

## 具有千兆位 GMII/RGMII 和 MII/RMII 接口的 集成 5 端口 10/100 管理型以太网开关

### 目标应用

- 采用符合 IEEE 802.3 标准的媒体访问控制器 (Media Access Controller, MAC) 的工业以太网应用 (以太网/IP、Profinet 和 MODBUS TCP 等)
- VoIP 电话
- 机顶盒/游戏盒
- 汽车
- 工业控制
- IPTV POF
- 带四个全线速 LAN 端口的 SOHO 家庭网关
- 宽带网关/防火墙/VPN
- 集成 DSL/电缆调制解调器
- 无线 LAN 接入点 + 网关
- 独立 10/100 开关
- 联网测量和控制系统

### 特性

- 管理功能
  - KSZ8795CLX 包含 10/100BASE-T/TX 开关系统的所有功能, 该系统整合了开关引擎、帧缓冲区管理、地址查找表、队列管理、MIB 计数器、介质访问控制器 (MAC) 和 PHY 收发器
  - 非阻塞存储与转发开关结构可确保通过利用 1024 条目转发表快速传送数据包
  - 端口镜像/监视/嗅探: 与任意端口进行传入和/或传出通信
  - 完全按照标准来收集统计信息的 MIB 计数器 (每个端口 36 个计数器)
  - 支持用于 MIB 计数器中基于端口的刷新和冻结命令的硬件
  - 为诊断提供远程、PHY 和 MAC 多环回模式支持
  - 为拓扑管理和环形/线性恢复提供快速生成树支持 (RSTP)
- 可靠的 PHY 端口
  - 四个符合 IEEE 802.3/802.3u 标准的集成以太网收发器, 支持 10BASE-T 和 100BASE-TX

- 支持 802.1az EEE
- 为差分对应用片内终结电阻和内部偏置以降低功耗
- HP 自动 MDI/MDI-X 交叉支持, 应用中无需区分直通电缆与交叉电缆
- MAC 和 GMAC 端口
  - 四个内部介质访问控制 (MAC1 至 MAC4) 单元和一个内部千兆位介质访问控制 (GMAC5) 单元
  - 为端口 5 GMAC5 及上行链路提供 GMII、RGMII、MII 或 RMII 接口支持
  - 2 KB 超大数据包支持
  - 端口 5 上支持尾部标记模式 (在 FCS 前添加一个字节), 用于通知处理器哪个传入端口接收数据包及其优先级
  - 支持精简介质无关接口 (Reduced Media Independent Interface, RMII), 采用 50 MHz 参考时钟输出
  - 端口 5 上支持 PHY 模式或 MAC 模式下的介质无关接口 (Media Independent Interface, MII)
  - LinkMD<sup>®</sup> 电缆诊断功能, 用于确定电缆的开路、短路和长度
- 高级开关功能
  - 非阻塞存储与转发开关结构可确保通过利用 1024 条目转发表快速传送数据包
  - 64 KB 帧缓冲区 RAM
  - 为最多 128 个有效 VLAN 组提供 IEEE 802.1q VLAN 支持 (VLAN ID 的整个范围为 4096)
  - 基于各个端口插入或移除 IEEE 802.1p/Q 标记 (传出)
  - 基于各个端口的 VLAN ID 标记/无标记选项
  - 完全符合 IEEE 802.3/802.3u 标准
  - 带强制模式选项的 IEEE 802.3x 全双工以及半双工背压冲突流控制
  - IEEE 802.1w 快速生成树协议支持

# KSZ8795CLX

---

- 用于多播数据包过滤的IGMP v1/v2/v3 侦听
- QoS/CoS数据包优先级支持：基于802.1p和DiffServ，根据四个优先级重映射每个端口的802.1p优先级字段
- IPv4/IPv6 QoS支持
- IPV6多播监听发现（Multicast Listener Discovery, MLD）侦听
- 基于各个端口的传入端口和传出端口可编程速率限制
- 基于各数据包无抖动的速率限制支持
- 端口5上支持尾部标记模式（在FCS前添加一个字节），用于通知处理器哪个传入端口接收数据包
- 采用百分比控制的广播风暴保护（基于全局和各个端口）
- 带64 KB帧缓冲区的1K条目转发表
- 支持动态数据包映射的4个优先级队列，适用于IEEE 802.1P、IPV4 TOS（DIFFSERV）和IPV6通信类等
- 通过AMD的魔术包支持WoL
- VLAN和地址过滤
- 通过访问控制列表（Access Control List, ACL）支持基于802.1x端口的安全、认证以及基于MAC的认证
- 提供基于端口和基于规则的ACL以支持第2层MAC SA/DA地址、第3层IP地址和IP掩码、第4层TCP/UDP端口编号、IP协议、TCP标记和针对端口安全过滤的补偿
- 基于每秒位数（Bit per Second, bps）的传入和传出速率限制和基于数据包的速率限制（pps）
- 配置寄存器访问
  - 高速SPI（4线，最高50 MHz）接口，用于访问所有内部寄存器
  - MII管理（MIIM和MDC/MDIO 2线）接口，用于访问符合IEEE 802.3规范条款22.2.4.5的所有PHY寄存器
  - I/O引脚配置脚功能，用于在复位期间设置I/O引脚的某些寄存器位
  - 可实时配置控制寄存器
- 电源及电源管理
  - 全芯片软件掉电（所有寄存器值均不保存，配置脚输入值将在从掉电状态释放后重新配置）
  - 按端口软件掉电
  - 电能检测掉电（Energy Detect Power-Down, EDPD），用于在电缆移除后禁止PHY收发器
- 即使电缆未移除，也支持IEEE P802.3az高效节能以太网（Energy Efficient Ethernet, EEE），可降低收发器在LPI状态下的功耗
- 动态时钟树控制，用于降低未使用区域中的时钟速度
- 低功耗，变压器上没有多余功耗
- 电压：使用外部LDO电源
- 3.3V或2.5V模拟V<sub>DDAT</sub>
- V<sub>DDIO</sub>支持3.3V、2.5V和1.8V
- 模拟和数字内核电源采用1.2V低电压
- 带可配置数据包控制的WoL支持
- 其他特性
  - 单一25 MHz +50 ppm参考时钟要求
  - 功能全面的可编程双LED指示灯，支持链路、活动、全/半双工和10/100速度指示
- 封装和环境
  - 商业级温度范围：0°C至+70°C
  - 工业级温度范围：-40°C至+85°C
  - 提供80引脚LQFP无铅（符合RoHS标准）封装
  - 支持5 kV的人体模型（Human Body Model, HBM）ESD额定值
  - 0.065 μm CMOS技术，可降低功耗

## 致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 [CTRC@microchip.com](mailto:CTRC@microchip.com)。我们期待您的反馈。

### 最新数据手册

欲获得本数据手册的最新版本，请访问我公司网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A\_CN 是文档的 A 版本。

### 勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站 <http://www.microchip.com>
- 当地 Microchip 销售办事处（见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本（包括文献编号）。

### 客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 [www.microchip.com](http://www.microchip.com) 上注册。

# KSZ8795CLX

---

## 目录

1.0 简介 .....	5
2.0 引脚说明和配置 .....	6
3.0 功能说明 .....	13
4.0 器件寄存器 .....	46
5.0 工作特性 .....	112
6.0 电气特性 .....	113
7.0 时序图 .....	115
8.0 复位电路 .....	125
9.0 隔离变压器的选择 .....	126
10.0 参考晶振的选择 .....	126
11.0 封装外形 .....	127
附录A: 数据手册版本历史 .....	128
Microchip网站 .....	129
变更通知客户服务 .....	129
客户支持 .....	129
产品标识体系 .....	130

## 1.0 简介

### 1.1 一般说明

KSZ8795CLX 是一款高度集成的第2层5端口管理型开关，具有多个旨在降低系统成本的功能。它适用于需要四个10/100 Mbps铜端口和一个10/100/1000 Mbps千兆位上行链路端口的成本敏感型应用。KSZ8795CLX兼具小封装外形、低功耗和内部偏置以及片内终结电阻。其丰富的功能集包括增强型电源管理、可编程速率限制和优先比、基于标记和端口的VLAN、基于端口的安全功能和基于ACL规则的数据包过滤技术、带四个队列的服务质量（Quality-of-Service, QoS）优先级、管理接口、增强型MIB计数器、高性能存储器带宽和基于共用存储器且支持非阻塞的开关结构。KSZ8795CLX为多个CPU数据接口提供支持，可有效满足当前和新兴快速以太网及千兆位以太网应用的需求，其中端口5 GMAC可配置为GMII、RGMII、MII和RMII模式中的任意一种。

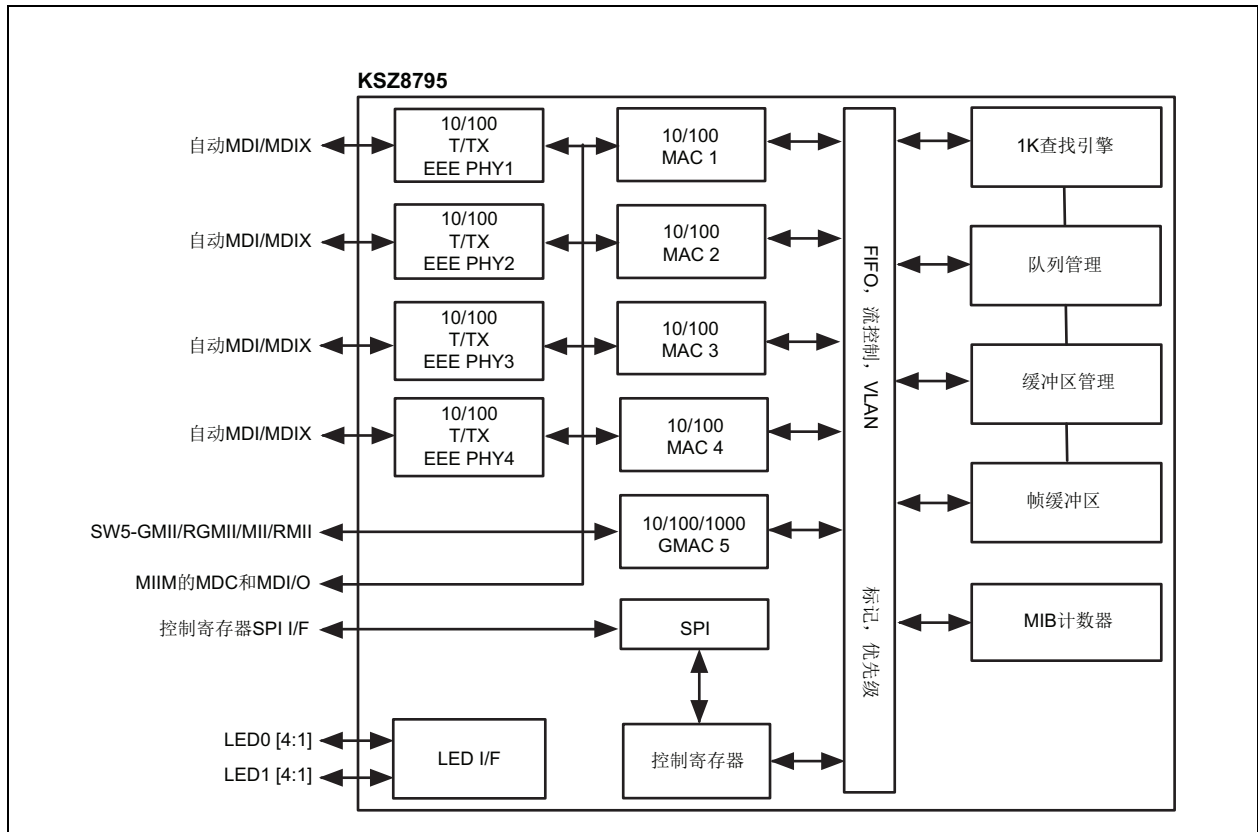
KSZ8795CLX采用行业领先的以太网模拟和数字技术制造，提供旨在减轻主机处理负载和简化整个设计的各种功能。

- 四个集成10/100BASE-T/TX MAC/PHY
- 一个集成10/100/1000BASE-T/TX GMAC，带可选的GMII、RGMII、MII和RMII接口
- 80引脚LQFP小型封装

为满足高效节能的环境要求，设计中加入了各种可靠的电源管理功能，包括节能以太网（EEE）、PME和LAN唤醒（Wake-on-LAN, WoL）。

KSZ8795CLX仅支持SPI和MIIM两种管理接口模式，SPI用于访问所有寄存器，而MIIM模式通过MDC/MDIO接口访问所有PHY寄存器。

图1-1: 功能框图



# KSZ8795CLX

## 2.0 引脚说明和配置

图2-1: 80-LQFP引脚分配 (俯视图)

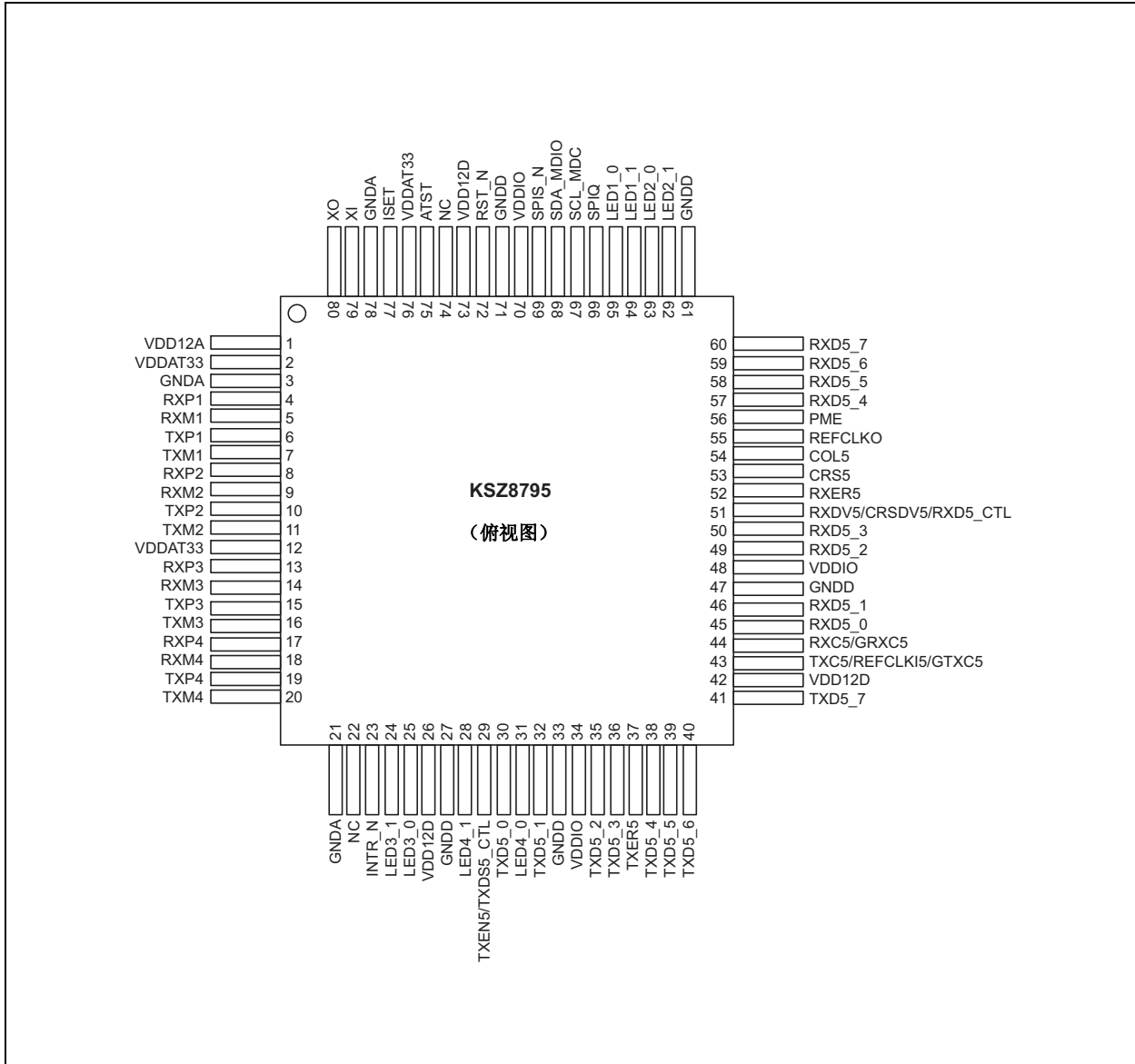


表2-1: 信号——KSZ8795CLX

引脚编号	引脚名称	类型 注2-1	端口	说明
1	VDD12A	P	—	1.2V内核电源。
2	VDDAT	P	—	3.3V或2.5V模拟电源。
3	GND A	GND	—	模拟地。
4	RXP1	I	1	端口1物理接收信号+（差分）。
5	RXM1	I	1	端口1物理接收信号-（差分）。
6	TXP1	O	1	端口1物理发送信号+（差分）。
7	TXM1	O	1	端口1物理发送信号-（差分）。
8	RXP2	I	2	端口2物理接收信号+（差分）。
9	RXM2	I	2	端口2物理接收信号-（差分）。
10	TXP2	O	2	端口2物理发送信号+（差分）。
11	TXM2	O	2	端口2物理发送信号-（差分）。
12	VDDAT	P		3.3V或2.5V模拟电源。
13	RXP3	I	3	端口3物理接收信号+（差分）。
14	RXM3	I	3	端口3物理接收信号-（差分）。
15	TXP3	O	3	端口3物理发送信号+（差分）。
16	TXM3	O	3	端口3物理发送信号-（差分）。
17	RXP4	I	4	端口4物理接收信号+（差分）。
18	RXM4	I	4	端口4物理接收信号-（差分）。
19	TXP4	O	4	端口4物理发送信号+（差分）。
20	TXM4	O	4	端口4物理发送信号-（差分）。
21	GND A	GND	—	模拟地。
22	NC	NC	—	无连接。
23	INTR_N	Opu	—	中断：低电平有效。该引脚是漏极开路输出引脚。 注：使用该引脚时，需连接外部上拉电阻。
24	LED3_1	lpu/O	3	端口3 LED 指示灯1： 有关详细信息，请参见全局寄存器11的bit[5:4]。 配置脚选项：开关端口5 GMAC5接口模式通过LED3[1:0]选择 00 = SW5-MII为MII 01 = SW5-RMII为RMII 10 = SW5-GMII为GMII 11 = SW5-RGMII为RGMII（默认值）
25	LED3_0	lpu/O	3	端口3 LED 指示灯0： 有关详细信息，请参见全局寄存器11的bit[5:4]。 配置脚选项：请参见LED3_1。

