



摘要

DRA8x 和 TDA4x 处理器系列基于逐步发展的 Jacinto™ 7 架构，这种架构的目标应用为 ADAS、网关和自动驾驶汽车 (AV)。它们利用了 TI 在 ADAS 处理器市场十余年优势地位中积累的广泛市场知识。Jacinto™ 处理器配备旨在提供可靠通信的集成式以太网交换机。它们统称为 CPSW_nG，其中 n 代表可用的硬件端口数。

此外，Jacinto™ 7 器件为用户和关键系统应用提供域隔离。拥有强大 A 内核的主域满足用户空间应用需要，拥有专用 R 和 M 内核的 MCU 域则提供安全功能。每个域通常配备一个专用的 CPSW_nG 交换机，其应用域拥有更多的端口。每个 CPSW_nG 可提供 x-1 个外部端口，这些端口可以连接不同的器件和外部交换机。

考虑到汽车硬件对成本敏感的性质，器件上通常省去 PHY，使用一种叫 MAC2MAC 的配置直接相互连接交换机，确保减少 BOM 并加快连接速度。MAC2MAC 是 TI Jacinto™ 7 处理器中广泛使用的应用场景，本文档将以各种示例和用例解释这种场景，同时提供更多的技术细节。

出于此讨论目的并为了方便阅读，本应用手册使用 DRA829/TDA4 SoC；但任何其他 Jacinto™ 7 处理器适用同样的原理。

本文档所讨论的工程配套资料和源代码可从以下 URL 下载：<https://www.ti.com/cn/lit/zip/SPRAD07>

内容

1 以太网和 CPSW 简介.....	2
1.1 以太网接口.....	2
1.2 以太网集成.....	2
1.3 CPSW.....	3
1.4 MAC2MAC.....	3
2 JACINTO7 MAC2MAC 解决方案.....	4
2.1 应用要求.....	4
2.2 RGMII 解决方案.....	8
2.3 SGMII 解决方案.....	9
3 MAC2MAC 实现.....	10
3.1 RGMII 实现.....	10
3.2 SGMII 实现.....	11
3.3 MAC2MAC 调试.....	11
4 总结.....	16
5 参考文献.....	16

商标

Jacinto™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

1 以太网和 CPSW 简介

1.1 以太网接口

以太网接口包含三个部分：CPU、MAC 和 PHY，如图 1-1 所示。通常，DMA 控制器是 SoC 的一部分，其中虚线表示处理器经常使用 DMA 访问数据。

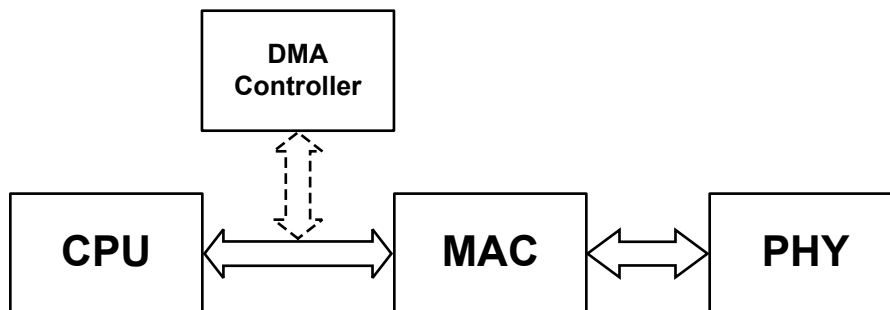


图 1-1. 以太网接口

CPU 是中央处理器的缩写。以太网 MAC 是媒体访问控制的缩写，属于 OSI 模型的数据链路层。PHY 是物理接口器件，属于 OSI 模型中的物理层。MAC 使用媒体独立接口 (MII) 与 PHY 连接。MII 的各种类型有 MII、GMII、RMII、RGMII、SGMII 等。MII 类型因 SoC 和端口而异。有关详细信息，请参阅器件的技术参考手册。

