-
    Capabilities: [90] MSI: Enable- Count=1/16 Maskable- 64bit+
        Address: 0000000000000000 Data: 0000
    Capabilities: [b0] Null
    Capabilities: [c0] Express (v2) Endpoint, MSI 00
    
```

## 5.11 热复位机制

测试

说明：

进行测试以验证 TMD5243EVM/TMD564EVM PCIe EP 上的热复位机制是否正常工作。

执行：

1. 在 PCIe EP 示例应用程序 *pcie\_enumerate\_ep* 中，在函数 *HotResetIsr* 内（例如，第 178 行）设置一个断点。由于此函数是相应机制的中断服务例程，因此程序可以在此处停止。

2. 在基于 Linux 的 RC 硬件上运行 *ti-sample-vfio*。
3. 检查 *pcie\_enumerate\_ep* 是否在下图所示的断点处停止。
4. 继续运行 *pcie\_enumerate\_ep* 并通过 SER\_TER 检查其状态。状态可以是消息 *PCIe: hot reset detected* , 如下图所示。

```

COM180 x
INFO: Bootloader_runCpu:155: CPU r5f1-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU r5f1-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU m4f0-0 is initialized to 400000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_runCpu:155: CPU a530-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_loadSelfCpu:207: CPU r5f0-0 is initialized to 800000000 Hz !!!
INFO: Bootloader_loadSelfCpu:207: CPU r5f0-1 is initialized to 800000000 Hz !!!
INFO: Bootloader_runSelfCpu:217: All done, resetting self ...

PCIe: EP initialized and waiting for link
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: lost PCIe link
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: power state entry
EP is in D3hot state
PCIe: signaling APPL halt
APPL: pcie not ready
EP is in D0 state
PCIe: signaling APPL ready
APPL: pcie ready
PCIe: lost PCIe link
PCIe: hot reset detected
PCIe: signaling APPL halt
APPL: pcie not ready
PCIe: link detected
PCIe link parameter: PCIe Gen2 with 5.0 GT/s speed, number of lanes: 1
PCIe: signaling APPL ready
APPL: pcie ready
..
  
```

## 6 Windows 示例驱动程序验证

本章定义并指定 Windows 示例驱动程序的测试。仅测试了经过精简的部分 EP 功能，以确保之前在 Linux 上测试的功能可在 Windows 上以类似方式运行。以下测试规范假定环境为 AM24\_WIN。

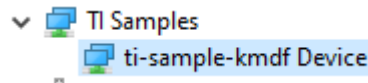
### 测试

#### 说明：

测试以验证 Windows KMDF 驱动程序的功能。

#### 执行：

1. 通过在 Windows 设备管理器中搜索 ti-sample-kmdf 设备，验证是否已加载 pcie\_enumerate\_ep 示例 EP 的 Windows 驱动程序：



2. 使用管理员权限打开命令提示符，并运行 ti-sample-console.exe 应用程序：

```
C:\Users\...Downloads\PCie EP Driver - 1.0-updated-output\PCie EP Driver - 1.0\src\ti-sample-kmdf\x64\Release>ti-sample-console.exe
-----
Starting PCIe RC KMDF test application
-----
Opening windows kernel mode driver \\.\SampleTI
-----
Start COPY test
IOCTL_TISAMPLEKMDF_TEST_DMA returned data, verifying...
COPY test passed
-----
Start MSI test
IOCTL_TISAMPLEKMDF_TEST_MSI returned, result: 0000ffff
MSI test passed
-----
Start Bar1/2 test
IOCTL_TISAMPLEKMDF_TEST_BARS returned, result: 00000001
BAR test passed
-----
Closing windows kernel mode driver
KMDF test application done
-----
C:\Users\...Downloads\PCie EP Driver - 1.0-updated-output\PCie EP Driver - 1.0\src\ti-sample-kmdf\x64\Release>
```

3. 验证所有测试均已通过，并且没有出现如上所示的错误。
4. 验证 EP 的 UART 上的输出是否与预期输出匹配，从而指示 DMA 测试、MSI 测试（发送 16 个不同的中断）和 BAR 测试完成：

```

DMA test done
Send MSI IRQ nr. 0
Send MSI IRQ nr. 1
Send MSI IRQ nr. 2
Send MSI IRQ nr. 3
Send MSI IRQ nr. 4
Send MSI IRQ nr. 5
Send MSI IRQ nr. 6
Send MSI IRQ nr. 7
Send MSI IRQ nr. 8
Send MSI IRQ nr. 9
Send MSI IRQ nr. 10
Send MSI IRQ nr. 11
Send MSI IRQ nr. 12
Send MSI IRQ nr. 13
Send MSI IRQ nr. 14
Send MSI IRQ nr. 15
MSI test done
BAR test done

```

## 理由

ti-sample-console 应用程序调用 ti-sample-kmdf 驱动程序并执行以下测试步骤：

- COPY 测试，其中 KMDF 驱动程序先前写入 EP Bar0 存储器的数据会正确发送回 Windows 主机的 DMA 缓冲区
- MSI 测试，其中 EP 触发每个已启用的 MSI 向量（多消息使能）一次。KMDF 驱动程序在 EP 中触发此测试，并等待接收所有 MSI 向量。如果测试返回，则表示已接收所有配置的向量。此外，还会显示接收到的 MSI 向量的位掩码（例如，结果 0000ffff 表示已经接收到向量 0-15）。
- Bar1/2 测试，其中 KMDF 驱动程序用已知模式填充 EP 的 bar1 和 bar2，然后在 EP 中触发测试执行。EP 验证 Bar1 和 Bar2 中的已知模式，成功时会将 MSI 发送回 RC。如果测试返回，则验证成功。

## 7 参考资料

- 德州仪器 (TI) , [AM64x MCU+ SDK : PCIE](#)
- 德州仪器 (TI) , [PCIe 端点 — Processor SDK AM64X 文档](#)
- 德州仪器 (TI) , [PCIe 根复合体 — Processor SDK AM64X 文档](#)
- 德州仪器 (TI) , [TMDS243EVM 评估板](#)
- 德州仪器 (TI) , [TMDS64EVM 评估板](#)



## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2024，德州仪器 (TI) 公司