# KSZ8795CLX

表 2-1: 信号——KSZ8795CLX (续)

引脚编号	引脚名称	类型 注 2-1	端口	说明
26	VDD12D	P	—	1.2V 内核电源。
27	GNDD	GND	—	数字地。
28	LED4_1	lpu/O	4	端口 4 LED 指示灯 1: 有关详细信息, 请参见全局寄存器 11 的 bit[5:4]。
29	TXEN5/ TXD5_CTL	lpd	5	GMII/MII/RMII: 端口 5 开关发送使能。 RGMII: 发送数据控制。
30	TXD5_0	lpd	5	GMII/RGMII/MII/RMII: 端口 5 开关发送 bit[0]。
31	LED4_0	lpu/O	4	端口 4 LED 指示灯 0: 有关详细信息, 请参见全局寄存器 11 的 bit[5:4]。
32	TXD5_1	lpd	5	GMII/RGMII/MII/RMII: 端口 5 开关发送 bit[1]。
33	GNDD	GND	—	数字地。
34	VDDIO	P	—	数字 I/O 电路的 3.3V、2.5V 或 1.8V 数字 VDD。
35	TXD5_2	lpd	5	GMII/RGMII/MII: 端口 5 开关发送 bit[2]。 RMII: 无连接。
36	TXD5_3	lpd	5	GMII/RGMII/MII: 端口 5 开关发送 bit[3]。 RMII: 无连接。
37	TXER5	lpd	5	GMII/MII: 端口 5 开关发送错误。 RGMII/RMII: 无连接。
38	TXD5_4	lpd	5	GMII: 端口 5 开关发送 bit[4]。 RGMII/MII/RMII: 无连接。
39	TXD5_5	lpd	5	GMII: 端口 5 开关发送 bit[5]。 RGMII/MII/RMII: 无连接。
40	TXD5_6	lpd	5	GMII: 端口 5 开关发送 bit[6]。 RGMII/MII/RMII: 无连接。
41	TXD5_7	lpd	5	GMII: 端口 5 开关发送 bit[7]。 RGMII/MII/RMII: 无连接。
42	VDD12D	P	—	1.2V 内核电源。
43	TXC5/ REFCLKI/ GTXC5	I/O	5	端口 5 开关 GMAC5 时钟引脚: MII: 2.5/25 MHz 时钟, PHY 模式为输出, MAC 模式为输入。 RMII: 用于在正常模式下接收 50 MHz 时钟的输入。 GMII: 输入 125 MHz 时钟, 用于发送操作。 RGMII: 输入 125 MHz 时钟, 在下降沿和上升沿锁存要发送的数据。
44	RXC5/ GRXC5	I/O	5	端口 5 开关 GMAC5 时钟引脚: MII: 2.5/25 MHz 时钟, PHY 模式为输出, MAC 模式为输入。 RMII: 输出 50 MHz 参考时钟, 用于在时钟模式下接收/发送。 GMII: 输出 125 MHz 时钟, 用于接收操作。 RGMII: 输出 125 MHz 时钟, 在下降沿和上升沿锁存要接收的数据。



**表 2-1: 信号——KSZ8795CLX (续)**

引脚编号	引脚名称	类型 注 2-1	端口	说明
45	RXD5_0	lpd/O	5	GMII/RGMII/MII/RMII: 端口 5 开关接收 bit[0]。
46	RXD5_1	lpd/O	5	GMII/RGMII/MII/RMII: 端口 5 开关接收 bit[1]。
47	GNDD	GND	—	数字地。
48	VDDIO	P	—	数字 I/O 电路的 3.3V、2.5V 或 1.8V 数字 VDD。
49	RXD5_2	lpd/O	5	GMII/RGMII/MII: 端口 5 开关接收 bit[2]。 RMII: 无连接
50	RXD5_3	lpd/O	5	GMII/RGMII/MII: 端口 5 开关接收 bit[3]。 RMII: 无连接
51	RXDV5/ CRSDV5/ RXD5_CTL	lpd/O	5	GMII/MII: RXDV5 用于端口 5 开关 GMII/MII 接收数据有效信号。 RMII: CRSDV5 用于端口 5 RMII 载波侦听/接收数据有效输出。 RGMII: RXD5_CTL 用于端口 5 RGMII 接收数据控制信号。
52	RXER5	lpd/O	5	GMII/MII: 端口 5 开关接收错误。 RGMII/RMII: 无连接。
53	CRS5	lpd/O	5	GMII/MII: 端口 5 开关 MII 模式载波侦听。 RGMII/RMII: 无连接。
54	COL5	lpd/O	5	GMII/MII: 端口 5 开关 MII 冲突检测。 RGMII/RMII: 无连接。
55	REFCLKO	lpu/O	—	25 MHz 时钟输出 (选项) 由配置脚引脚 LED2_0 和全局寄存器 11 bit[1] 控制。默认情况下为使能状态, 不使用时最好将其禁止。
56	PME_N	I/O	—	电源管理事件 该输出信号用于指示是否在检测到唤醒帧后检测到 WoL 事件。 KSZ8795CLX 正请求系统从低功耗模式唤醒。其有效极性可编程, 默认极性为低电平有效。
57	RXD5_4	lpd/O	5	GMII: 端口 5 开关接收 bit[4]。 RGMII/MII/RMII: 无连接。
58	RXD5_5	lpd/O	5	GMII: 端口 5 开关接收 bit[5]。 RGMII/MII/RMII: 无连接。
59	RXD5_6	lpd/O	5	GMII: 端口 5 开关接收 bit[6]。 RGMII/MII/RMII: 无连接。
60	RXD5_7	lpd/O	5	GMII: 端口 5 开关接收 bit[7]。 RGMII/MII/RMII: 无连接。
61	GNDD	GND	—	数字地。

# KSZ8795CLX

表 2-1: 信号——KSZ8795CLX (续)

引脚编号	引脚名称	类型 注 2-1	端口	说明
62	LED2_1	lpu/O	2	<p>端口 2 LED 指示灯 1: 有关详细信息, 请参见全局寄存器 11 的 bit[5:4]。 配置脚选项: 端口 5 GMII/MII 和 RMII 模式选择 当端口 5 处于 GMII/MII 模式时: PU = GMII/MII 处于 GMAC/MAC 模式。(默认) PD = GMII/MII 处于 GPHY/PHY 模式。 <b>注:</b> 将 GMAC5 GMII 设为 GPHY 模式时, CRS 和 COL 引脚会从输入更改为输出。将 MII 设为 PHY 模式时, CRS、COL、RXC 和 TXC 引脚将从输入更改为输出。</p> <p>当端口 5 处于 RMII 模式时: PU = RMII 时钟模式, 使用 25 MHz OSC 时钟并通过引脚 RXC5 提供 50 MHz RMII 时钟。 PD = RMII 正常模式, 端口 5 RMII 上的 TXC5/REFCLKI5 引脚将接收 50 MHz 外部时钟。 <b>注:</b> 端口 5 还可在 RMII 模式下使用内部或外部时钟, 具体取决于此配置脚引脚或寄存器 86 (0x56) bit[7] 的设置。</p>
63	LED2_0	lpu/O	2	<p>端口 2 LED 指示灯 0: 有关详细信息, 请参见全局寄存器 11 的 bit[5:4]。 配置脚选项: REFCLKO 使能 PU = 使能 REFCLK_O (25 MHz) (默认)。 PD = 禁止 REFCLK_O。 <b>注:</b> 如果没有为系统提供额外的 25 MHz 时钟, 最好禁止该 25 MHz 时钟。</p>
64	LED1_1	lpu/O	1	<p>端口 1 LED 指示灯 1: 有关详细信息, 请参见全局寄存器 11 的 bit[5:4]。 配置脚选项: PLL 时钟选择 PU = 尽管处于端口 5 RMII 正常模式, 但仍使用 XI/XO 引脚提供的 25 MHz 时钟。 PD = 在端口 5 RMII 正常模式下, 使用引脚 TXC5 提供的外部时钟。 <b>注:</b> 如果在端口 5 RMII 正常模式下接收的时钟有很大的时钟抖动, 则可选择将 25 MHz 晶振/振荡器作为开关的时钟源。</p>
65	LED1_0	lpu/O	1	<p>端口 1 LED 指示灯 0: 有关详细信息, 请参见全局寄存器 11 的 bit[5:4]。 配置脚选项: GMII/RGMII 下的速度选择 PU = GMII/RGMII 下为 1 Gbps (默认)。 PD = GMII/RGMII 下为 10/100 Mbps。 <b>注:</b> 也可通过内部寄存器编程。</p>
66	SPIQ	lpd/O	所有	<p>SPI 从模式下的 SPI 串行数据输出: 配置脚选项: 串行总线配置。 PD = SPI 从模式。 PU = MDC/MDIO 模式。 <b>注:</b> 需要外部上拉电阻或下拉电阻。</p>
67	SCL_MDC	lpu	所有	<p>SPI 或 MDC/MDIO 接口的时钟输入: SPI 从模式下最高 50 MHz 输入时钟。 MDC/MDIO 下最高 25 MHz 输入时钟 (用于 MIIM 访问)。</p>

表 2-1: 信号——KSZ8795CLX (续)

引脚编号	引脚名称	类型 注 2-1	端口	说明
68	SDA_MDIO	lpu/O	所有	SPI 或 MDC/MDIO 接口的数据： SPI 从模式下的串行数据输入。 MDC/MDIO 接口数据输入/输出。
69	SPIS_N	lpu	所有	SPI 从模式芯片选择（低电平有效）： 在 SPI 从模式下开始 SPI 数据传输。当 SPIS_N 为高电平时，KSZ8795CLX 会被取消选择，SPIQ 保持高阻抗状态。从高电平跳变到低电平将发起 SPI 数据传输。该引脚为低电平有效。
70	VDDIO	P	—	数字 I/O 电路的 3.3V、2.5V 或 1.8V 数字 VDD。
71	GNDD	GND	—	数字地。
72	RST_N	lpu	—	复位：该低电平有效信号将复位器件中的硬件。请参见本部分中的时序要求。
73	VDD12D	P	—	1.2V 内核电源。
74	NC	NC	—	无连接。
75	ATST	NC	—	无连接。出厂测试引脚。
76	VDDAT	P	—	3.3V 或 2.5V 模拟电源。
77	ISET		—	发送输出电流设置： 该引脚用于配置物理发送输出电流。它应通过 12.4 kΩ 1% 电阻连接到 GND。
78	GNDA	GND	—	模拟地。
79	XI	I	—	晶振时钟输入/振荡器输入： 使用 25 MHz 晶振时，该输入连接到晶振电路的一端。使用 3.3V 振荡器时，该输入为振荡器的输入。 晶振或振荡器的容差应为 ±50 ppm。
80	XO	O	—	晶振时钟输出： 使用 25 MHz 晶振时，该输出连接到晶振电路的一端。

注 2-1 P = 电源；GND = 地；I = 输入；O = 输出

I/O = 双向

lpu = 带内部上拉的输入。

lpd = 带内部下拉的输入。

lpd/O = 复位期间为带内部下拉的输入，否则为输出引脚。

lpu/O = 复位期间为带内部上拉的输入，否则为输出引脚。

OTRI = 输出为三态。

PU = 配置脚引脚上拉。

PD = 配置脚引脚下拉。

NC = 对于该产品为无连接或接地。

KSZ8795CLX 可用作管理型开关并利用配置脚输入引脚将器件配置为不同模式。配置脚输入引脚通过使用外部上拉/下拉电阻配置，以使掉电复位或热复位期间采样的引脚上产生高电平或低电平状态。下表详细介绍了相应功能。

# KSZ8795CLX

表2-2: 配置脚输入选项——KSZ8795CLX

引脚编号	引脚名称	类型 (注2-2)	说明
24和25	LED3[1,0]	lpu/O	<p>开关端口5 GMAC5接口模式选择: 配置脚选项: 00 = SW5-MII为MII 01 = SW5-RMII为RMII 10 = SW5-GMII为GMII 11 = SW5-RGMII为RGMII (默认值)</p>
62	LED2_1	lpu/O	<p>端口5 GMII/MII和RMII模式选择: 配置脚选项: 当端口5处于GMII/MII模式时: PU = GMII/MII处于GMAC/MAC模式 (默认)。 PD = GMII/MII处于GPHY/PHY模式。 <b>注:</b> 将GMAC5 GMII设为GPHY模式时, CRS和COL引脚会从输入更改为输出。将MII设为PHY模式时, CRS、COL、RXC和TXC引脚将从输入更改为输出。</p> <p>当端口5处于RMII模式时: PU = RMII时钟模式, 使用25 MHz OSC时钟并通过引脚RXC5提供50 MHz RMII时钟。 PD = RMII正常模式, 端口5 RMII上的TXC5/REFCLKI5引脚将接收50 MHz外部时钟。 <b>注:</b> 端口5还可在RMII模式下使用内部或外部时钟, 具体取决于此配置脚引脚或寄存器86 (0x56) bit[7]的设置。</p>
63	LED2_0	lpu/O	<p>REFCLKO使能: 配置脚选项: PU = 使能REFCLK_O (25 MHz) (默认)。 PD = 禁止REFCLK_O。</p>
64	LED1_1	lpu/O	<p>PLL时钟源选择: 配置脚选项: PU = 尽管处于端口5 RMII正常模式, 但仍使用XI/XO引脚提供的25 MHz时钟。 PD = 在端口5 RMII正常模式下, 使用引脚TXC5提供的外部时钟。 <b>注:</b> 如果在端口5 RMII正常模式下接收的时钟有很大的时钟抖动, 则仍可选择使用25 MHz晶振/振荡器作为开关的时钟源。</p>
65	LED1_0	lpu/O	<p>端口5千兆位选择: 配置脚选项: PU = GMII/RGMII模式下为1 Gbps (默认) PD = GMII/RGMII模式下为10/100 Mbps <b>注:</b> 也可通过内部寄存器编程</p>
66	SPIQ	lpd/O	<p>串行总线配置 配置脚选项: PD = SPI从模式 (默认)。 PU = MDC/MDIO模式。 <b>注:</b> 需要外部上拉电阻或下拉电阻。如果上行链路端口用于RGMII接口, 则建议使用SPI模式为RGMII v2.0设置寄存器86 (0x56) bit[4:3]; MDC/MDIO模式无法设置该功能。</p>

注2-2 lpd/O = 复位期间为带内部下拉的输入, 否则为输出引脚。

lpu/O = 复位期间为带内部上拉的输入, 否则为输出引脚。

## 3.0 功能说明

KSZ8795CLX包含4个10/100M物理层收发器、4个介质访问控制（MAC）单元以及1个集成第2层管理开关的千兆位介质访问控制（GMAC）单元。KSZ8795CLX有两种运行模式。第一种模式是用作4端口的独立开关，第二种模式是用作5端口的开关，其中第5个端口通过支持GMII、RGMII、MII和RMII的千兆位介质无关接口提供。该器件可用于实现集成宽带路由器。

KSZ8795CLX可灵活地驻留在管理模式下。在管理模式下，主机处理器可通过SPI总线或MDC/MDIO接口完全控制KSZ8795CLX。

在介质方面，KSZ8795CLX在支持自动MDI/MDI-X的所有铜端口上支持IEEE 802.3 10BASE-T和100BASE-TX。KSZ8795CLX可用作完全管理型的5端口开关，或者通过其SW-GMII/RGMII/MII/RMII接口连接到微处理器，以将器件集成到各种环境中。