1.2 以太网集成

PHY 提供 MAC 与网络之间的电气接口。考虑到芯片面积和数字与模拟信号的混合，以下结构是可能的：

1. CPU 集成了以太网 MAC 和 PHY（参阅图 1-2）。

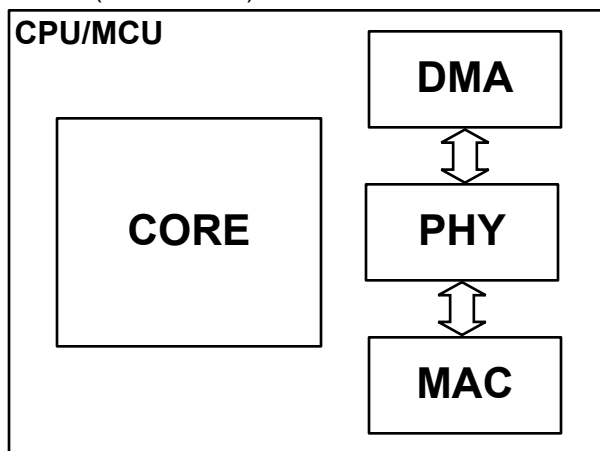


图 1-2. CPU 集成了 MAC 和 PHY

2. CPU 集成了以太网 MAC，PHY 在外部（参阅图 1-3）。

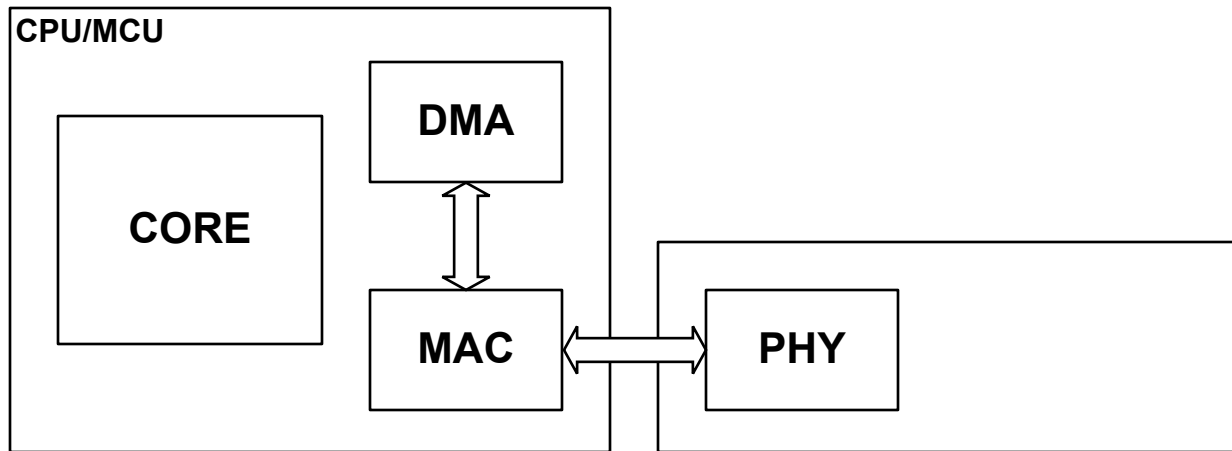


图 1-3. CPU 集成了 MAC

3. MAC 和 PHY 都在外部 (参阅图 1-4)。

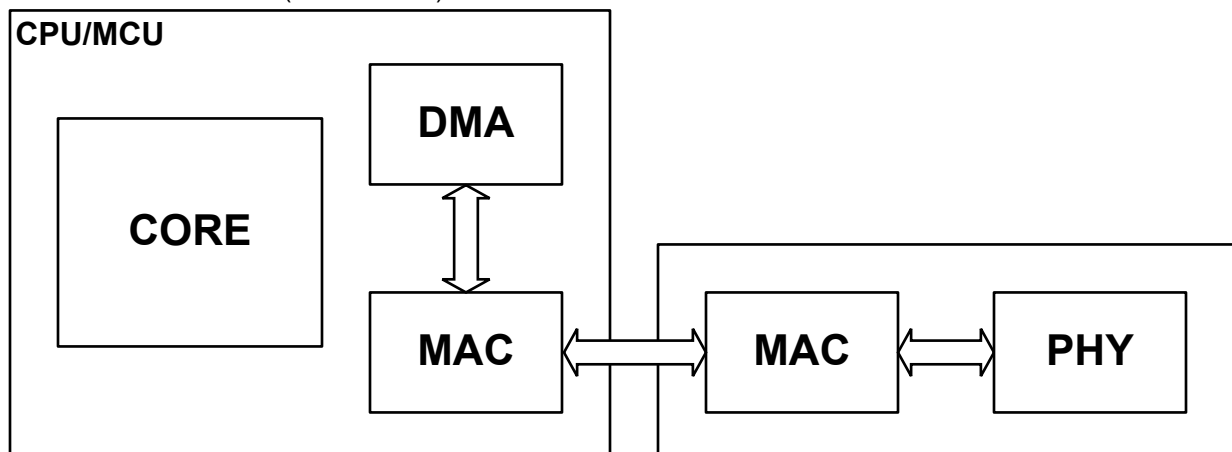


图 1-4. 独立的 MAC 和 PHY

在上述三种方案中，第二种最受欢迎，很少有 CPU 内核带有集成的 PHY。大多数 CPU 内核仅包含集成的 MAC。

1.3 CPSW

以 DRA829/TDA4 为例，SoC 附带集成式 CPSW9G 和 CPSW2G 接口。CPSW 是通用平台以太网交换机的缩写。在 DRA829/TDA4 架构中，CPSW9G 在主域中，CPSW2G 在 MCU 域中。CPSW9G 有 9 个端口：8 个是物理端口，1 个是与 CPU 内核连接的软件端口。同理，CPSW2G 有两个端口：一个用于 CPU 核心，另一个是物理端口。

这两个交换机的所有物理端口都可以使用 MAC2MAC 轻松连接到另一设备或交换机。

1.4 MAC2MAC

前面已经说明，MAC 使用 PHY 连接到网络。但当对象是要连接到受控环境 (例如 PCB) 中的另一个 MAC 时，PHY 通常是多余的，而且可能增加工程成本。在这种情况下，Rx、Tx 和时钟引脚通常使用缓冲器直接连接。这叫做 MAC2MAC 连接。

2 JACINTO7 MAC2MAC 解决方案

2.1 应用要求

由于 Jacinto 7 采用多核异构架构，因此它不仅集成 Arm A72 内核，还集成 TI 的 C7000 和 C6000 DSP、Arm R5F MCU 内核等。这种多核异构架构不仅性能出众，而且灵活性高且便于设计。

图 2-1 显示 CPSW 的架构。从 MAC 接口类型来看，一般的 RGMII (简化千兆位媒体独立接口) 和 SGMII (串行 GMII) 可以支持 1000M 以太网。

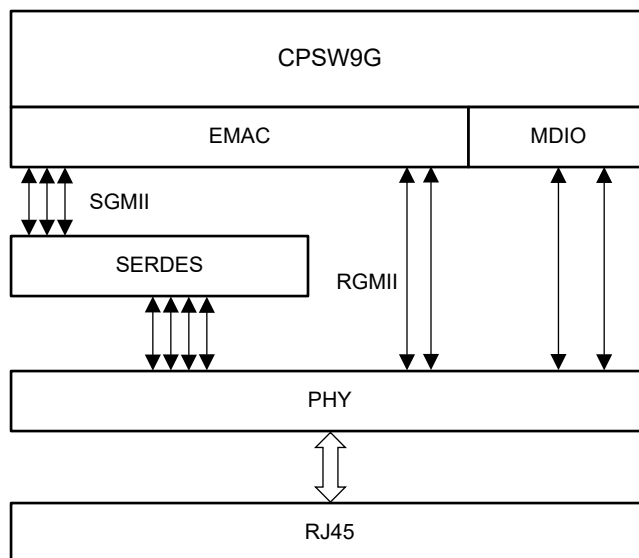


图 2-1. CPSW 架构

RGMII 代表简化千兆位媒体独立接口，即引脚数量减少。时钟频率仍然是 125MHz，但 TX/RX 数据位宽从 8 位更改为 4 位。为了在减少数据线路后保持 1Gbps 传输速率不变，在时钟的上升沿和下降沿都发送数据。RGMII 还兼容 100Mbps 和 10Mbps 速率。为此，基准时钟频率分别为 25MHz 和 2.5MHz。图 2-2 显示典型的 RGMII 应用连接。