该器件通过使用获得专利的模拟电路和DSP技术来增强发送和接收物理信号，从而使设计更高效并实现更低的功耗和更小的芯片尺寸。

从KSZ8995和KS8895到KSZ8795CLX的主要增强包括更多主机接口选项（例如GMII和RGMII接口）、节能特性（例如IEEE 802.1az节能以太网（EEE））、MLD侦听、LAN唤醒（Wake-on-LAN, WoL）、用于端口安全的基于端口的ACL过滤、增强型服务质量（QoS）优先级、快速生成树、IGMP侦听、端口镜像支持以及灵活的速率限制。

## 3.1 物理层（PHY）

### 3.1.1 100BASE-TX发送

100BASE-TX发送功能可执行并串转换、4B/5B编码、加扰、NRZ到NRZI转换以及MLT3编码和发送。电路首先执行并串转换，以将来自MAC的MII数据转换为125 MHz的串行比特流。该数据和控制比特流随后转换为4B/5B编码，紧接着进入加扰器。经过串行化的数据进一步从NRZ转换为NRZI格式，然后以MLT3电流输出的形式发送。输出电流通过一个精度为1%的12.4 kΩ外部电阻设置，以实现1:1的变压比。其上升/下降时间典型值为4 ns，符合ANSI TP-PMD标准中关于幅值平衡、过冲和时序抖动的规定。经过波形处理的10BASE-T输出也会整合到100BASE-TX发送器中。

### 3.1.2 100BASE-TX接收

100BASE-TX接收功能可执行自适应均衡、直流恢复、MLT3到NRZI转换、数据和时钟恢复、NRZI到NRZ转换、解扰、4B/5B解码以及串并转换。接收端首先通过均衡滤波器补偿双绞线上的码间干扰（Inter-Symbol Interference, ISI）。由于幅值损失和相位失真受电缆长度影响，均衡器必须调整其特性以优化性能。在本设计中，可变均衡器会将传入信号强度与某些已知电缆特性进行比较，并据此做出初始估计，然后调节自己的特性以实现优化。这是一个持续进行的过程，能够自动适应环境的变化（例如，温度变化）。

经过均衡的信号随后通过直流恢复和数据转换模块。直流恢复电路用于补偿基线漂移的影响并提高动态范围。差分数据转换电路会将MLT3格式转换回NRZI格式。限幅阈值也是自适应的。

时钟恢复电路从NRZI信号的边沿提取125 MHz时钟。随后，使用恢复后的时钟信号将NRZI信号转换为NRZ格式。接着，信号依次通过解扰器和4B/5B解码器。最后，NRZ串行数据转换为MII格式信号，并作为输入数据提供给MAC。

### 3.1.3 PLL时钟合成器

KSZ8795CLX为系统时序生成125 MHz、83 MHz、41 MHz、25 MHz和10 MHz的时钟信号。内部时钟由外部25 MHz晶振或振荡器生成。

### 3.1.4 加扰器/解扰器（仅限100BASE-TX）

加扰器的作用是扩展信号的功率频谱，以减小EMI和基线漂移。通过使用一个11位宽的线性反馈移位寄存器（Linear Feedback Shift Register, LFSR）对数据进行加扰。这可生成一个2047位的非重复序列。接收器随后将对序列与发送器中序列相同的传入数据流进行解扰。

# KSZ8795CLX

## 3.1.5 10BASE-T 发送

10BASE-T输出驱动器已整合到100BASE-T驱动器中以使用相同的磁件进行发送。它们在内部经过波形处理并被预加重至典型幅值为2.3V的输出中。由一个全1的曼彻斯特编码信号驱动时，谐波成分比基波信号至少低27 dB。

## 3.1.6 10BASE-T 接收

在接收端，采用输入缓冲区和电平检测静噪电路。差分输入接收器电路和PLL执行解码功能。曼彻斯特编码数据流分为时钟信号和NRZ数据。静噪电路抑制电平低于400 mV或脉冲宽度较窄的信号，以避免RXP或RXM输入端的噪声误触发解码器。当输入超过静噪限制时，PLL会锁定传入信号，KSZ8795CLX会解码数据帧。在相邻两次数据接收之间的空闲周期内，接收器时钟保持有效。

## 3.1.7 MDI/MDI-X 自动交叉

为了在两个相似器件之间省去交叉电缆，KSZ8795CLX支持HP自动MDI/MDI-X和IEEE 802.3u标准MDI/MDI-X自动交叉。请注意，默认为HP自动MDI/MDI-X。

自动检测功能检测远程发送和接收对，并为KSZ8795CLX器件正确分配发送和接收对。当最终用户不知道电缆型号时，自动检测功能极其有用，同时还可节省额外的上行链路配置连接。通过端口控制寄存器或MIIM PHY寄存器，可以禁止自动交叉功能。表3-1给出了IEEE 802.3u标准MDI和MDI-X的定义。

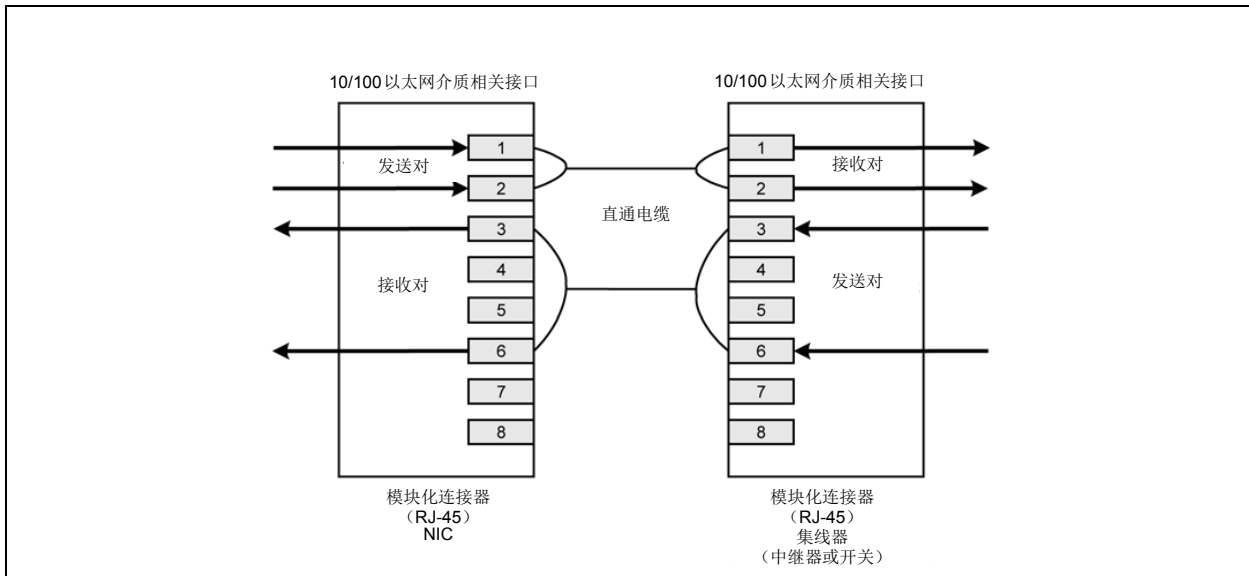
表3-1: MDI/MDI-X 引脚定义

MDI		MDI-X	
RJ-45 引脚	信号	RJ-45 引脚	信号
1	TD+	1	RD+
2	TD-	2	RD-
3	RD+	3	TD+
6	RD-	6	TD-

### 3.1.7.1 直通电缆

直通电缆将MDI器件连接到MDI-X器件或将MDI-X器件连接到MDI器件。图3-1给出了NIC卡（MDI）与开关或集线器（MDI-X）之间的典型直通电缆连接。

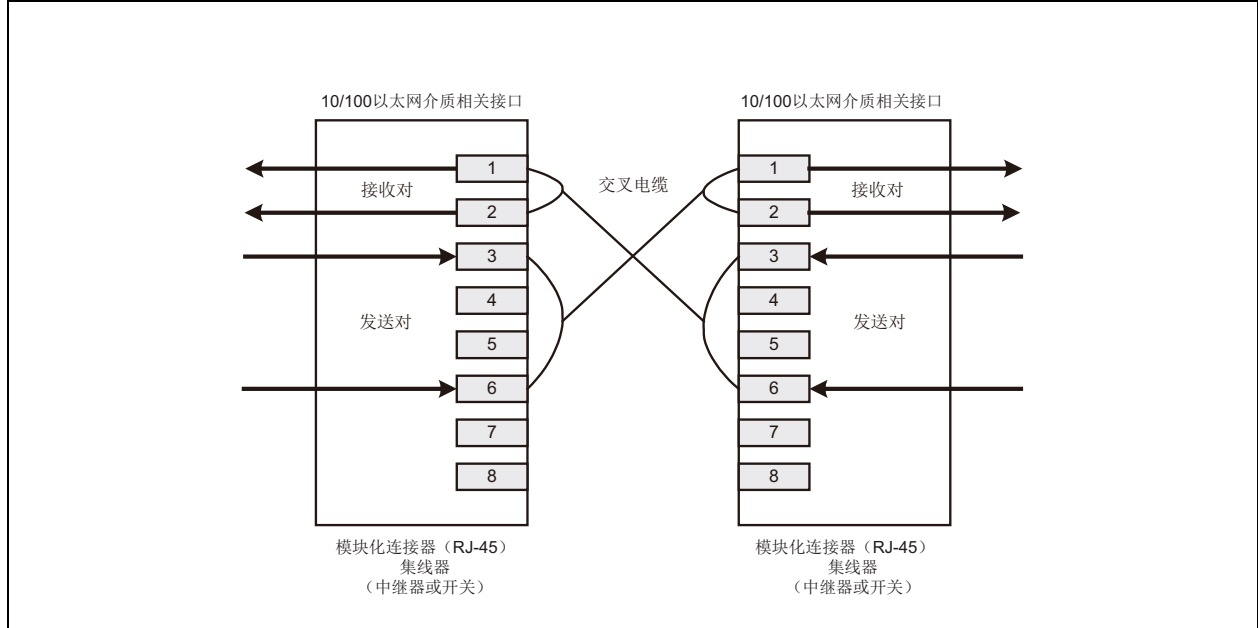
图3-1: 典型直通电缆连接



## 3.1.7.2 交叉电缆

交叉电缆将MDI器件连接到另一个MDI器件或将MDI-X器件连接到另一个MDI-X器件。下图给出了两个开关或集线器（两个MDI-X器件）之间的典型交叉电缆连接。

图3-2: 典型交叉电缆连接



## 3.1.8 自动协商

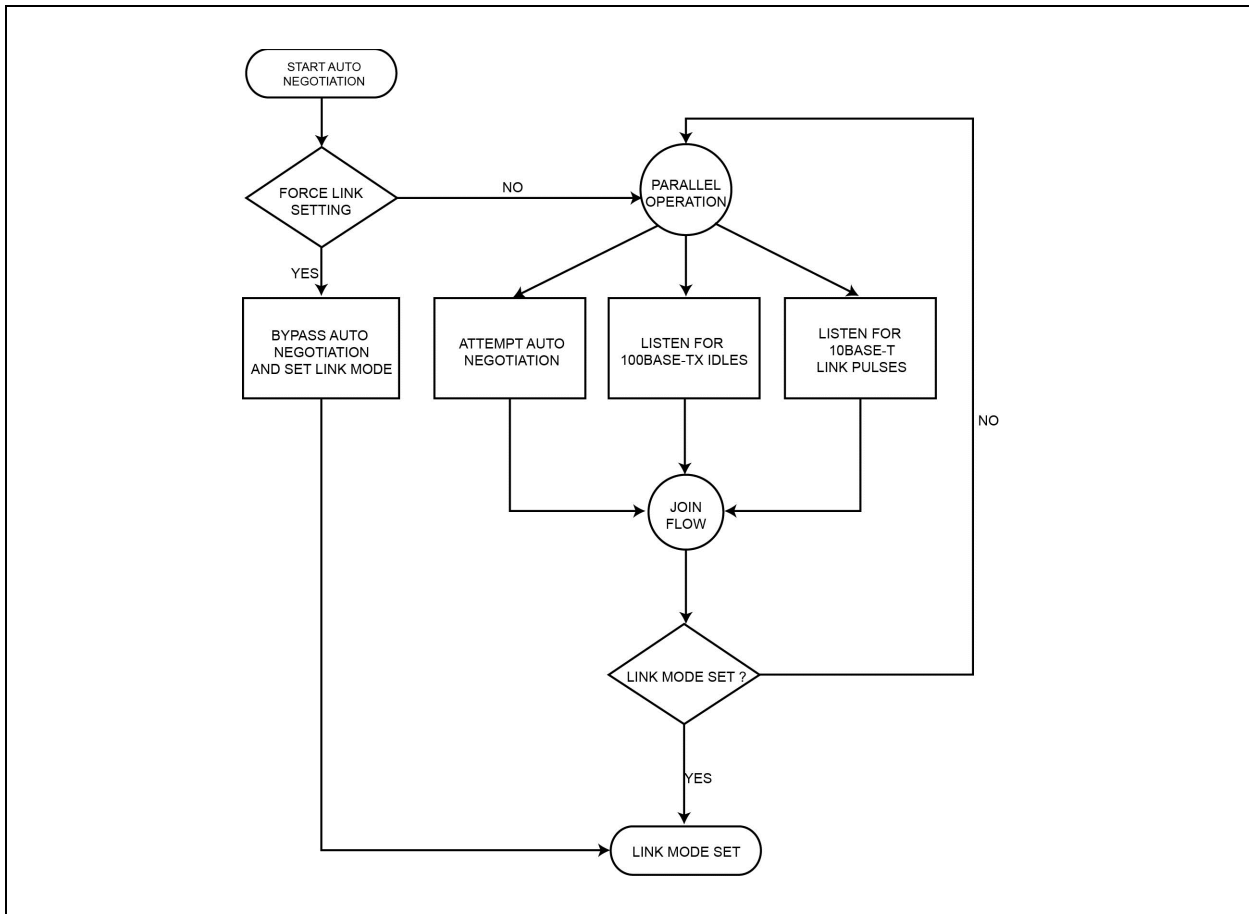
KSZ8795CLX遵循802.3委员会描述的自动协商协议。自动协商允许非屏蔽双绞线（Unshielded Twisted Pair, UTP）链路伙伴选择共用程度最高的工作模式。链路伙伴相互告知各自的能力，并将自己的能力与从链路伙伴接收的能力进行比较。将选择两个链路伙伴共用的最高速度和双工方式作为工作模式。仅铜端口支持自动协商。

下面列出了不同的速度和双工工作模式（最高到最低）：

- 100BASE-TX，全双工
- 100BASE-TX，半双工
- 10BASE-T，全双工
- 10BASE-T，半双工

如果不支持自动协商或KSZ8795CLX链路伙伴被强制为绕开自动协商，则KSZ8795CLX将通过观察其接收器的信号来设置其工作模式。这称为并行检测，允许KSZ8795CLX在不存在自动协商通告协议时通过侦听固定信号协议来建立链路。自动协商链接过程如图3-3所示。

图3-3: 自动协商与并行操作



### 3.1.9 LINKMD<sup>®</sup> 电缆诊断

LinkMD 功能利用时域反射测量法 (Time-Domain Reflectometry, TDR) 分析电缆设备的常见电缆问题, 例如开路、短路和阻抗不匹配。

LinkMD 通过在 MDI 和 MDI-X 对上发送一个已知幅值和持续时间的脉冲, 然后分析反射信号的形状。通过对脉冲持续时间进行计时可指出到电缆故障点的距离, 最大距离为 200m, 精度为  $\pm 2m$ 。内部电路以用户可读的数字格式显示 TDR 信息。

**注:** 电缆诊断仅对铜缆连接有效。

#### 3.1.9.1 访问

通过分别设置端口 1、2、3 和 4 的 PHY 特殊控制/状态寄存器 26、42、58 和 74 以及 LinkMD 结果寄存器 27、43、59 和 75, 可初始化 LinkMD; 通过分别设置端口 1、2、3 和 4 的端口控制 10 寄存器, 可禁止自动 MDI/MDI-X。

此外, MIIM PHY 寄存器 0 和 1d 也可以用于 LinkMD 访问。

#### 3.1.9.2 用途

下面是一个在端口 1 上通过寄存器 {26、27 和 29} 使用 LinkMD 的示例过程:

1. 向寄存器 29 bit[2] 写入 1 来禁止自动 MDI/MDI-X, 以便能对用于发送 LinkMD 脉冲的差分对的手动控制。
2. 向寄存器 26 bit[4] 写入 1 来启动电缆诊断测试。该使能位自清零。
3. 等待 (查询) 寄存器 26 bit[4] 返回 0, 这指示电缆诊断测试完成。
4. 读取寄存器 26 bit[6:5] 中的电缆诊断测试结果。结果如下:



- 00 = 正常状态（有效测试）
- 01 = 电缆中检测到开路状态（有效测试）
- 10 = 电缆中检测到短路状态（有效测试）
- 11 = 电缆诊断测试失败（无效测试）

当KSZ8795CLX无法关闭链路伙伴时，将发生情形‘11’，即无效测试。在这种情况下，由于KSZ8795CLX无法确定检测到信号是生成的反射信号还是来自另一个源的信号，因此测试不会运行。

5. 通过将寄存器26 bit[0]与寄存器27 bit[7:0]相连，并将结果乘以一个常数0.4，可得出到故障点的距离。到电缆故障点的距离可由以下公式确定：

$$D \text{ (到电缆故障点的距离, 用米表示)} = 0.4 \times (\text{寄存器26 bit[0], 寄存器27 bit[7:0]})$$

乘以0.4之前，应将寄存器26 bit[0]和寄存器27 bit[7:0]的串接值转换为十进制。

对于不同的电缆条件（包括传播速度与正常情况显著不同的电缆），可对常数（0.4）进行校准。

使用端口2、3和4以及MIIM PHY寄存器时，LinkMD的用法类似。

### 3.1.9.3 LinkMD 示例

下面是一个在强制MDI-X模式的端口1、2、3和4上使用LinkMD的示例过程：

```
//禁止MDI/MDI-X并强制为MDI-X模式
//w为写（WRITE）寄存器；r为读（READ）寄存器。
w 1d 04
w 2d 04
w 3d 04
w 4d 04

//通过间接寄存器将内部寄存器暂时置1，针对LinkMD进行调整
w 6e a0
w 6f 4d
w a0 80

//为端口1、2、3和4使能LinkMD故障电缆测试
w 1a 10
w 2a 10
w 3a 10
w 4a 10

//等待至端口控制寄存器8 bit[4]返回0（自清零）
//诊断结果
r 1a
r 1b
r 2a
r 2b
r 3a
r 3b
r 4a
r 4b

//例如，在端口1上，根据寄存器0x1a和寄存器0x1b的值进行结果分析
//寄存器0x1a bit[6:5]用于开路或短路检测
//寄存器0x1a bit[0] + 寄存器0x1b bit[7:0] = CDT_Fault_Count [8:0]
//到故障点的距离约为0.4 x (CDT_Fault_Count [8:0])
```

# KSZ8795CLX

---

## 3.1.10 片内终结和内部偏置

KSZ8795CLX对所有端口和RX/TX差分对采用片内终结电阻，而不使用外部终结电阻，因此可以降低电路板成本并精简电路板布局。对于多个开关而言，与使用外部偏置和终结电阻相比，由于变压器不再消耗功耗，因此片内终结和内部偏置的组合可节省更多PCB空间和功耗。变压器的中心抽头不需要连接到模拟电源。

## 3.2 介质访问控制器（MAC）操作

KSZ8795CLX严格遵守IEEE 802.3标准，以最大化兼容性。

### 3.2.1 包间隔（IPG）

成功发送一帧后，将测量两个连续的MTXEN之间的96位时间IPG。如果当前数据包正在经历冲突，将测量MCRS和下一MTXEN之间的96位时间IPG。

### 3.2.2 后退算法

KSZ8795CLX实现IEEE标准802.3二进制指数后退算法以及可选的“积极模式”后退算法。经历16次冲突后，可根据寄存器3中的芯片配置选择丢弃该数据包。

### 3.2.3 迟冲突

如果一个发送数据包在发送512位时间后经历冲突，则会丢弃该数据包。

### 3.2.4 非法帧

KSZ8795CLX会丢弃小于64字节的帧，并且可编程为在寄存器4中接收最大1536字节的帧。对于特殊应用，KSZ8795CLX也可编程为在寄存器3 bit[6]中接收最大2K字节的帧。由于KSZ8795CLX支持VLAN标记，因此当存在这些标记时，可以调整最大大小。

### 3.2.5 流控制

KSZ8795CLX在发送端和接收端都支持标准802.3x流控制帧。

在接收端，如果KSZ8795CLX接收到暂停控制帧，KSZ8795CLX将不再发送下一个正常帧，直到暂停控制帧中指定的定时器超时。如果在当前定时器超时前接收到另一个暂停帧，定时器将使用第二个暂停帧中的新值进行更新。在这段时间内（处于流控制下），将只发送KSZ8795CLX的流控制帧。

在发送端，KSZ8795CLX以智能且有效的方式来决定何时调用流控制。流控制基于系统资源的可用性，包括可用缓冲区、可用发送队列和可用接收队列。

如果目标端口资源忙，KSZ8795CLX会对刚刚接收到数据包的端口进行流控制。KSZ8795CLX发出一个流控制帧（XOFF），其中包含IEEE 802.3x标准中定义的最大暂停时间。一旦资源被释放，KSZ8795CLX便会发出暂停时间为零的其他流控制帧（XON），以关闭流控制（开启向端口的发送）。此外，还提供迟滞特性来避免过度激活和停用流控制机制。

如果接收队列变满，KSZ8795CLX会对所有端口进行流控制。

### 3.2.6 半双工背压

KSZ8795CLX还提供了半双工背压选项（请注意，此选项不在IEEE 802.3标准中）。此选项的激活和停用条件与针对全双工指定的条件相同。如果需要背压，KSZ8795CLX会发送前导，以延迟其他站点的发送（载波监听延迟）。为避免发生IEEE 802.3标准中定义的Jabber和过度延迟，经过特定的一段时间后，KSZ8795CLX会停止载波监听，但在丢弃数据包后又迅速恢复，以禁止其他发送。这段较短的安静时间（无载波监听）旨在避免其他站点发送数据包，并将其站点保持在载波监听延迟状态。在存在背压的情况下，如果端口有要发送的数据包，载波监听型背压会中断，转而发

送这些数据包。如果没有更多数据包需要发送，载波监听型背压会重新激活，直到开关资源空闲。如果发生冲突，则会跳过二进制指数后退算法并立即生成载波监听，以减少进一步冲突的机会，并保持载波监听以避免接收数据包。要确保10BASE-T或100BASE-TX半双工模式下不丢失数据包，用户必须使能以下选项：

- 积极后退（寄存器3 bit[0]）
- 无冲突过量丢弃（寄存器4 bit[3]）
- 背压（寄存器4 bit[5]）

由于这不属于IEEE标准，因此默认不设置这些位。

### 3.2.7 广播风暴保护

KSZ8795CLX有一个智能选项，可避免开关系统接收过多的广播数据包。广播数据包通常转发给源端口以外的所有端口，因此会使用过多开关资源（带宽和发送队列中的可用空间）。KSZ8795CLX可选择包含“多播数据包”来实现风暴控制。广播风暴速率参数是全局编程的，可在各端口上单独使能或禁止。对于100BT，速率基于50 ms（0.05s）间隔；对于10BT，速率基于500 ms（0.5s）间隔。在每个间隔开始，计数器会清零，然后速率限制机制开始对间隔内的字节数进行计数。寄存器6和寄存器7中说明了速率定义。寄存器6和寄存器7中的默认设置为0x4A（十进制下的74）。这等于1%的比率，计算公式如下：

$$148.80 \text{ 帧/s} \times 50 \text{ ms} (0.05\text{s}) / \text{间隔} \times 1\% = 74 \text{ 帧/间隔 (近似值)} = 0x4A$$

## 3.3 开关内核

### 3.3.1 地址查找

内部查找表存储MAC地址及其相关信息。它包含1K的单播地址表和开关信息。KSZ8795CLX可以学习1K地址，并将器件本身从基于哈希的查找表中区分出来；根据工作环境和概率，此查找表可能无法保证器件能够学习的绝对地址数。

### 3.3.2 学习

如果满足以下条件，则内部查找引擎将使用新条目更新查找表：

- 查找表中不存在接收数据包的源地址（Source Address, SA）。
- 接收数据包完好；数据包无接收错误且长度合法。

查找引擎将合格的SA连同端口号和时间戳一起插入到表中。如果表已满，则首先删除表中的最后一个条目，以为新条目释放空间。

### 3.3.3 迁移

内部查找引擎还会监视某个站点是否被移动。如果发生这种情况，则会相应更新查找表。当满足以下条件时，将发生迁移：

- 接收数据包的SA位于表中，但相关的源端口信息不同。
- 接收数据包完好；数据包无接收错误且长度合法。

查找引擎会将表中的现有记录更新为新的源端口信息。

### 3.3.4 老化

无论何时出现相应的SA，查找引擎都会更新记录的时间戳信息。老化过程中会使用时间戳。当某个记录在一段时间内未更新时，查找引擎会将其从表中删除。查找引擎持续执行老化过程，并不断删除老化记录。老化周期为300s（±75s）。可通过寄存器3 bit[2]使能或禁止此功能。

### 3.3.5 转发

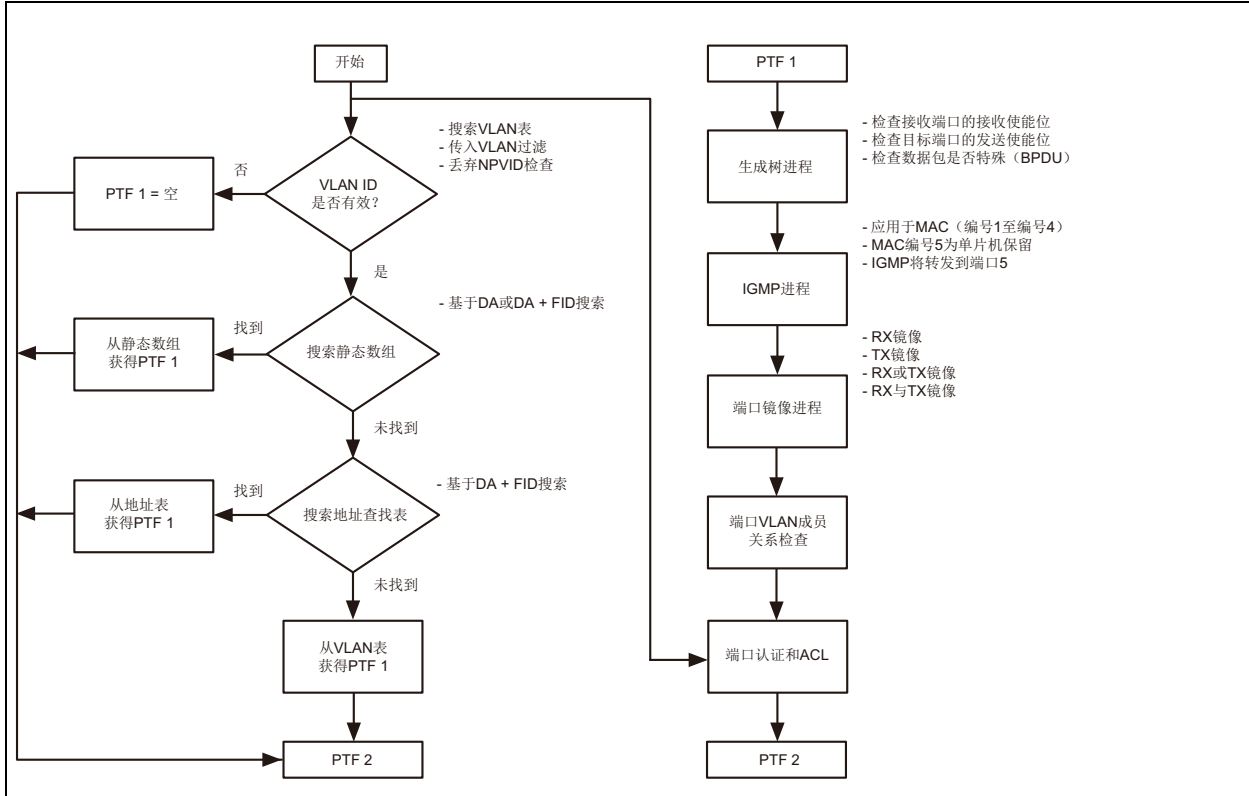
KSZ8795CLX使用以下流程图所述的算法转发数据包。图3-4给出了转发算法的第一阶段，其中搜索引擎在VLAN ID、静态表和动态表中查找目标地址，并得到“转发端口1”（PTF1）。PTF1随后被生成树、IGMP侦听、端口镜像以及端口VLAN过程和认证进一步修改，以得到“转发端口2”（PTF2），如图3-4所示。认证和ACL在转发过程中具有最高优先级；ACL结果将覆盖转发过程的结果。这是数据包将被发送到的地址。

# KSZ8795CLX

KSZ8795CLX不会转发以下数据包：

- 错误数据包。其中包括帧错误、帧校验序列（Frame Check Sequence, FCS）错误、对齐错误以及数据包大小非法错误。
- IEEE802.3x 暂停帧。KSZ8795CLX 拦截这些错误数据包并相应执行全双工流控制。
- “本地”数据包。根据目标地址（Destination Address, DA）查找，如果查找表中的目标端口与数据包来源端口匹配，则将该数据包定义为“本地”数据包。

图3-4： 目标地址查找和解析流程图



### 3.3.6 开关引擎

KSZ8795CLX 具有一个高性能开关引擎，可将数据移入或移出 MAC 的数据包缓冲区。开关引擎在存储和转发模式下工作，其高效的开关机制可降低总延时。KSZ8795CLX 具有 64 kB 的内部帧缓冲区。该资源在全部 5 个端口之间共享。总共有 512 个可用的缓冲区。每个缓冲区的大小为 128 字节。

### 3.4 电源及电源管理

KSZ8795CLX 器件需要 3.3V 模拟电源。外部 1.2V LDO 提供必需的 1.2V 电源以为模拟和数字逻辑内核供电。各个 I/O 的工作电压为 1.8V、2.5V 和 3.3V。表 3-2 给出了器件的不同电压选项和要求。

表3-2： Ksz8795CLX 电压选项和要求

电源信号名称	器件引脚	要求
VDDAT	2、12 和 76	为器件中收发器的模拟模块提供 3.3V 或 2.5V 输入电源。
VDDIO	34、48 和 70	为 I/O 电路选择 1.8V、2.5V 或 3.3V 电源。通过这些输入电源引脚为器件的 I/O 电路供电。

**表3-2: KSZ8795CLX 电压选项和要求 (续)**

电源信号名称	器件引脚	要求
VDD12A	1	1.2V 内核电源。经过滤波的 1.2V 输入电压。这些引脚馈送 1.2V 以为内部模拟和数字内核供电。
VDD12D	26、42 和 73	
GND A	3、21 和 78	模拟地。
GND D	27、33、47、61 和 71	数字地。

KSZ8795CLX 支持低功耗状态下的增强型电源管理以及电能检测，可确保器件空闲期间的功耗较低。电源管理功能下有多种工作模式，由寄存器 14 bit[4:3] 和端口控制 10 寄存器 bit[3] 按如下方式控制：

- 寄存器 14 bit[4:3] = 00：正常工作模式
- 寄存器 14 bit[4:3] = 01：电能检测模式
- 寄存器 14 bit[4:3] = 10：软掉电模式
- 寄存器 14 bit[4:3] = 11：保留

端口控制 10 寄存器 29、45、61 和 77 bit[3] = 1 用于基于端口的掉电模式。表 3-3 指出了所有内部功能模块在四种不同电源管理工作模式下的状态。

**表3-3: 内部功能模块状态**

KSZ8795CLX 功能模块	电源管理工作模式		
	正常模式	电能检测模式	软掉电模式
内部 PLL 时钟	使能	禁止	禁止
TX/RX PHY	使能	RX 端电能检测	禁止
MAC	使能	禁止	禁止
主机接口	使能	禁止	禁止

### 3.4.1 正常工作模式

这是芯片上电或硬件复位后的默认设置，即寄存器 14 bit[4:3] = 00。当 KSZ8795CLX 处于正常工作模式时，所有 PLL 时钟运行，PHY 和 MAC 工作，并且主机接口已准备好进行 CPU 读或写操作。

正常工作模式期间，主机 CPU 可以将寄存器 14 bit[4:3] 置 1，以将当前正常工作模式更改为其他三种电源管理工作模式之一。

### 3.4.2 电能检测模式

当 KSZ8795CLX 端口未连接到有效链路伙伴时，电能检测模式会提供一种比正常工作模式更节能的机制。在该模式下，当电缆拔出时，器件将节省更多电能。如果电缆未插入，器件会自动进入低功耗状态：电能检测模式。在该模式下，器件将保持以每秒 1 个脉冲的速率发送宽度为 120 ns 的脉冲。一旦由于插入电缆或远端尝试建立链路而使活动恢复，器件便会在电能检测模式下自动上电并进入正常功耗状态。