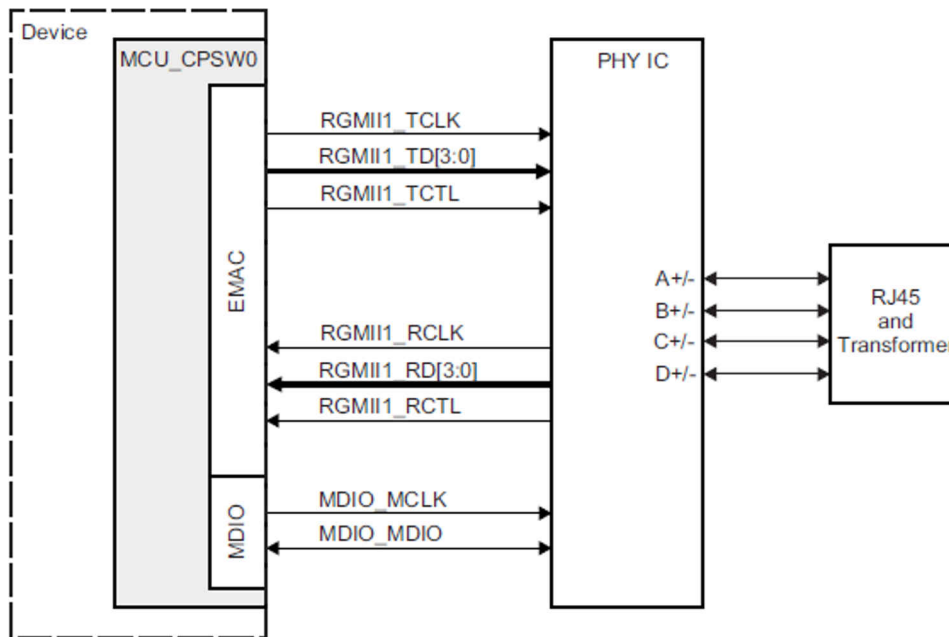


图 2-2. RGMII 连接

SGMII 代表串行 GMII，每当需要较高的速度和较少的引脚时会用到。代价是增加了复杂度，需要串行器和解串器模块，该模块简称为 SerDes。它有一对差分信号线，分别用于发送和接收信号。时钟频率为 625MHz，在时钟信号的上升沿和下降沿采样。基准时钟 RX_CLK 可由 PHY 提供，但它是可选的，主要在没有时钟时用于 MAC 侧。正常情况下不使用 RX_CLK，传输和接收可以从数据中恢复时钟。大多数 MAC 芯片的 SGMII 接口都可以配置为 SerDes 接口（物理兼容，只需要配置寄存器），直接连接到其他模块，不需要 PHY 层芯片，时钟频率仍然是 625MHz。图 2-3（参考 J721E TRM）显示如何配置有 SerDes 接口的 CPSW SGMII。

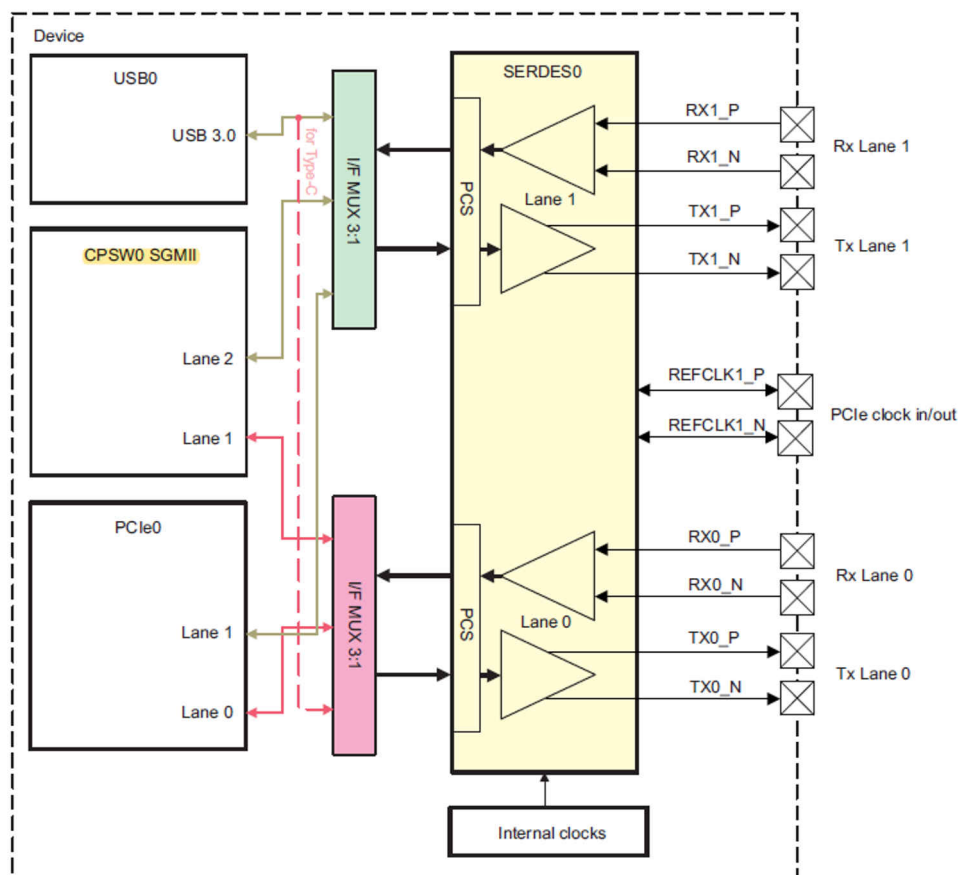


图 2-3. Serdes0 概述

不同的 SGMII 端口对应不同的 SERDES 通道。表 2-1 显示了这种关系。

表 2-1. SGMII 端口 SERDES

SGMII 端口	SERDES - 通道
端口 1	SERDES 0 - 通道 0
端口 2	SERDES 0 - 通道 1
端口 3	SERDES 1 - 通道 0
端口 4	SERDES 1 - 通道 1
端口 5	SERDES 4 - 通道 0
端口 6	SERDES 4 - 通道 1
端口 7	SERDES 4 - 通道 2
端口 8	SERDES 4 - 通道 3

因此，MAC2MAC 互连解决方案通常有两种方式，即基于 RGMII 和基于 SGMII。MAC2MAC 可以有许多应用场景，这里说明其中 3 种：

1. Jacinto7 分为两部分，MCU 域和主域。CPSW2G 通常分配给 MCU 域，CPSW9G 通常由主域内核 A72 和 R5F 使用。鉴于功能安全要求，MCU 域通常使用 AUTOSAR OS。除了内部数据路径，从外部看，通过 MAC2MAC 连接两个交换机也非常方便。图 2-4 显示系统架构。