电能检测模式包含两种状态：正常功耗状态和低功耗状态。处于低功耗状态时，器件通过禁止除接收器的电能检测电路以外的所有电路来降低功耗。通过将寄存器 14 bit[4:3] 设置为 01，可进入电能检测模式。当 KSZ8795CLX 处于该模式下时，器件会监视电缆电能。如果电缆上未检测到电能的时间超过在寄存器 15 bit[7:0] 中预先配置的进入休眠时间值，KSZ8795CLX 将进入低功耗状态。当 KSZ8795CLX 处于低功耗状态时，它会持续监视电缆电能。一旦检测到电缆上有电，器件便会进入正常功耗状态。当器件处于正常功耗状态时，它能够通过电缆发送或接收数据包。

### 3.4.3 软掉电模式

通过将寄存器 14 bit[4:3] 设置为 10，可进入软掉电模式。当 KSZ8795CLX 处于软掉电模式下时，将禁止所有 PLL 时钟，同时所有 PHY 和 MAC 停止工作。任何虚拟的主机访问都会将该器件从当前软掉电模式唤醒为正常工作模式，并发出内部复位指令，使所有内部寄存器恢复为默认值。

# KSZ8795CLX

## 3.4.4 基于端口的掉电模式

此外，KSZ8795CLX还具有按端口掉电模式。为节省电能，通过设置端口控制10寄存器bit[3]或MIIM PHY寄存器0 bit[11]，可以使未使用的PHY端口掉电。

## 3.4.5 节能以太网（EEE）

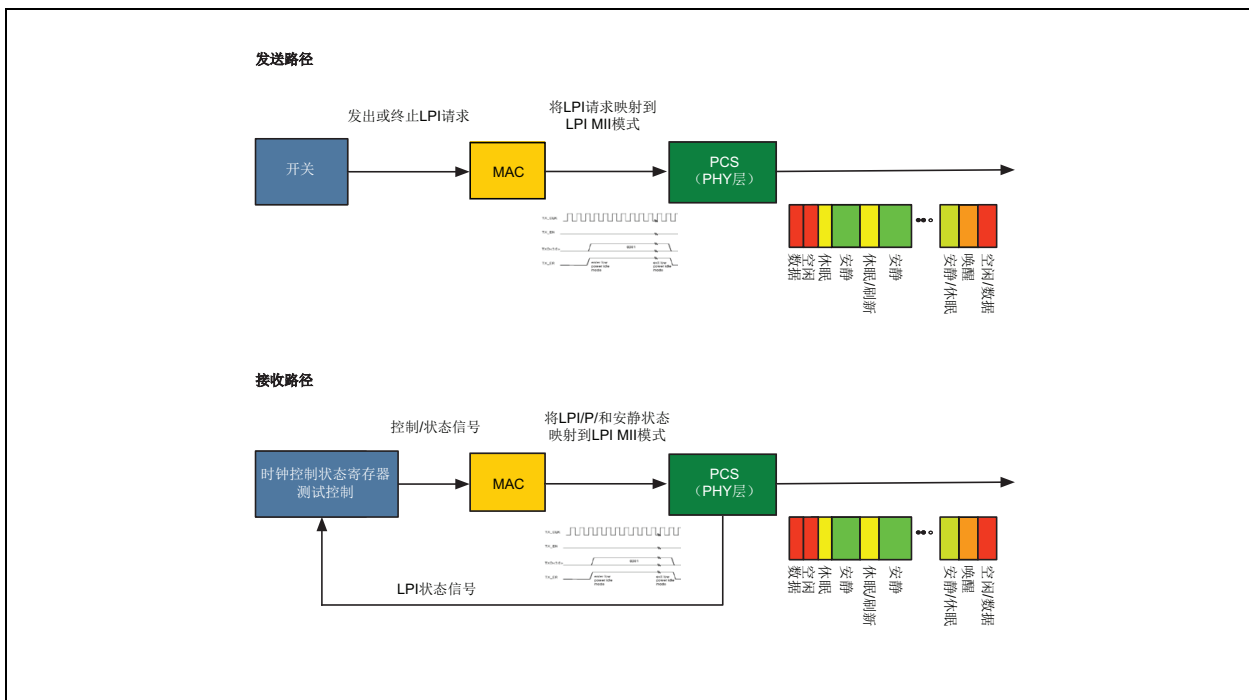
除了支持不同类型的节能模式（硬件掉电、软件掉电和电能检测模式），KSZ8795CLX还支持IEEE P802.3az（2010年3月）中定义的节能以太网功能，从而进一步扩展了其绿色功能的能力。KSZ8795CLX支持10BASE-T和100BASE-TX EEE功能。在100BASE-TX中，EEE在同一链路上非对称工作，即一个方向可能处于低功耗空闲（Low-Power Idle, LPI）状态，同时另一个方向可能存在数据包传输活动。与其他类型的节能模式不同，EEE能够在保持链路的同时实现节能。根据EEE规范，EEE可在PHY级实现节能。KSZ8795CLX不仅可降低PHY级功耗，还可在器件处于低功耗空闲期时尽可能多地关闭未使用的时钟信号，从而降低MAC和开关级功耗。

KSZ8795CLX的10 Mbps和100 Mbps接口均支持IEEE 802.3az标准。EEE功能将开关、MAC和PHY组合在一起，以支持在LPI模式下工作。使能LPI模式时，链路两侧的系统可以在链路使用率较低时实现节能。

EEE实现提供了一种协议，以协调转换到低功耗或从低功耗中退出，同时不改变链路状态且不丢失或损坏帧。进入或退出低功耗的转换时间保持足够小，以便对上层协议和应用透明。EEE规定两个链路伙伴之间交换各自的能力以判断是否支持EEE，并选择两端都能承担的最佳参数组。

除了支持100BASE-TX PHY EEE外，KSZ8795CLX还支持发送幅值要求较低的10BASE-T（针对10 Mbps模式），从而能够降低功耗。

图3-5: IEEE发送和接收信号路径



## 3.4.5.1 LPI 信号传输

LPI 信号传输允许开关向 PHY 和链路伙伴指示数据流中预期将出现中断，开关可利用该信息进入需要额外时间来恢复正常工作的节能模式。当链路伙伴发送此类指示后，LPI 信号也会向开关发出通知。LPI 信号的定义使用 MAC 实现精简的全双工操作（带载波监听延迟）。当 PHY 处于 LPI 模式下时，这可提供全双工操作，但会使用载波监听信号来延迟发送。

开关决定何时向链路伙伴发送 LPI 信号（LPI 请求）并通过 MAC MII 接口传送到 PHY。当链路伙伴发出 LPI 信号，指示 MAC 接口上的 LPI 激活（LPI 指示）时，也会通知开关。实现部分指定了当接收到链路伙伴的 LPI 时开关决定发送 LPI 的条件以及开关会采取哪些措施。

## 3.4.5.2 LPI 有效

未将 LPI 置为有效时，MII 模块上保持正常通信传输。只要将 LPI 请求置为有效，LPI 有效功能便开始在 MII 上发送“将 LPI 置为有效”编码并使 MAC 停止正常发送操作。一旦将 LPI 请求置为无效，LPI 有效功能便重新开始向 MII 上发送正常的帧间编码。经过一段延时后，允许 MAC 重新开始发送。该延时的目的是让链路伙伴为正常工作做好准备。图 3-6 给出了两个活动数据空闲期之间的 EEE LPI。

## 3.4.5.3 LPI 检测

如果接收 MII 上没有“将 LPI 置为有效”编码，LPI 检测功能会将接收 MII 信号映射为正常状态。在 LPI 启动时，即从正常帧间编码转换为接收 MII 上的“将 LPI 置为有效”编码时，LPI 检测功能继续指示接口处于空闲状态，并将 LP\_IDLE 指示置为有效。在 LPI 结束时，即从“将 LPI 置为有效”编码转换为接收 MII 上的任何其他编码时，会将 LP\_IDLE 指示置为无效并恢复正常解码操作。

## 3.4.5.4 PHY LPI 发送操作

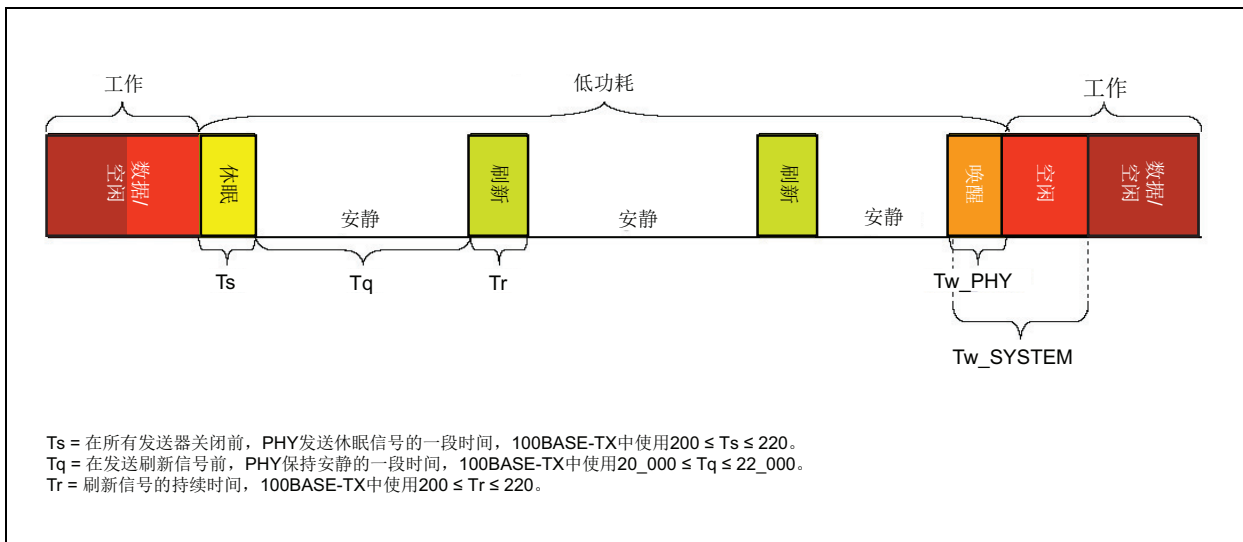
当 PHY 检测到 MII 上“将 LPI 置为有效”编码开始时，PHY 信号会向其链路伙伴发送休眠信号，以指示本地发送器正进入 LPI 模式。EEE 功能要求 PHY 发送器在发出休眠信号后进入安静状态。LPI 请求从链路一端传递到另一端，并且即使 PHY 链路未进入低功耗模式，也能够实现系统节能。

为发送链路伙伴用来更新自适应滤波器和时序电路的刷新信号，要定期使能本地 PHY 的发送功能。这有助于保持链路完整性。这种安静-刷新循环会一直持续，直到在 MII 上接收到正常的帧间编码。PHY 的发送功能通过在预定义的时间段内发送唤醒信号，将此信息传达给链路伙伴。随后 PHY 进入正常工作状态。在正常工作状态与 LPI 模式转换期间，不会丢失或损坏任何数据帧。

在 100BT/全双工 EEE 操作中，刷新发送用于保持链路，安静周期则用于节能。大约每隔 20 ms 至 22 ms，便会向链路伙伴发送一个 200  $\mu$ s 至 220  $\mu$ s 的刷新。图 3-6 显示了刷新发送和安静周期。



图3-6: 通信活动和EEE LPI操作



### 3.4.5.5 PHY LPI接收操作

在接收端，通过接收来自链路伙伴的休眠信号触发进入LPI模式，此信号指示链路伙伴将要进入LPI模式。发送休眠信号后，链路伙伴停止发送。当接收器检测到休眠信号时，本地PHY指示MII上的“将LPI置为有效”，本地接收器可以禁止某些功能以降低功耗。链路伙伴定期地发送本地PHY使用的刷新信号。这种安静-刷新循环一直持续，直到链路伙伴开始通过在由LPI有效功能控制的预定时间段内发送唤醒信号以转换回正常模式时为止。这样，本地接收器便可在MII上为正常工作并从“将LPI置为有效”编码转换为正常帧间编码做好准备。经过系统指定的恢复时间后，链路支持标称的工作数据速率。

### 3.4.5.6 带EEE功能的协商

EEE功能应当在自动协商阶段进行通告。自动协商提供了一个能够检测链路另一端设备支持的功能、确定共有能力并针对联合操作进行配置的已链接设备。当管理员因链路故障或用户干预而发出命令以对器件上电或复位时，将执行自动协商。

自动协商期间，两个链路伙伴都会指示其EEE功能。自动协商期间，仅当本地设备和链路伙伴均针对已解析的PHY类型通告其EEE功能时，才支持EEE。如果不支持EEE，则所有EEE功能都会被禁止，并且LPI客户端不会将LPI置为有效。针对已协商的PHY类型，如果两个链路伙伴均支持EEE，则可在任一方向上独立使用EEE功能。

### 3.4.6 LAN唤醒（WOL）

LAN唤醒（WoL）允许通过网络消息打开或唤醒计算机。消息通常由同一局域网中另一台计算机上执行的程序发出。只要在网络上将有意义的数据提交给系统，便会使用唤醒帧事件唤醒系统。有意义的数据示例包括Magic Packet™（魔术包）接收、远程管理员发出的管理请求或者仅仅是直接以本地系统为目标的网络通信。经过编程，KSZ8795CLX可以将唤醒帧检测通知给主机，这种检测可将中断信号（INTR\_N）或电源管理事务信号（PME）置为有效。PME由PME间接寄存器控制。

KSZ8795CLX MAC支持检测以下唤醒事件：

- 电信号超过预配置值检测：PME间接寄存器中的端口PME控制状态寄存器bit[0]。
- 网络链路状态接通检测：PME间接寄存器中的端口PME控制状态寄存器bit[1]。
- 魔术包接收：PME间接寄存器中的端口PME控制状态寄存器bit[2]。



此外，还有此处未列出的其他类型唤醒事件，因为制造商可能按照自己的方式来实现这些唤醒事件。

### 3.4.6.1 电能流向

电能将在电缆上进行检测并在超过预配置值的时间内持续呈现，尤其是在电能变化可能影响到系统应当重新进入正常功耗状态所需的电平时。

### 3.4.6.2 链路接通方向

链路状态唤醒事件对于指示网络连接状态中的链路接通非常有用。

### 3.4.6.3 魔术包

魔术包是一个包含任意位置的广播帧，其有效数据载荷为6个字节的全1 (FF FF FF FF FF FF)，后面重复16次目标计算机的48位DA MAC地址。由于只扫描魔术包中的上述字符串，而实际上并未通过完整的协议栈解析魔术包，因此可将魔术包作为任意网络层和传输层协议进行发送。

魔术包技术用于远程唤醒LAN上已休眠或关机的PC。这可通过向网络上某个节点发送称为魔术包帧的特定信息数据包来实现。当能够接收特定帧的PC进入休眠状态时，PC会在LAN控制器中使能魔术包接收模式；当LAN控制器接收到魔术包帧时，它会提醒系统唤醒。一旦KSZ8795CLX通过PME间接寄存器中的端口PME控制屏蔽寄存器bit[2]使能了魔术包检测，它便会以特定数据序列扫描寻址到该节点的所有传入帧，此序列向控制器指示这是魔术包帧。

魔术包帧还必须满足所选LAN技术的基本要求，例如源地址 (SA)、目标地址 (DA)，这些地址可能是接收站点的IEEE MAC地址或者多播或广播地址和CRC。该特定序列由该节点重复16次的MAC地址组成，无间断或中断。该序列可位于数据包内的任意位置，但前面必须是同步流。该同步流定义为6个字节的0xFF。只要重复16次的IEEE地址与待唤醒机器的地址匹配，器件也会接收广播帧。

#### 魔术包示例：

如果网络上特定节点的IEEE地址为11h 22h 33h 44h 55h 66h，则LAN控制器会扫描下面的数据序列（假定是以太网帧）：

```
DA - SA - TYPE - FF FF FF FF FF FF - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - 11 22 33 44 55 66 - MISC-CRC。
```

对于魔术包帧，没有其他限制。例如，该序列可以位于TCP/IP数据包或IPX数据包中。可以在网络中对该帧进行桥接或路由，而不影响其唤醒帧目标节点的能力。如果扫描未找到上面所示的特定序列，则丢弃该帧并且不采取任何操作。不过，如果KSZ8795CLX检测到数据序列，则会提醒PC的电源管理电路（将PME引脚置为有效）唤醒系统。

### 3.4.7 中断 (INT\_N/PME\_N)

INT\_N是一个中断信号，用于通知外部控制器KSZ8795CLX中断状态寄存器中存在状态更新。寄存器125 bit[3:0]是中断屏蔽控制位，用于使能和禁止将INT\_N信号置为有效的条件。寄存器124 bit[3:0]是中断状态位，用于指示哪些中断条件已发生。寄存器124中的中断状态位在被读取后清零。

PME\_N是一个可选PME中断信号，用于通知外部控制器KSZ8795CLX中断状态寄存器中存在状态更新。寄存器125 bit[4]是PME中断屏蔽控制位，用于使能和禁止将PME\_N信号置为有效的条件。寄存器124 bit[4]是PME中断状态位，用于指示哪些PME中断条件已发生。寄存器124中的PME中断状态位bit[4]在被读取后清零。