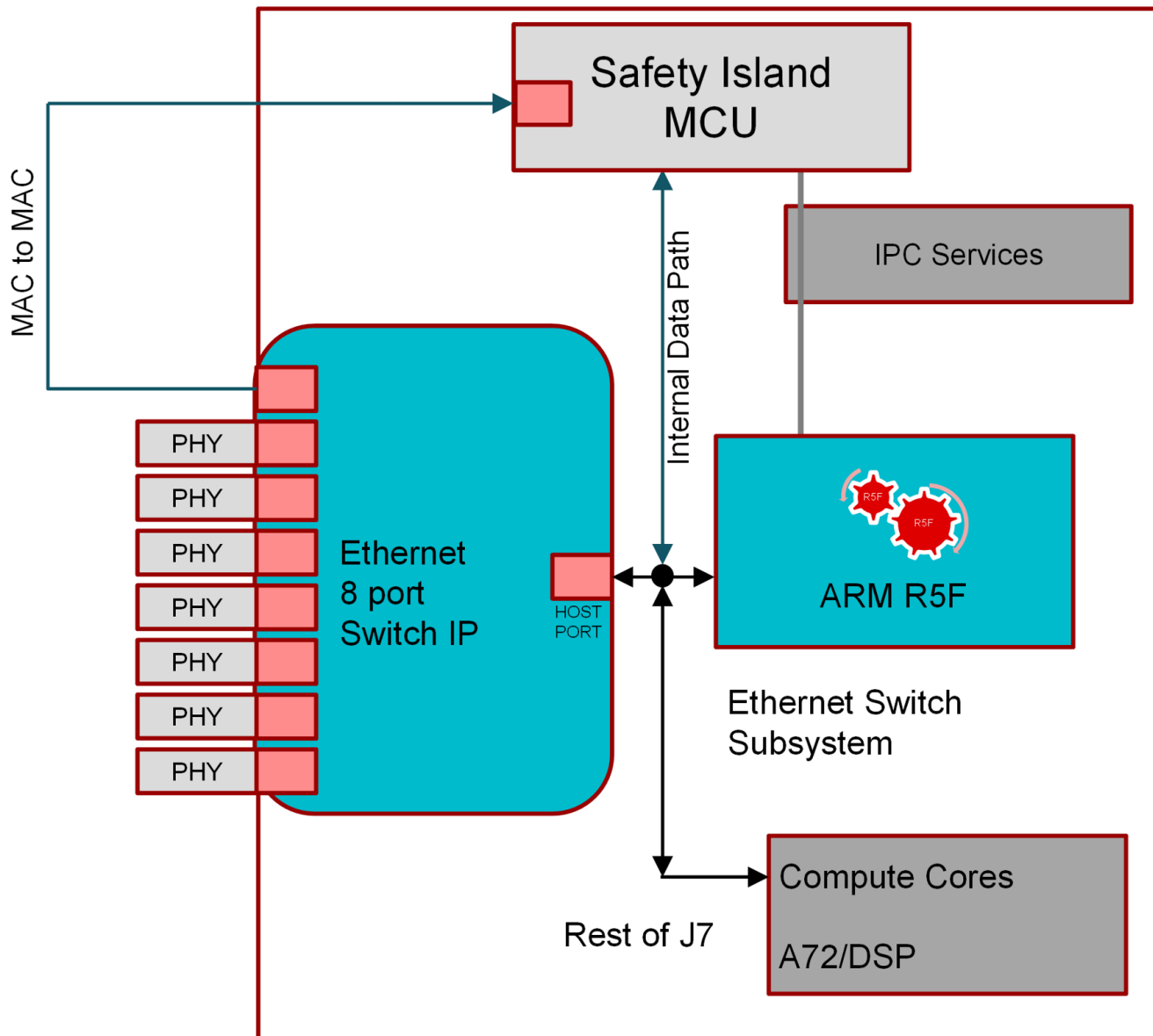


图 2-4. CPSW9G 和 CPSW2G MAC2MAC

- 当单个 Jactinto7 处理器的性能不能满足系统要求时，将多个 Jactinto7 SOC 集成在一块电路板上也是非常方便的。在这种情况下，有助于通过 MAC2MAC 连接 CPSW 交换机。图 2-5 显示系统方框图。