此外，INT\_N和PME\_N的中断引脚无需处理器查询开关是否存在状态变化。

# KSZ8795CLX

## 3.5 接口

KSZ8795CLX 器件集成了大量接口，使其非常适合标准网络环境以及供应商的独特环境。表 3-4 总结了这些可用的接口。随后的部分给出了表中各接口用途的详细信息。

表 3-4: 可用接口

接口	类型	用途	访问的寄存器
SPI	配置和寄存器访问	[作为从串行总线]——外部 CPU 或控制器可以通过该接口读/写所有内部寄存器。	全部
MIIM	配置和寄存器访问	具有 MDC/MDIO 的 CPU 或控制器可以读/写 4 个 PHY 寄存器。	仅 PHY
GMII	数据流	到端口 5 GMAC 的接口，采用标准 GMII 时序。	N/A
MII	数据流	到端口 5 GMAC 的接口，采用标准 MII 时序。	N/A
RGMII	数据流	到端口 5 GMAC 的接口，采用更快的精简 GMII 时序。	N/A
RMII	数据流	到端口 5 GMAC 的接口，采用更快的精简 MII 时序。	N/A

### 3.5.1 配置接口

#### 3.5.1.1 SPI 从串行总线配置

KSZ8795CLX 也可以用作 SPI 从器件。通过 SPI，可以使能整个功能集，包括“VLAN”、“IGMP 侦听”以及“MIB 计数器”等。外部 SPI 主器件可以随机访问数据手册中显示的任意寄存器。SPI 模式可以配置所有所需的设置，包括间接寄存器和表。KSZ8795 默认处于“启动开关”模式，寄存器 1 bit[0] = 1；要禁止开关，将 0 写入寄存器 1 bit[0]。

支持 2 个标准 SPI 命令（00000011 用于“读数据”，00000010 用于“写数据”）。为缩短配置时间，KSZ8795CLX 还支持多重读或写操作。对 KSZ8795CLX 进行一个字节的读写操作后，如果 SPI 从选择信号（SPIS\_N）持续驱动为低电平，则内部地址计数器会自动递增。如果读取第一个字节后 SPIS\_N 仍保持低电平，则位于下一个地址的下一个字节将从 SPIQ 上移出。如果写入第一个字节后 SPIS\_N 仍保持低电平，则主器件输出从器件输入（SPID）线上的位将被写入下一个地址。将 SPIS\_N 置为高电平会终止读或写操作。这意味着在发送另一个命令和地址前，必须先将 SPIS\_N 信号置为高电平，然后再置为低电平。达到最高地址后，地址计数器会绕回至零。因此，通过发送单个命令和地址可以对整个寄存器组进行读写操作。

KSZ8795CLX 能够支持最高 50 MHz 的 SPI 总线。推荐使用高性能 SPI 主器件，以避免内部计数器溢出。

要使用 KSZ8795CLX SPI：

1. 在电路板级，按表 3-5 中的详细说明连接 KSZ8795CLX 引脚。
2. 利用下拉电阻来下拉引脚 SPIQ，可将串行通信配置为 SPI 从模式。
3. 按照图 3-7 中所示的典型 SPI 写数据周期或图 3-8 中所示的 SPI 多重写操作将配置数据写入寄存器。请注意，SDA 上的数据输入在 SCL 时钟的上升沿寄存。
4. 寄存器可以被读取并且配置可以通过图 3-7 中所示的典型 SPI 数据读取周期或图 3-8 中所示的多重 SPI 数据读取周期进行验证。请注意，SPIQ 中的读数据将在 SCL 时钟的下降沿寄存。

表 3-5: SPI 连接

KSZ8795CLX 信号名称	微处理器信号说明
SPIS_N (S_CS)	SPI 从选择
SCL (S_CLK)	SPI 时钟
SDA (S_DI)	主输出。从输入。
SPIQ (S_DO)	主输入。从输出。

图3-7: SPI访问时序

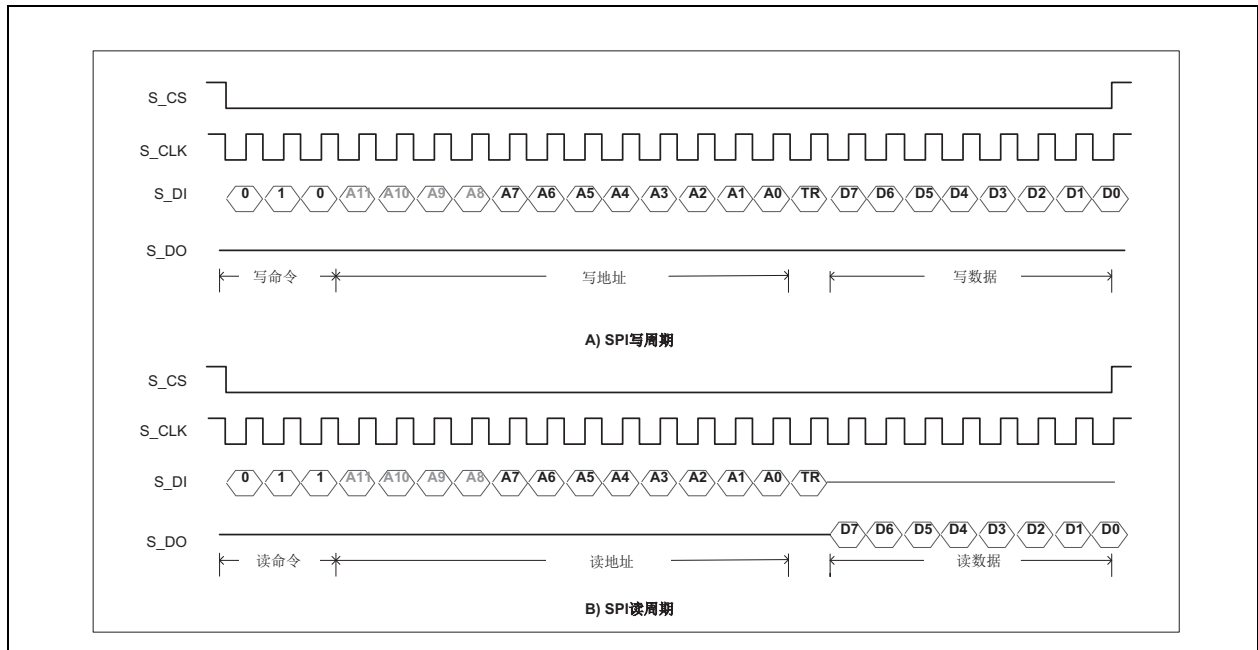
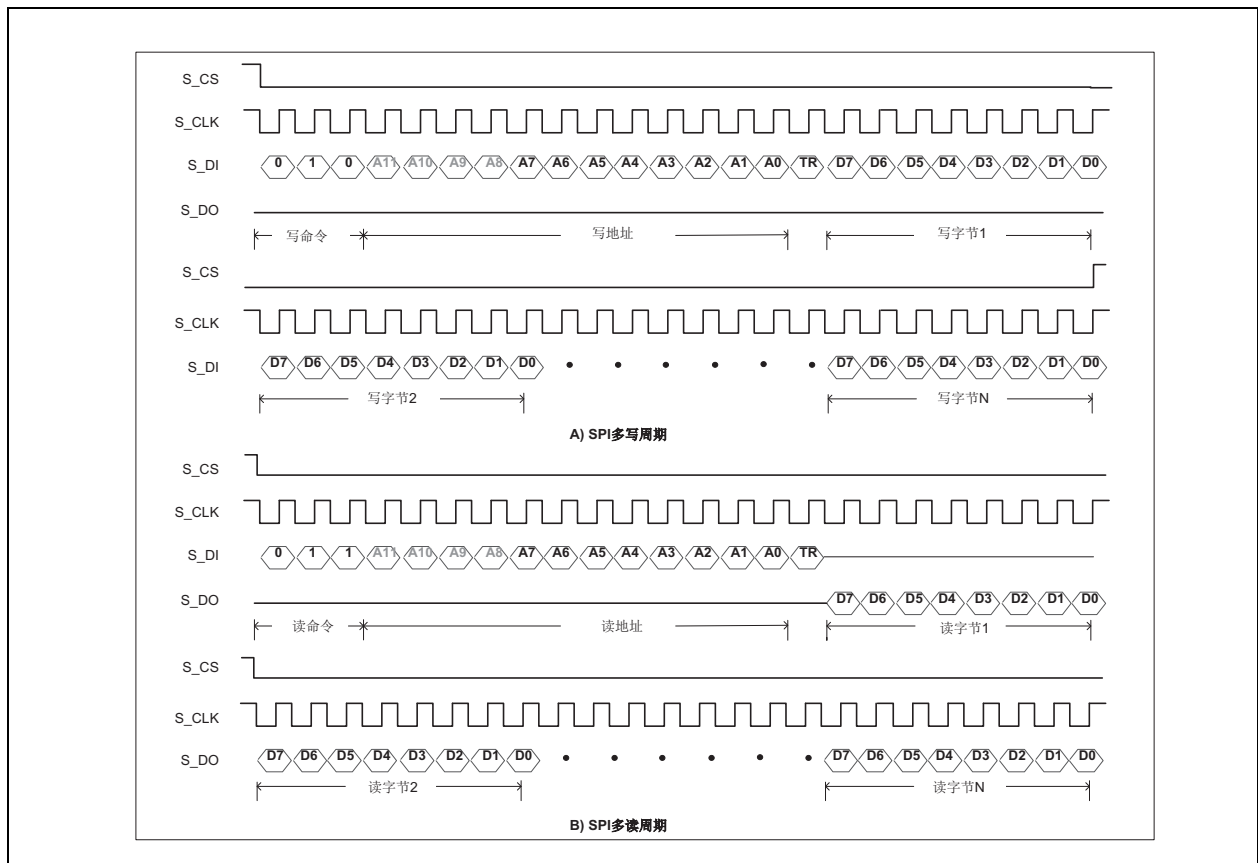


图3-8: SPI多重访问时序



# KSZ8795CLX

## 3.5.1.2 MII管理接口 (MIIM)

KSZ8795CLX支持标准IEEE 802.3 MII管理接口（也称做管理数据输入/输出（Management Data Input/Output, MDIO）接口）。该接口允许上层器件监视和控制KSZ8795CLX的状态。具有MDC/MDIO功能的外部器件用于读取PHY状态或配置PHY设置。有关MIIM接口的更多详细信息，请参见IEEE 802.3u规范。

MIIM接口的组成部分如下：

- 一个兼具数据线MDIO和时钟线MDC的物理连接。
- 一个在上述物理连接上工作，允许外部控制器与KSZ8795CLX器件通信的特定协议。
- 访问一组共8个16位寄存器，由8个标准MIIM寄存器[0:5h]组成，每个端口都有1d和1f MIIM寄存器。

MIIM接口MDC/MDIO可以支持的最大MDC时钟速度达25 MHz。

表3-6给出了MII管理接口帧格式。

表3-6: MII管理接口帧格式 (注3-1)

	前导码	帧起始	读/写操作码	PHY地址 Bit[4:0]	REG地址 Bit[4:0]	TA	数据 Bit[15:0]	空闲
读	32个1	01	10	AAAAA	RRRRR	Z0	DDDDDDDD_DDDDDDDD	Z
写	32个1	01	01	AAAAA	RRRRR	10	DDDDDDDD_DDDDDDDD	Z

注3-1 前导码——由32个1组成

帧起始——“01”模式表示帧起始。该模式确保线路状态从默认逻辑1到0并回到1的跳变。

读/写操作码——读事务的操作码为“10”，写事务的操作码为“01”。

PHY地址 Bit[4:0]——PHY地址为5位，支持32个惟一的PHY地址。发送和接收的第一个PHY地址位是地址的MSB。

寄存器地址 Bit[4:0]——寄存器地址为5位，支持在每个PHY中寻址32个独立寄存器。发送和接收的第一个寄存器地址位是地址的MSB。

TA (周转)——周转时间为帧的寄存器地址字段与数据字段之间的2位时间间隔，用于避免在读事务期间发生争用。对于读事务，主器件和PHY在周转的第一个位时间内均应保持高阻态。在读事务周转的第二个位时间内，PHY将驱动一个“0”位。在写事务期间，主器件应在周转的第一个和第二个位时间内分别驱动一个“1”位和一个“0”位。

数据 Bit[15:0]——数据字段为16位。发送和接收的第一个数据位应为正被寻址的寄存器的bit[15]。

在每个事务开始时，主器件应在MDIO上发送由32个连续的逻辑“1”位组成的序列，使用MDC上32个相应的周期作为时钟信号，为器件提供可用于建立同步的模式。器件在观察到MDIO上32个连续的“1”位序列以及MDC上的32个相应周期后，才开始响应任意事务。

MIIM接口没有访问KSZ8795CLX中所有配置寄存器的权限。它仅能访问标准MIIM寄存器（见MIIM寄存器部分）。另一方面，SPI接口可用于访问KSZ8795CLX功能集中的所有寄存器。

## 3.5.2 开关端口5 GMAC接口

KSZ8795CLX GMAC5接口支持GMII/MII/RGMII/RMII四种接口协议并共享一组输入/输出信号。该接口的作用是在GMAC/MAC子层与GPHY/PHY之间提供一种简单、低成本且易于实现的互连。这些接口上的数据帧采用IEEE以太网标准。因此，它包含前导码、帧开始分隔符、以太网报头、协议特定数据以及循环冗余校验（CRC）校验和。

表3-7给出了GMII/MII/RGMII/RMII接口的发送和接收信号。

**表3-7: GMII/RGMII/MII/RMII的信号**

方向类型	GMII	RGMII	MII	RMII
输入（输出）	GTXC	GTXC	TXC	REFCLKI
输入	TXER	—	TXER	—
输入	TXEN	TXD_CTL	TXEN	TXEN
输入（输出）	COL	—	COL	—
输入	TXD[7:0]	TXD[3:0]	TXD[3:0]	TXD[1:0]
输入（输出）	GRXC	GRXC	RXC	RXC
输出	RXER	—	RXER	RXER
输出	RXDV	RXD_CTL	RXDV	CRS_DV
输入（输出）	CRS	—	CRS	—
输出	RXD[7:0]	RXD[3:0]	RXD[3:0]	RXD[1:0]

### 3.5.2.1 标准 GMII/MII 接口

GMII 和 MII 接口能够支持 10/100 Mbps 和 1000 Mbps 操作。数据和分隔符同步到时钟参考。它提供独立的 4/8 位宽发送和接收数据路径，并利用信号电平提供两个介质状态信号。CRS 指示存在载波信号，COL 指示发生冲突。MII 提供半双工和全双工操作，GMII 使用全双工操作。

GMII 基于 MII。它保留了 MII 接口的信号名称并且二者的大部分信号功能都相同，但针对 1000 Mbps 操作定义了附加有效信号组合。GMII 仅支持 1000 Mbps 操作。MII 接口支持 10 Mbps 和 100 Mbps 操作。

MII 在各方向上使用 4 位字（半字节）传送数据，其时钟频率为 2.5/25 MHz，可达到 10/100 Mbps 速度。GMII 在各方向上使用 8 位字（半字节）传送数据，其时钟频率为 125 MHz，可达到 1000 Mbps 速度。

### 3.5.2.2 精简的千兆位介质无关接口（RGMII）

RGMII 适合作为 IEEE802.3u MII 和 IEEE802.3z GMII 的备用接口。其主要目标是以经济高效和技术独立的方式减少 GMAC 和 GPHY 互连所需的引脚数。为实现此目标，将精简数据路径和所有关联控制信号，同时复用控制信号并使用时钟的两个边沿。对于千兆位操作，时钟将在 125 MHz 速率下工作，并在上升沿和下降沿锁存数据。

### 3.5.2.3 精简的介质无关接口（RMII）

精简的介质无关接口（RMII）是低引脚数的介质无关接口（MII）。KSZ8795CLX 的端口 5 GMAC5 支持 RMII 接口并提供以下关键特性：

- 支持 10 Mbps 和 100 Mbps 数据速率。
- 采用单个 50 MHz 参考时钟（内部或外部提供）：在内部模式下，芯片通过 RXC5 引脚将参考时钟提供给 RMII 接口的相对时钟输入引脚。在外部模式下，芯片从外部振荡器或相对的 RMII 接口接收 50 MHz 参考时钟。
- 提供独立的 2 位宽（双位）发送和接收数据路径。