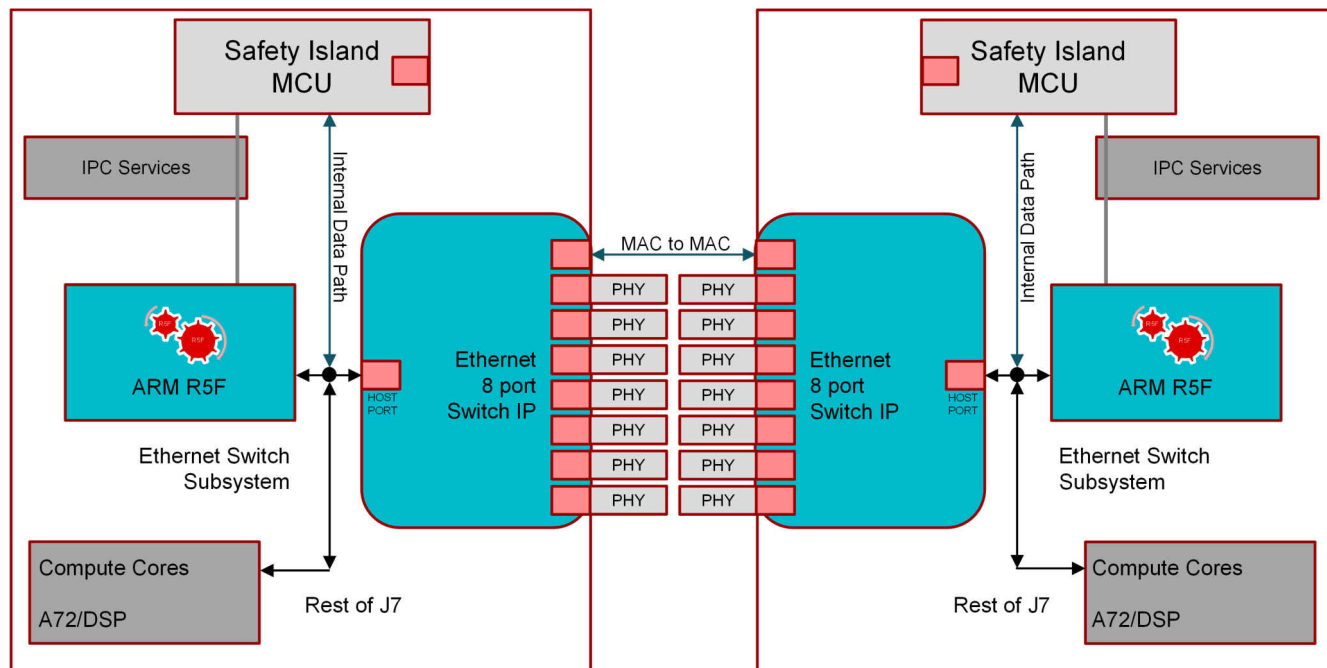


图 2-5. SOC 与 SOC 之间的 MAC2MAC

3. MAC2MAC 可用于连接外部交换机或其他 SoC 上的 MAC。它有两个目的，第一是增加端口数，第二是可用互连 SoC。图 2-6 显示系统方框图。

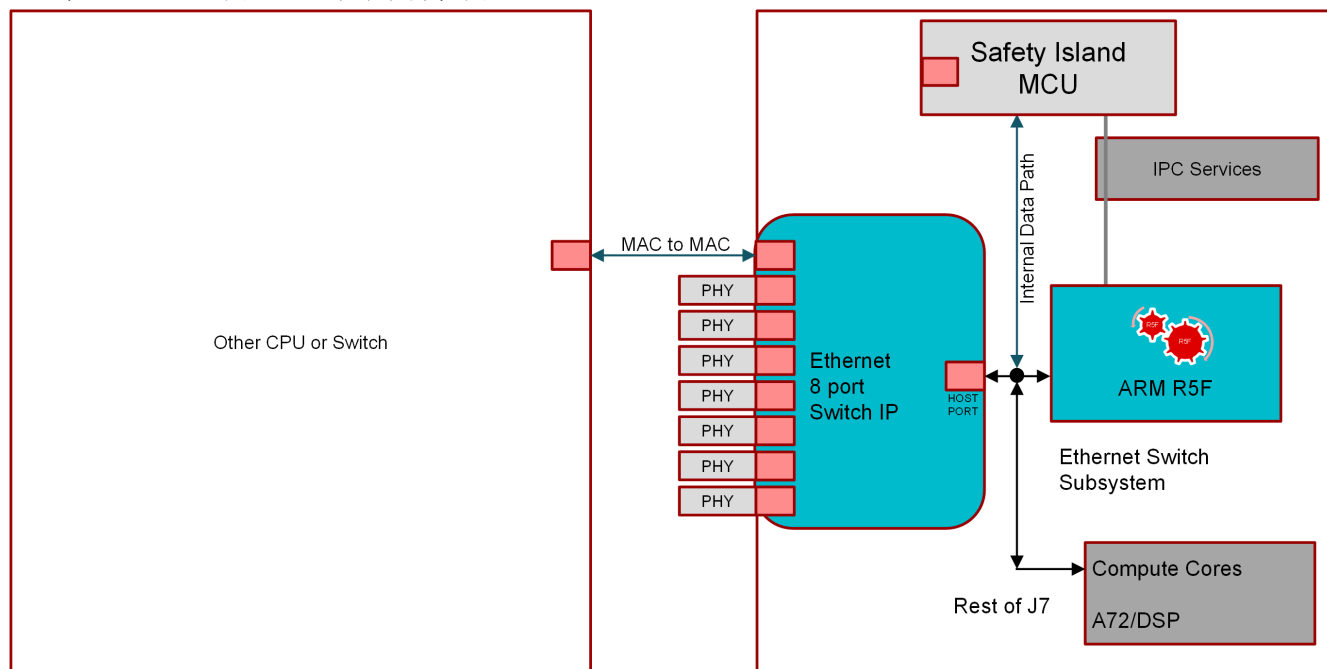


图 2-6. 外部 Soc 连接到 TDA4VM

2.2 RGMII 解决方案

现在来看使用 CPSW2G 和 CPSW9G 互连的第一个示例。内部 MAC 和 MAC 通过 RGMII 互连。RGMII 比起 SGMII 相对简单，MAC 直接连接到 MAC。图 2-7 中显示了该连接。

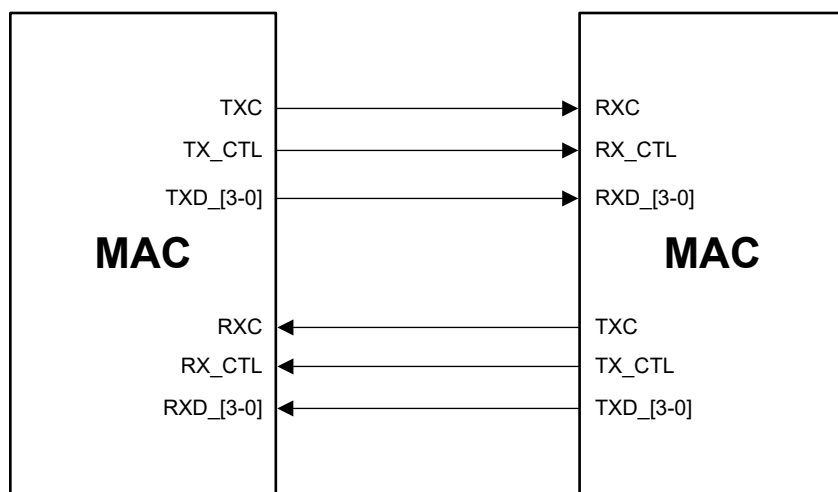


图 2-7. RGMII MAC2MAC

2.3 SGMII 解决方案

这里提供了一个 SGMII 互连示例。两个 TDA4VM 通过 SGMII MAC2MAC 互连。与 RGMII 相比，SGMII 需要额外配置 SERDES。系统方框图如图 2-8 中所示。

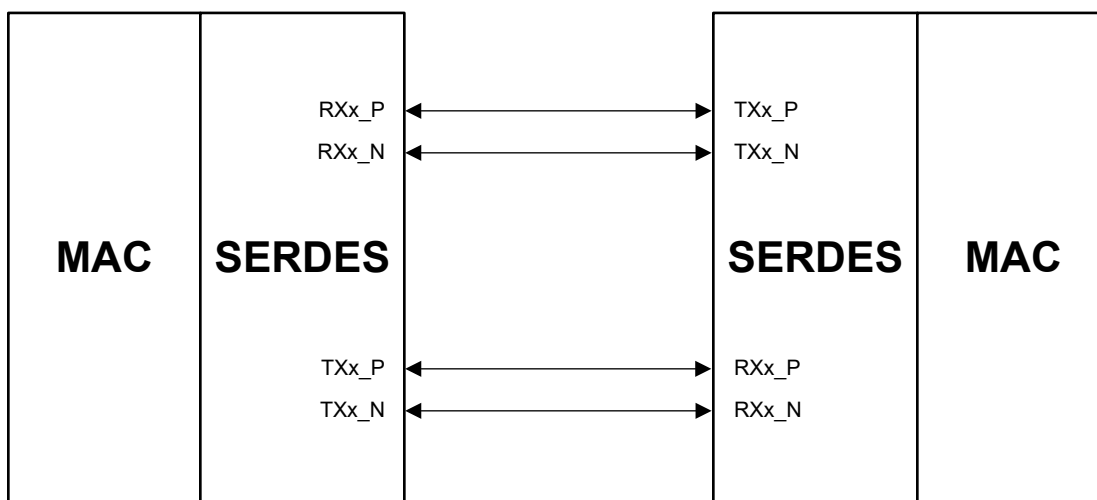


图 2-8. SGMII MAC2MAC

3 MAC2MAC 实现

3.1 RGMII 实现

下面，我们讨论之前提到的 CPSW2G 和 CPSW9G 互连用例。CPSW2G 由 A72 控制，CPSW9G 由 Main R5F 控制，CPSW2G 和 CPSW9G 通过 MAC2MAC 链路连接。配置模式为 RGMII 1000M、全双工。

3.1.1 CPSW 2G 更改

由于 CPSW 2G 有本机 Linux 驱动程序（本文不涉及 QNX 用例），所以它有一个对应的设备树节点，也就是指 PHY。鉴于 MAC2MAC 连接中不涉及 PHY，我们将利用固定链接功能在 init 配置链接属性。在这种情况下，寻找链路状态变化通知的 PHY 状态机逻辑也将禁用。

下面显示了此方面的更改。参考补丁 (0001-*.patch for ti-processor-sdk-linux-j7-evm-07_03_00_05) 可从工程配套资料和源代码链接下载。

备注

在 ti-processor-sdk-linux-j7-evm-07_03_00_05 上应用 0001-*.patch。在应用补丁后，重建使用 “make linux-dtbs” 的 dtb。接下来，将新的 dtb 文件更新到引导分区。

对于默认 ti-processor-sdk-linux-j7-evm-07_03_00_05，我们将 CPSW2G 配置为 PHY 模式。所以，我们需要移除 MDIO 和 PHY 配置。

3.1.2 CPSW 9G 更改

CPSW 9G 由主 R5F 2_0 上运行的以太网固件控制，因此所有更改必须在以太网固件和 ENET-LLD 中完成。要求的主要更改都在端口开放序列中，后者用于获取端口参数并配置 PHY。参考补丁 (0002-*.patch for ti-processor-sdk-rtos-j721e-evm-07_03_00_07) 可从工程配套资料和源代码链接下载。

备注

在 ti-processor-sdk-rtos-j721e-evm-07_03_00_07 上应用 0002-*.patch。应用补丁后，重建使用 “make pdk ; make ethfw; make vision_apps” 的 PDK/ETHFW/VISION_APPS。接下来，运行 “make linux_fs_install” 将映像更新到 targetfs。

1. 在 *EnetPer_PortLinkCfg::macPort* 中设置端口号。
2. 通过 *EnetPer_PortLinkCfg::mii* 的层、子层和变量字段设置 MAC 端口接口（RMII、RGMII、SGMII 等）。
3. 使用 *CpswMacPort_initCfg()* 初始化 MAC 配置参数，手动覆盖不同于默认参数的任何参数。
4. 将 PHY 地址设置为 *ENETPHY_INVALID_PHYADDR*，以指示这是没有 PHY 的连接。
5. 在 *EnetPer_PortLinkCfg::linkCfg* 中设置链路速度和双工配置，使之与 MAC 搭档端口的配置匹配。速度和双工必须是固定值，不能是 *ENET_SPEED_AUTO* 或 *ENET_DUPLEX_AUTO*，因为它们用于自动协商，这与 MAC 到 MAC 模式无关。
6. 完成所有 MAC 和 PHY 配置后，可通过调用 *ENET_PER_IOCTL_OPEN_PORT_LINK* 命令打开端口。
7. 如果上述所有步骤都成功，没有任何错误，则 Enet 驱动程序、以太网外设和 MAC 端口已成功打开。
8. 检测到链路接通时，该应用应使用 *CPSW_ALE_IOCTL_SET_PORT_STATE_IOCTL* 命令将 ALE 端口状态设置为正向状态，并通过调用 *ENET_HOSTPORT_IOCTL_ENABLE* 命令启用 CPSW 的主机端口。

所述更改源于 ENET-LLD 视角，但在以太网固件环境下，这些步骤分别在以太网固件和 Enet LLD utils 库中实施。下面介绍了以太网固件和 Enet LLD 如何互动以实现端口配置。

在以太网固件中配置端口时，发生如下序列：

1. *EthFw_initLinkArgs()* 由 Enet MCM（多客户端管理器）调用，以获取要用于给定端口的端口配置参数。
2. 以太网固件填充 *EthFw_initLinkArgs()* 中的常见参数，例如 CPSW MAC 端口默认配置、速度/双工。
 - a. MAC 到 MAC 模式的速度和双工当前设置为 1Gbps 全双工，需要时可更改它。
3. 以太网固件调用 *EnetBoard_setPhyConfig()* 以设置接口类型和 PHY 配置参数（例如 PHY 地址）。
4. *EnetBoard_setPhyConfig()* 是一个 Enet LLD 电路板 utils 函数。
 - a. 需要在此函数中对 MAC 到 MAC 进行更改。

5. `EnetBoard_setPhyConfig()` 中提供参考补丁及更改，其中 MAC 端口 4 是 MAC 到 MAC 连接需要使用的端口。参考补丁 (0003-*.patch) 可从工程配套资料和源代码链接下载。
6. 完成所有更改后，再次编译以太网固件并加载。如果使用 Linux，则将固件 (`app_remoteswitchcfg_server_strip.xer5f`) 复制到 `rootfs` 分区中的 `/lib/firmware/ethfw` 并重新引导。