### 3.5.2.4 端口 5 GMAC5 SW5-MII 接口

表 3-8 给出了两种连接方法。

1. 第一种是 SW5-MII PHY 模式下的外部 MAC 连接。
2. 第二种是 SW5-MII MAC 模式下的外部 PHY 连接。

配置脚引脚 62 LED2\_1 确定是 MAC 模式还是 PHY 模式设置。

# KSZ8795CLX

表3-8: 端口5 SW5-MII连接

MAC到MAC连接 KSZ8795CLX SW5-MII PHY 模式			说明	MAC到PHY连接 KSZ8795CLX SW5-MII PHY 模式		
外部MAC	KSZ8795CLX SW5-MII信号	类型		外部PHY	KSZ8795CLX SW5-MII信号	类型
MTXEN	TXEN5	输入	发送使能	MTXEN	RXDV5	输出
MTXER	TXER5	输入	发送错误	MTXER	RXER5	输出
MTXD[3:0]	TXD5[3:0]	输入	发送数据位 [3:0]	MTXD[3:0]	RXD5[3:0]	输出
MTXC	TXC5	输出	发送时钟	MTXC	RXC5	输入
MCOL	COL5	输出	冲突检测	MCOL	COL5	输入
MCRS	CRS5	输出	载波监听	MCRS	CRS5	输入
MRXDV	RXDV5	输出	接收数据有效	MRXDV	TXEN5	输入
MRXER	RXER5	输出	接收错误	MRXER	TXER5	输入
MRXD[3:0]	RXD5[3:0]	输出	接收数据位 [3:0]	MRXD[3:0]	TXD5[3:0]	输入
MRXC	RXC5	输出	接收时钟	MRXC	TXC5	输入

MII接口可在MAC模式或PHY模式下工作。这些接口采用半字节宽的数据接口，因此其运行速率为网络位速率（未编码）的四分之一。发送端的附加信号指示数据何时有效或者传输期间何时发生了错误。同样，接收端的附加信号指示数据何时有效且无物理层故障。对于半双工操作，有COL信号来指示传输期间发生了冲突。

**注：**通常情况下，MRXER指示来自物理层器件的接收错误。MTXER指示MAC器件的发送错误。这些信号不适用于该配置。对于带外部MAC的PHY模式操作，如果与KSZ8795CLX接口的器件具有MRXER引脚，则可将该引脚连接到低电平。对于带外部PHY的MAC模式操作，如果与KSZ8795CLX接口的器件具有MTXER引脚，则可将该引脚连接到低电平。

### 3.5.2.5 端口5 GMAC5 SW5-GMII接口

表3-9给出了将GMII连接到外部GMAC或GPHY的两种方法。

- 第一种是SW5-GMII GPHY模式下的外部GMAC连接。
- 第二种是SW5-GMII GMAC模式下的外部GPHY连接。

配置脚引脚62 LED2\_1确定是GMAC模式还是GPHY模式。

表3-9: 端口5 SW5-GMII连接

GMAC到GMAC连接 KSZ8795CLX SW5-GMII GPHY 模式			说明	GMAC到GPHY连接 KSZ8795CLX SW5-GMII GMAC 模式		
外部GMAC	KSZ8795CLX SW5-GMII信号	类型		外部GPHY	KSZ8795CLX SW5-GMII信号	类型
MRXDV	TXEN5	输入	发送使能	MTXEN	RXDV5	输出
MRXER	TXER5	输入	发送错误	MTXER	RXER5	输出
MRXD[7:0]	TXD5[7:0]	输入	发送数据位 [7:0]	MTXD[7:0]	RXD5[7:0]	输出
MGRXC	GTXC5	输入	发送时钟	MGTXC	GRXC5	输出

**表3-9: 端口5 SW5-GMII连接 (续)**

GMAC到GMAC连接 KSZ8795CLX SW5-GMII GPHY模式			说明	GMAC到GPHY连接 KSZ8795CLX SW5-GMII GMAC模式		
外部GMAC	KSZ8795CLX SW5-GMII信号	类型		外部GPHY	KSZ8795CLX SW5-GMII信号	类型
MCOL	COL5	输出	冲突检测	MCOL	COL5	输入
MCRS	CRS5	输出	载波监听	MCRS	CRS5	输入
MRXEN	RXDV5	输出	接收数据有效	MRXDV	TXEN5	输入
MTXER	RXER5	输出	接收错误	MRXER	TXER5	输入
MRXD[7:0]	RXD5[7:0]	输出	接收数据位 [7:0]	MRXD[7:0]	TXD5[7:0]	输入
MGTXC	GRXC5	输出	接收时钟	MGRXC	GTXC5	输入

端口5 GMAC5 SW5-GMII接口的工作速率最高达1000 Mbps。在1 Gbps模式下，GMII仅支持全双工。GMII接口在各方向上使用8位数据。发送端的附加信号指示数据何时有效或者传输期间何时发生了错误。同样，接收端的附加信号指示数据何时有效且无物理层故障。对于10/100 Mbps模式下的半双工操作，有COL信号来指示传输期间发生了冲突。

### 3.5.2.6 端口5 GMAC5 SW5-RGMII接口

表3-10给出了将RGMII连接到外部GMAC或GPHY的精简方法。

**表3-10: 端口5 SW5-RGMII连接**

KSZ8795CLX SW5-RGMII连接			说明
外部GMAC/GPHY	KSZ8795CLX SW5-RGMII信号	类型	
MRX_CTL	TXD5_CTL	输入	发送控制
MRXD[3:0]	TXD5[3:0]	输入	发送数据位 [3:0]
MRX_CLK	GTX5_CLK	输入	发送时钟
MTX_CLK	RXD5_CTL	输出	接收控制
MTXD[3:0]	RXD5[3:0]	输出	接收数据位 [3:0]
MGTX_CLK	GRXC5	输出	接收时钟

RGMII接口可在最高1000 Mbps速率下工作。附加的发送和接收信号控制不同的数据传输方向。该RGMII接口支持RGMII版本2.0，可通过寄存器86 (0x56) 调节传入时钟及传出时钟延时。

要为连接伙伴正确配置RGMII，需要正确设置寄存器86 (0x56) bit[4:3]。表3-11给出了配置表。

**表3-11: 适合连接伙伴的端口5 SW5-RGMII时钟延时配置**

KSZ8795CLX 寄存器86 bit[4:3]配置	RGMII时钟模式 (接收和发送)	KSZ8795CLX 寄存器86 (0x56)	KSZ8795CLX RGMII 时钟延时/压摆率配置	连接伙伴RGMII 时钟配置 (注3-1)
bit[4:3] = 11 模式	传入时钟输入	bit[4] = 1	延时	无延时
	传出时钟输出	bit[3] = 1	延时	无延时
bit[4:3] = 10 模式	传入时钟输入	bit[4] = 1	延时	无延时
	传出时钟输出	bit[3] = 1	无延时	延时



# KSZ8795CLX

表3-11: 适合连接伙伴的端口5 SW5-RGMII时钟延时配置 (续)

KSZ8795CLX 寄存器86 bit[4:3]配置	RGMII时钟模式 (接收和发送)	KSZ8795CLX 寄存器86 (0x56)	KSZ8795CLX RGMII 时钟延时/压摆率配置	连接伙伴RGMII 时钟配置 (注3-1)
bit[4:3] = 01 模式	传入时钟输入	bit[4] = 0 (默认)	无延时	延时
	传出时钟输出	bit[3] = 0 (默认)	延时	无延时
bit[4:3] = 00 模式	传入时钟输入	bit[4] = 0	无延时	延时
	传出时钟输出	bit[3] = 0	无延时	延时

注3-1 对于带RGMII的处理器，需要外部GPHY或KSZ8795CLX背对背连接。

例如，两个KSZ8795器件采用背对背连接。如果将一个器件的bit[4:3]设置为11，则应将另一个器件的bit[4:3]设置为00。如果将一个器件的bit[4:3]设置为01，则应将另一个器件的bit[4:3]也设置为01。

RGMII模式通过配置脚引脚LED3[1:0] = 11 (默认) 或寄存器86 (0x56) bit[1:0] = 11 (默认) 来配置。通过配置脚引脚LED1\_0或寄存器86 (0x56) bit[6]选择速度，默认速度为1 Gbps (bit[6] = 1)，在RGMII模式下，设置bit[6] = 0可选择10/100 Mbps速度。KSZ8795CLX通过寄存器86 bit[4:3]调节时钟延时以及通过寄存器164 bit[6:4]调节驱动强度，可在1 Gbps模式下获得电路板级的最佳RGMII时序。

### 3.5.2.7 端口5 GMAC5 SW5-RMII接口

RMII指低引脚数MII。KSZ8795CLX的端口5支持RMII接口并提供以下关键特性：

- 支持10 Mbps和100 Mbps数据速率。
- 采用单个50 MHz参考时钟（内部或外部提供）：在内部模式下，当端口5 RMII被设为时钟模式时，芯片会通过RXC5引脚将参考时钟提供给RMII接口的相对时钟输入引脚。
- 在外部模式下，当器件被设为正常模式时，芯片的TXC5/REFCLKI5引脚从外部振荡器或相对的RMII接口接收50 MHz参考时钟。
- 提供独立的2位宽（双位）发送和接收数据路径。

有关SW5-RMII（端口5 GMAC5 RMII）信号连接的详细信息，请参见表3-12。

当器件通过配置脚进入正常模式时，参考时钟来自TXC5/REFCLKI5引脚，并用作器件的时钟源。设置配置脚引脚LED1\_1可选择器件的时钟源是来自TXC5/REFCLKI5引脚还是来自XI/XO引脚上的外部25 MHz晶振/振荡器时钟。

在内部模式下，当使用内部50 MHz时钟作为SW5-RMII参考时钟时，应通过配置脚引脚LED2\_1或端口寄存器86 bit[7]将KSZ8795CLX端口5设置为时钟模式。KSZ8795CLX器件的时钟模式会将50 MHz参考时钟提供给端口5 RMII接口。

在外部模式下，当使用外部50 MHz时钟源作为SW5-RMII参考时钟时，应通过配置脚引脚LED2\_1或端口寄存器86 bit[7]将KSZ8795CLX端口5设置为正常模式。当KSZ8795CLX器件的TXC5/REFCLKI5引脚从外部50 MHz时钟源接收50 MHz参考时钟时，其正常模式将开始工作。



表3-12: 端口5 SW5-RMII连接

SW5-RMII MAC到MAC连接 (PHY模式)			说明	SW5-RMII MAC到PHY连接 (MAC模式)		
外部MAC	KSZ8795CLX SW5-RMII信号	类型		外部PHY	KSZ8795CLX SW5-RMII信号	类型
REF_CLKI	RXC5	时钟模式下 输出 50 MHz	参考时钟	50 MHz	REFCLKI5	正常模式下 输入 50 MHz
CRS_DV	RXDV5/ CRSDV5	输出	载波监听/ 接收数据有效	CRS_DV	TXEN5	输入
—	—	—	接收错误	RXER	TXER5	输入
RXD[1:0]	RXD5[1:0]	输出	接收数据位 [1:0]	RXD[1:0]	TXD5[1:0]	输入
TX_EN	TXEN5	输入	发送数据使能	TX_EN	RXDV5/ CRSDV5	输出
TXD[1:0]	TXD5[1:0]	输入	发送数据位 [1:0]	TXD[1:0]	RXD[1:0]	输出
50 MHz	REFCLKI5	正常模式下 输入 50 MHz	参考时钟	REF_CLKI	RXC5	时钟模式下 输出 50 MHz

## 3.6 高级功能

### 3.6.1 QOS 优先级支持

KSZ8795CLX为VoIP和视频会议等应用提供服务质量(QoS)优先级功能。KSZ8795CLX通过设置端口控制9寄存器bit[1]和端口控制0寄存器bit[0]为每个端口提供1个、2个或4个优先级队列，1/2/4个队列划分如下：

- [端口控制9寄存器bit[1], 控制0 bit[0]] = 00: 单一输出队列(默认)。
- [端口控制9寄存器bit[1], 控制0 bit[0]] = 01: 可将传出端口划分为2个优先级发送队列。
- [端口控制9寄存器bit[1], 控制0 bit[0]] = 10: 可将传出端口划分为4个优先级发送队列。

4个优先级发送队列是KSZ8795CLX中的新功能。队列3为最高优先级队列，队列0为最低优先级队列。端口控制9寄存器bit[1]和端口控制0寄存器bit[0]分别用于使能端口1、2、3、4和5的划分发送队列。如果某个端口的发送队列未划分，则高优先级和低优先级数据包在发送队列中具有相同的优先级。

此外，还有一个附加选项，或者始终首先发送高优先级数据包，或者通过端口控制14、15、16和17寄存器(按照bit[6:0]，默认值为8、4、2和1)针对4个优先级队列比例使用可编程加权公平队列。

选择2队列配置时，将使用寄存器130 bit[7:6] Prio\_2Q[1:0]。这些位用于将IEEE 802.1p的2位结果从寄存器128和129或者TOS/DiffServ的2位结果从寄存器144-159(对于4个队列)映射到具有高优先级或低优先级的2队列模式中。

有关详细信息，请参见寄存器130 bit[7:6]的说明。

#### 3.6.1.1 基于端口的优先级

对于基于端口的优先级，每个传入端口分别归类为优先级0-3的接收端口。优先级3接收端口接收的所有数据包标记为高优先级，并将被发送到高优先级发送队列(如果已划分相应的发送队列)。端口控制0寄存器bit[4:3]用于分别使能端口1、2、3、4和5的基于端口的优先级。

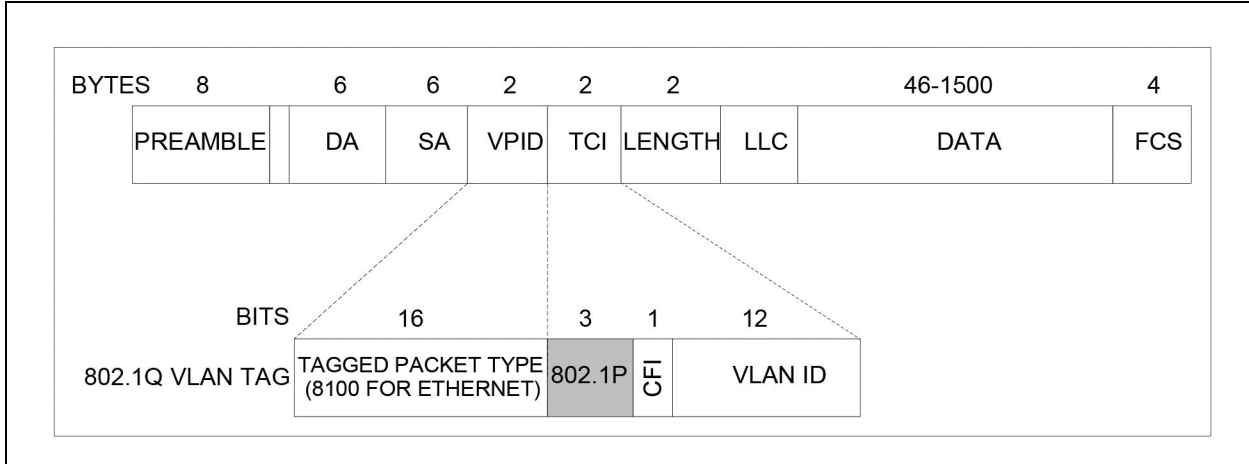
# KSZ8795CLX

## 3.6.1.2 基于802.1p的优先级

对于基于802.1p的优先级，KSZ8795CLX会检查传入数据包以确定其是否带标记。如果带标记，则检索VLAN标记中的3位优先级字段，并与根据寄存器128和129指定的“优先级映射”值比较，寄存器128和129可将值0-7的3位优先级字段映射到0-3优先级的2位结果中。“优先级映射”值可编程。

图3-9演示了如何将802.1p优先级字段嵌入802.1Q VLAN标记。

图3-9: 802.1P 优先级字段格式



端口1、2、3、4和5的端口控制0寄存器bit[5]分别使能基于802.1p的优先级。

KSZ8795CLX提供在各独立传出端口上插入或删除带优先级标记的帧的报头。此报头由2字节的VLAN协议ID（VLAN Protocol ID, VPID）和2字节的标记控制信息（Tag Control Information, TCI）字段组成，也称为IEEE 802.1Q VLAN标记。

通过端口控制0寄存器bit[2]使能标记插入，通过端口控制8寄存器选择将哪个源端口（传入端口）的VPID分别插入端口1、2、3、4和5的传出端口。在传出端口中，使用传入端口的默认标记来标记无标记数据包。默认标记分别在端口1、2、3、4或5的端口控制3和控制4寄存器中进行编程。KSZ8795CLX不会向已带标记数据包添加标记。

端口1、2、3、4或5的端口控制0寄存器bit[1]分别使能标记删除。在传出端口中，带标记数据包的802.1Q VLAN标记将被删除。KSZ8795CLX不会修改无标记数据包。

无论是标记插入还是标记删除，都会重新计算CRC。

802.1p 优先级字段重映射是一种QoS功能，允许KSZ8795CLX通过端口控制2寄存器bit[7]在任意传入端口中设置“用户优先级上限”。如果传入数据包优先级字段中的优先级值高于传入端口默认标记优先级字段中的优先级值，则使用默认标记优先级字段来替换数据包的优先级字段。