3.2 SGMII 实现

这是从 TI 的 CPSW9G 到 CPSW9G 互连两个 TDA4VM 的示例。如图 12 所述，通过 SGMII MAC2MAC 连接一个 TDA4VM CPSW9G 和另一个 TDA4VM CPSW9G。CPSW9G 由以太网固件控制，因此唯一需要的更改要在这里完成。从这个意义上说，除了 MAC 端口接口被配置为 RGMII 的部分之外，所需的大多数更改都与 RGMII 需要的更改类似（如上所述）。需要额外执行的唯一更改是：参考补丁 (0004-*.patch for ti-processor-sdk-rtos-j721e-evm-07_03_00_07) 可从工程配套资料和源代码链接下载。

备注

请在 `ti-processor-sdk-rtos-j721e-evm-07_03_00_07` 上应用 0004-*.patch。应用补丁后，应重建使用 “`make pdk ; make ethfw; make vision_apps`” 的 PDK/ETHFW/VISION_APPS。然后，需要运行 “`make linux_fs_install`” 将映像更新到 `targetfs`。

1. 设置 SERDES 的时钟和设置 SERDES
 2. 按照表 1 映射 SERDES 通道和 CPSW 端口。SGMII 端口 SERDES 表
- 必须注意确保 SERDES 通道不与 PCIe 或 USB 冲突。

3.3 MAC2MAC 调试

启动过程中可能出现的问题。按照发生的可能性排序，问题可分为三类：

1. 软件和配置问题
2. Pinmux 问题
3. 硬件问题

3.3.1 软件和配置问题

当以下任何一个或多个问题发生时，就会出现：

1. 以太网固件或 ENET-LLD 的更改没有正确完成
2. Linux DT 更改没有正确完成
3. SERDES 多路复用没有按照建议正确完成（仅适用于 SGMII）
4. RGMII 连接应注意通过 `ENETn_CTRL` 寄存器设置建议的 RGMII 延迟。

它们通常表现为软件初始化序列错误。例如，某个 IOCTL 可能失败或初始化例程可能返回错误。在大多数情况下，它会导致完全失败和端口不工作。

3.3.2 调试软件和配置问题

要调试此类问题，首先要验证 Linux 引导日志和以太网固件日志，确保两者中都没有关键字 “error”。

以太网固件日志可在第三个主 UART 实例上找到，如果无法访问它，也可以通过运行命令 “`cat /sys/kernel/debug/remoteproc/remoteproc*/trace0`” 从 Linux 终端获取日志。

以太网固件日志示例 (`ethfw_freertos_log.txt` for `ti-processor-sdk-rtos-j721e-evm-07_03_00_07`) 可从工程配套资料和源代码链接下载。注意，它显示：

1. ENET 注册成功
2. 远程内核初始化成功
3. Phy 初始化和链路接通
4. Rx 流和 VLAN 路由配置

根据启用的配置不同，日志可能会有所不同，但这些基本步骤保持不变。

也可通过运行 `pdk/packages/ti/drv/enet/tools/debug_gels` 中的 GEL 文件转储各种寄存器的内容

下面提供了有关 GEL 文件的详细信息：

1. **cpsw_ale_print_table.gel**：用于打印 ALE 配置。可用于检查 ALE 配置是否成功
2. **cpsw_enetctrl_cfg.gel**：这对于 SGMII 和 RGMII 配置的调试很重要。它从 ENET 控制寄存器打印信息，这些寄存器控制是否配置 RGMII 或 SGMII 模式。
3. **cpsw_mac.gel**：此 GEL 文件用于转储与双工、流启用等相关的信息。对 MAC 到 MAC 配置有用
4. **cpsw_mdio_config.gel**：它用来打印 MDIO 配置，与 MAC 到 MAC 调试无关
5. **cpsw_print_reg.gel**：它是所有其他 GEL 的超集，可用于打印所有 CPSW 寄存器。
6. **cpsw_sgml2_diag.gel**：它用于打印 SERDES 配置，是调试 SGMII 的重要 GEL 文件。用户应运行此文件，验证 SERDES 配置符合配置设置。

cpsw_stats.gel：用于转储统计数据，所有端口（包括主机端口）接收/传输的帧数。用户应运行此文件，检查是否有任何错误帧，以及是否正确接收/发送所有数据包。如果 ping 无效，则首先转储统计数据，确定是接收还是传输有问题，亦或二者都有问题。

3.3.3 Pinmux 问题

Pinmux 问题的发生条件：

1. 引脚没有设置正确，例如为 PCIe 等配置 RGMII Tx 引脚
2. Pinmux 文件没有正确导入到 bootloader。
3. 另一个内核或软件覆盖引脚多路复用焊盘。

Pinmux 问题并非 MAC2MAC 特有的问题，而是可产生更难捕捉的一般问题，表现为时钟禁用或某些数据线路不切换。排除软件问题后，最好将示波器连接到数据和时钟线路，确保它们按照标准切换。

Pinmux 问题可导致错误的统计数据，例如短帧、SFD 错误等。软件问题不会发生这种错误。

避免此类问题的方法：

1. 使用 pinmux 工具并避免手动编辑
2. 在完全启动后转储焊盘寄存器数据，并确保值在预期的线路上
3. 使用尽可能类似 EVM 的 Pinmux，这有助于避免其他驱动程序覆盖焊盘的错误。

3.3.4 硬件问题

硬件问题是指电路板上的路由错误、缓冲器问题、晶体问题或电容问题。为避免这些问题，请将原理图交给 TI 审阅。您的 TI 代表将帮助您转交。

这些问题通常表现为时钟偏移、信号不良、统计错误。如果运行 `pdk/packages/ti/drv/enet/tools/debug_gels/cpsw_stats.gel` 时显示错误帧，则很可能是硬件或 Pinmux 问题。为调试这些问题，请连接示波器并检查信号。

下面介绍了 RGMII 和 SGMII 的一些寄存器和相应设置，可用于检查配置是否正确完成。这些设置也可使用前面提到的调试 gel 文件转储。

3.3.4.1 RGMII 调试

1. 如图 3-1 所示，确认对应的端口模式设置：RGMII

Table 5-915. CTRLMMR_ENET1_CTRL Instances

Instance	Proxy0 Physical Address	Proxy1 Physical Address
CTRL_MMRO	0010 4044h	0010 6044h

Figure 5-435. CTRLMMR_ENET1_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RGMII_ID_MODE	RESERVED	PORT_MODE_SEL		
R-0h			R/W-0h	R-0h	R/W-2h		

LEGEND: R = Read Only; R/W = Read/Write; -n = value after reset

Table 5-916. CTRLMMR_ENET1_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	RGMII_ID_MODE	R/W	0h	Port1 RGMII internal transmit delay selection 0h - Internal transmit delay 1h - Reserved
3	RESERVED	R	0h	Reserved
2-0	PORT_MODE_SEL	R/W	2h	Selects Ethernet switch Port1 interface 0h - GMII/MII (not supported) 1h - RMII 2h - RGMII 3h - SGMII 4h - QSGMII 5h - XFI (not supported) 6h - QSGMII_SUB 7h - Reserved

图 3-1. CTRLMMR_ENET1_CTRL 寄存器

2. 确认对应的 RGMII 全双工/速度/链路设置如图 3-2 所示，并确保链路接通。

Table 12-2139. CPSW_SS_RGMII1_STATUS_REG Instances

Instance	Physical Address
CPSW0_NUSS_SS	0C00 0030h

Figure 12-921. CPSW_SS_RGMII1_STATUS_REG Register

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED							
R-X							
7	6	5	4	3	2	1	0
RESERVED				FULLDUPLEX	SPEED		LINK
R-X				R-0h	R-0h		R-0h

LEGEND: R = Read Only; -n = value after reset

Table 12-2140. CPSW_SS_RGMII1_STATUS_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3	FULLDUPLEX	R	0h	RGMII Port 1 full duplex: 0h = Half-duplex 1h = Full-duplex
2-1	SPEED	R	0h	RGMII Port 1 speed: 0h = 10Mbps 1h = 100Mbps 2h = 1000Mbps 3h = Reserved
0	LINK	R	0h	RGMII Port 1 link indicator: 0h = Link is down, 1h = Link is up

图 3-2. CPSW_SS_RGMII1_STATUS_REG 寄存器

3.3.4.2 SGMII 调试

1. 如图 3-3 所示，确认对应的端口模式设置：RGMII

Table 5-915. CTRLMMR_ENET1_CTRL Instances

Instance	Proxy0 Physical Address	Proxy1 Physical Address
CTRL_MMRO	0010 4044h	0010 6044h

Figure 5-435. CTRLMMR_ENET1_CTRL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RGMII_ID_MODE	RESERVED	PORT_MODE_SEL		
R-0h			R/W-0h	R-0h	R/W-2h		

LEGEND: R = Read Only; R/W = Read/Write; -n = value after reset

Table 5-916. CTRLMMR_ENET1_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4	RGMII_ID_MODE	R/W	0h	Port1 RGMII internal transmit delay selection 0h - Internal transmit delay 1h - Reserved
3	RESERVED	R	0h	Reserved
2-0	PORT_MODE_SEL	R/W	2h	Selects Ethernet switch Port1 interface 0h - GMII/MII (not supported) 1h - RMII 2h - RGMII 3h - SGMII 4h - QSGMII 5h - XFI (not supported) 6h - QSGMII_SUB 7h - Reserved

图 3-3. CTRLMMR_ENET1_CTRL 寄存器

- 确认将对应端口的 phyAddr 配置为 CPSW_PHY_INVALID_PHYADDR
- 如图 3-4 所示，确认对应端口的主模式。

Table 12-2175. CPSW_SGMII_CONTROL_REG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	X	
6	TEST_PATTERN_EN	R/W	0h	Test Pattern Enable. Force the output of K28.5 on TX_ENC for test purposes. 0h = Operation 1h = Forced K28.5 on transmit output
5	MASTER	R/W	0h	Master Mode. 0h = Slave Mode 1h = Master mode – Set to one for one side of a direct connection. When this bit is set, the control logic uses the CPSW_SGMII_MR_ADV_ABILITY_REG register to determine speed and duplexity instead of the CPSW_SGMII_MR_LP_ADV_ABILITY_REG register. Master mode allows a CPSGMII direct connection with auto-negotiation or with a forced link.

图 3-4. CPSW_SGMII_CONTROL_REG_j 寄存器

- 如图 3-5 所示，确认对应的 SGMII 状态：SERDES PLL 和链路状态。

Table 12-2178. CPSW_SGMII_STATUS_REG_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	X	
5	FIB_SIG_DETECT	R	0h	Fiber Signal Detect. This is the FIB_SIG_DETECT input pin.
4	LOCK	R	0h	Lock. This is the LOCK input pin. Indicates that the SERDES PLL is locked.
3	MR_PAGE_RX	R	0h	Next Page Received. This bit is set to one by the auto-negotiation state machine when the next page has been received. This bit is cleared to 0h by a host write of 1h to the [3]MR_NP_LOADED bit in the CPSW_SGMII_CONTROL_REG register. This value is not valid until the lock status bit is ([4] LOCK) asserted.
2	MR_AN_COMPLETE	R	0h	Auto negotiation complete. This value is not valid until the lock status bit is asserted. 0h = Auto-negotiation is not complete 1h = Auto-negotiation is completed.
1	AN_ERROR	R	0h	Auto negotiation error. For SGMII mode, an auto-negotiation error occurs when halfduplex Gigabit is commanded. For FIBER mode, an auto-negotiation error occurs if both sides cannot be full duplex. This value is not valid until the lock status bit is asserted. 0h = No auto-negotiation error 1h = Auto-negotiation error
0	LINK	R	0h	Link indicator. This value is not valid until the lock status bit is asserted. 0h = Link is not up. 1h = Link is up.

图 3-5. CPSW_SGMII_STATUS_REG_j 寄存器

4 总结

首先，本文介绍了 MAC2MAC 的背景。其次，它描述了各种应用场景来辅助客户设计产品。然后介绍了两种 MAC2MAC 解决方案：基于 RGMII 的解决方案和基于 SGMII 的解决方案。最后，我们展示了实现这两种解决方案的步骤并提供了一些调试建议。

在实际应用中，我们建议在 IO 端口资源充裕时先使用基于 RGMII 的 MAC2MAC 解决方案。此解决方案没有 SERDES，因此配置简单而且容易使用。

5 参考文献

1. [Processor SDK Linus for AM335x - CPSW 内核驱动程序](#)
2. [Wikipedia - OSI 模型](#)
3. [DRA829/TDA4VM/AM752x 技术参考手册](#)
4. [适用于 ADAS 和自动驾驶汽车的 TDA4VM Jacinto™ 处理器数据表](#)

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 © 2022，德州仪器 (TI) 公司