## 3.6.1.3 基于DiffServ的优先级

基于DiffServ的优先级使用“高级控制寄存器”小节中的ToS寄存器（寄存器144至寄存器159）。ToS优先级控制寄存器实现了一个完全解码的128位差分服务代码点（Differentiated Services Code Point, DSCP）寄存器，用来根据IP报头中的6位ToS字段确定数据包优先级。将ToS字段的高6位完全解码时，将为DSCP生成64个代码点。这些代码点将与DSCP寄存器中的相应位进行比较以确定优先级。

## 3.6.2 生成树支持

端口5是支持生成树的指定端口。

通过分别对应于端口1、2、3和4的寄存器18、34、50和66中的“发送使能”、“接收使能”和“学习禁止”寄存器设置，可以将其他端口（端口1至端口4）配置为五种生成树状态之一。下面介绍针对5种生成树状态中的每一种状态采取的端口设置和软件操作。

KSZ8795CLX支持公共生成树（Common Spanning Tree, CST）。为支持生成树，主机端口（端口5）是处理器的指定端口。通过端口控制2寄存器中的“发送使能”、“接收使能”和“学习禁止”寄存器设置，可以将其他端口配置为五种生成树状态之一。表3-13给出了针对5种生成树状态中的每一种状态采取的端口设置和软件操作。

**表3-13： 生成树的端口设置和软件操作**

禁止状态	端口设置	软件操作
端口不应转发或接收任何数据包。禁止学习功能。	“发送使能 = 0， 接收使能 = 0， 学习禁止 = 1。”	处理器不应向端口发送任何数据包。开关可能仍向处理器发送特定数据包（与静态表中的一些条目匹配且“改写位”置1的数据包），处理器应将这些数据包丢弃。注：处理器通过MII模块连接到端口5。该状态下禁止端口的地址学习功能。
<b>阻塞状态</b>	<b>端口设置</b>	<b>软件操作</b>
仅转发发送到处理器的数据包。禁止学习功能。	“发送使能 = 0， 接收使能 = 0， 学习禁止 = 1。”	该状态下处理器不向端口发送任何数据包。处理器应将需要接收的条目（如BPDU数据包）编程到静态MAC表中。此外，应将“改写”位置1，以使开关将这些特定数据包转发到处理器。该状态下禁止端口的地址学习功能。
<b>侦听状态</b>	<b>端口设置</b>	<b>软件操作</b>
仅转发来自或发送到处理器的数据包。禁止学习功能。	“发送使能 = 0， 接收使能 = 0， 学习禁止 = 1。”	处理器应将需要接收的条目（如BPDU数据包）编程到静态MAC表中。应将“改写”位置1，以使开关将这些特定数据包转发到处理器。在该状态下，处理器可能向（多个）端口发送数据包（有关详细信息，请参见“尾部标记模式”一节）。该状态下禁止端口的地址学习功能。
<b>学习状态</b>	<b>端口设置</b>	<b>软件操作</b>
仅转发来自或发送到处理器的数据包。使能学习功能。	“发送使能 = 0， 接收使能 = 0， 学习禁止 = 0。”	处理器应将需要接收的条目（如BPDU数据包）编程到静态MAC表中。应将“改写”位置1，以使开关将这些特定数据包转发到处理器。在该状态下，处理器可能向（多个）端口发送数据包（有关详细信息，请参见“尾部标记模式”一节）。该状态下使能端口的地址学习功能。
<b>转发状态</b>	<b>端口设置</b>	<b>软件操作</b>
正常转发和接收数据包。使能学习功能。	“发送使能 = 1， 接收使能 = 1， 学习禁止 = 0。”	处理器应将需要接收的条目（如BPDU数据包）编程到静态MAC表中。应将“改写”位置1，以使开关将这些特定数据包转发到处理器。在该状态下，处理器可能向（多个）端口发送数据包（有关详细信息，请参见“尾部标记模式”一节）。该状态下使能端口的地址学习功能。

# KSZ8795CLX

## 3.6.3 快速生成树支持

对于RSTP，每个端口指定了3种工作状态，即丢弃、学习和转发。丢弃端口既不参与活动拓扑也不学习MAC地址。学习状态下的端口学习MAC地址，但不转发用户通信。转发状态下的端口完全参与数据转发和MAC学习。RSTP仅使用一种类型的BPDU（称为RSTP BPDU）。RSTP BPDU与STP配置BPDU类似，但以下两方面除外：RSTP的类型字段设置为“版本2”而STP的类型字段设置为“版本0”；RSTP有承载附加信息的标志字段。

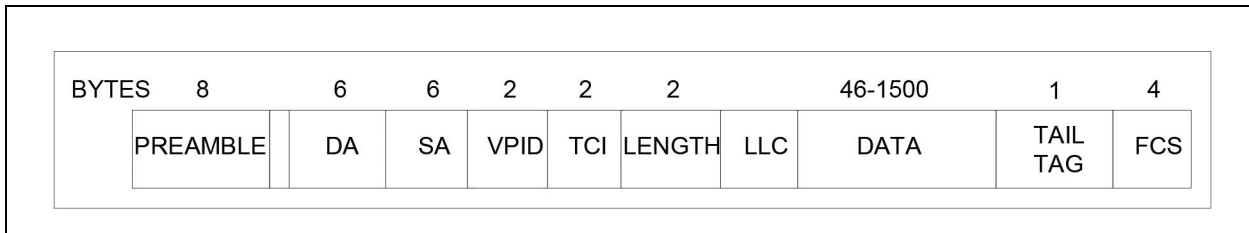
**表3-14： 快速生成树的端口设置和软件操作**

禁止状态	端口设置	软件操作
STP包括三种状态，即禁止、阻塞和学习。	“发送使能 = 0， 接收使能 = 0， 学习禁止 = 1。”	处理器不应向端口发送任何数据包。开关可能仍向处理器发送特定数据包（与静态表中的一些条目匹配且“改写位”置1的数据包），处理器应将这些数据包丢弃。禁止端口的学习功能（学习禁止 = 1）时，将寄存器1 bit[5]和bit[4]置1可快速刷新动态MAC表和静态MAC表中的端口相关条目。注：处理器通过MII模块连接到端口5。该状态下禁止端口的地址学习功能。
学习状态	端口设置	软件操作
仅转发来自或发送到处理器的数据包。使能学习功能。	“发送使能 = 0， 接收使能 = 0， 学习禁止 = 0。”	处理器应将需要接收的条目（如BPDU数据包）编程到静态MAC表中。应将“改写”位置1，以使开关将这些特定数据包转发到处理器。在该状态下，处理器可能向（多个）端口发送数据包（有关详细信息，请参见“尾部标记模式”一节）。该状态下使能端口的地址学习功能。
转发状态	端口设置	软件操作
正常转发和接收数据包。使能学习功能。	“发送使能 = 1， 接收使能 = 1， 学习禁止 = 0。”	处理器应将需要接收的条目（如BPDU数据包）编程到静态MAC表中。应将“改写”位置1，以使开关将这些特定数据包转发到处理器。在该状态下，处理器可能向（多个）端口发送数据包（有关详细信息，请参见“尾部标记模式”一节）。该状态下使能端口的地址学习功能。

## 3.6.4 尾部标记模式

尾部标记只能由端口5接口看到和使用，该接口应通过SW5-GMII、RGMII、MII或RMII接口连接到处理器。一个字节的尾部标记用于指示端口5上的源/目标端口。在尾部标记字节中，只有bit[3:0]用于目标端口。不使用其他位。通过将寄存器12 bit[1]置1使能尾部标记功能。

**图3-10： 尾部标记帧格式**



**表3-15: 尾部标记规则**

传入端口5（主机到KSZ8795CLX）	
bit[3:0]	目标
0,0,0,0	保留
0,0,0,1	端口1（直接转发到端口1）
0,0,1,0	端口2（直接转发到端口2）
0,1,0,0	端口3（直接转发到端口3）
1,0,0,0	端口4（直接转发到端口4）
1,1,1,1	端口1、2、3和4（直接转发到端口1、2、3和4）
bit[7:4]	—
0,0,0,0	目标端口使用队列0
0,0,0,1	目标端口使用队列1
0,0,1,0	目标端口使用队列2
0,0,1,1	目标端口使用队列3
0,1,x,x	以任意方式将数据包发送到bit[3:0]中的指定端口
1,x,x,x	bit[6:0]将被忽略为正常状态（地址查找）
从端口5传出（KSZ8795CLX到主机）	
bit[1:0]	源
0,0	端口1（来自端口1的数据包）
0,1	端口2（来自端口2的数据包）
1,0	端口3（来自端口3的数据包）
1,1	端口4（来自端口4的数据包）

### 3.6.5 IGMP支持

第2层中有两部分与支持Internet组管理协议（Internet Group Management Protocol, IGMP）有关。第一部分是IGMP侦听，第二部分是回送到订阅端口的IGMP数据包。下面分别介绍这两部分。

#### 3.6.5.1 IGMP侦听

KSZ8795CLX捕获IGMP数据包并仅转发给处理器（端口5 SW5-RGMII/MII/RMII）。IGMP数据包被识别为IP数据包（以太网IP数据包或IEEE 802.3 SNAP IP数据包），IP版本 = 0x4，协议版本号 = 0x2。将寄存器5 bit[6]设置为1使能IGMP侦听。

#### 3.6.5.2 IGMP回送到订阅端口

主机响应接收到的IGMP数据包后，主机应了解原始IGMP传入端口并仅将IGMP数据包回送到该端口，以避免向所有端口广播此IGMP数据包，因为这样做会降低性能。

如果使能了尾部标记模式，主机将通过尾部标记位[1:0]了解IGMP数据包接收自哪个端口，并可通过将尾部标记位[3:0]置1将响应IGMP数据包回送到该订阅端口。将寄存器12 bit[1]置1使能尾部标记模式。

### 3.6.6 IPV6 MLD侦听

KSZ8795CLX捕获IPv6多播侦听发现（MLD）数据包并仅转发给处理器（端口5）。MLD侦听由寄存器164 bit[2]（MLD侦听使能）和寄存器164 bit[3]（MLD选项）控制。

通过使能MLD侦听，KSZ8795CLX捕获符合以下所有条件的数据包：

- IPv6多播数据包
- 跳数限制 = 1
- IPv6下一个报头 = 1或58（或 = 0且逐个跳段下一个报头 = 1或58），如果MLD选项位设置为1，则KSZ8795CLX捕获符合以下附加条件的数据包：
  - IPv6下一个报头 = 43、44、50、51或60（或 = 0且逐个跳段下一个报头 = 43、44、50、51或60）

# KSZ8795CLX

对于MLD侦听，还需要使能尾部标记模式，从而使处理器了解MLD数据包接收自哪个端口。这可通过将寄存器12 bit[1]置1实现。

## 3.6.7 端口镜像支持

KSZ8795CLX支持“端口镜像”，如下面所述：

### 3.6.7.1 端口上的“仅接收”镜像

端口上接收的所有数据包将镜像到嗅探器端口。例如，端口1设定为“RX嗅探”，端口5设定为“嗅探器端口”。端口1上接收的数据包在内部查找后发送到端口4。KSZ8795CLX会将该数据包同时转发到端口4和端口5。KSZ8795CLX甚至可以选择将接收的“不良”数据包转发到端口5。

### 3.6.7.2 端口上的“仅发送”镜像

端口上发送的所有数据包将镜像到嗅探器端口。例如，端口1设定为“TX嗅探”，端口5设定为“嗅探器端口”。任意端口上接收的数据包在内部查找后发送到端口1。KSZ8795CLX会将该数据包同时转发到端口1和端口5。

### 3.6.7.3 两个端口上的“接收和发送”镜像

端口A上接收的和端口B上发送的所有数据包将镜像到嗅探器端口。为开启“AND”功能，将寄存器5 bit[0]和bit[1]置1。例如，端口1设定为“RX嗅探”，端口2设定为“TX嗅探”以及端口5设定为“嗅探器端口”。端口1上接收的数据包在内部查找后发送到端口4。KSZ8795CLX仅将该数据包转发到端口4，因为它不满足“AND”条件。端口1上接收的数据包在内部查找后发送到端口2。KSZ8795CLX会将该数据包同时转发到端口2和端口5。

可将多个端口选定为“RX嗅探”或者“TX嗅探”。可将任意端口选定为“嗅探器端口”。所有这些端口功能均可通过端口控制寄存器1选择。

## 3.6.8 VLAN支持

KSZ8795CLX支持128个活动VLAN和IEEE 802.1q中指定的4096个可能的VID。KSZ8795CLX提供一个128条目VLAN表（对应于4096个可能的VID），可转换为FID（7位）以通过地址查找最多128个活动VLAN。如果接收到无标记或带空VID标记的数据包，则通过全局寄存器5控制3 bit[7]使能802.1q时，将使用传入端口的VID进行查找。在VLAN模式下，查找过程从VLAN表查找开始，以确定VID是否有效。如果VID无效，将丢弃该数据包并且不学习其地址。如果VID有效，则通过静态MAC表或动态MAC表检索FID以供进一步查找。FID+DA用于确定目标端口。

表3-16给出了VLAN表完成查找操作后静态MAC表和动态MAC表中不同DA和FID+DA情形下的不同操作。FID+SA用于学习目的。此外，表3-17还给出了当VLAN表对静态MAC表完成查找且未找到有效条目时动态表中的学习情形。

表3-16: VLAN模式下的FID+DA查找

在静态MAC表中找到DA?	使用FID标志?	FID匹配?	在动态MAC表中找到FID+DA?	操作
否	无关	无关	否	广播到VLAN表bit[11:7]中定义的成员端口。
否	无关	无关	是	发送到动态MAC地址表bit[58:56]中定义的目标端口。
是	0	无关	无关	发送到静态MAC地址表bit[52:48]中定义的目标端口。
是	1	否	否	广播到VLAN表bit[11:7]中定义的成员端口。

**表3-16: VLAN 模式下的FID+DA查找 (续)**

在静态MAC表中找到DA?	使用FID标志?	FID匹配?	在动态MAC表中找到FID+DA?	操作
是	1	否	是	发送到动态MAC地址表bit[58:56]中定义的目标端口。
是	1	是	无关	发送到静态MAC地址表bit[52:48]中定义的目标端口。

**表3-17: VLAN 模式下的FID+SA查找**

在动态MAC表中找到FID+SA?	操作
否	FID+SA将被学习到动态表中。
是	将更新时间戳。

KSZ8795CLX还支持高级VLAN功能，例如端口控制2寄存器bit[6:5]中定义的“VLAN传入过滤”和“丢弃非PVID”。这些功能可以基于各个端口进行控制。

### 3.6.9 速率限制支持

KSZ8795CLX提供了基于bps（位每秒）和pps（数据包每秒）的高精度分辨率硬件速率限制功能。

对于bps，100BT或10BT在速率限制小于1 Mbps时的速率步进为64 Kbps，1000BT在速率限制小于1 Mbps时的速率步进为640 Kbps。100BT或10BT在速率限制大于1 Mbps的速率步进为1 Mbps，1000BT在速率限制大于1 Mbps时的速率步进为10 Mbps。

对于pps，100BT或10BT在速率限制小于1.92 Kpps时的速率步进为128 pps（除了第一个步进为64 pps以外），1000BT在速率限制小于1.92 Kpps时的速率步进为1280 pps（除了第一个步进为640 pps以外）。100BT或10BT在速率限制大于1.92 Kpps时的速率步进为1.92 Kpps，1000BT在速率限制大于1.92 Kpps时的速率步进为19.2 Kpps（见表3-18）。

pps限制受每个pps设置的bps速率约束。表3-18的第二列中给出了映射。

**表3-18: 不同速率限制的10/100/1000 Mbps速率选择**

项目	pps的bps约束 (仅传出)	10 Mbps		100 Mbps		1000 Mbps	
		19.2 Kpps	10 Mbps	19.2 Kpps	100 Mbps	1.92 Mpps	1000 Mbps
7d'0	7d'0	19.2 Kpps	10 Mbps	19.2 Kpps	100 Mbps	1.92 Mpps	1000 Mbps
7d'1 - 7d'10	7d'3, 6, (8x)10	1.92 Kpps x 编码	1 Mbps x 编码	1.92 Kpps x 编码	1 Mbps x 编码	19.2 Kpps x 编码	10 Mbps x 编码
7d'11 - 7d'100	7d'11 - 7d'100	—	10 Mbps	1.92 Kpps x 编码	1 Mbps x 编码	19.2 Kpps x 编码	10 Mbps x 编码
7d'101	7d'102	64 pps	64 Kbps	64 pps	64 Kbps	640 pps	640 Kbps
7d'102	7d'104	128 pps	128 Kbps	128 pps	128 Kbps	1280 pps	1280 Kbps
7d'103	7d'108	256 pps	192 Kbps	256 pps	192 Kbps	2560 pps	1920 Kbps
7d'104	7d'112	384 pps	256 Kbps	384 pps	256 Kbps	3840 pps	2560 Kbps
7d'105	7d'001	512 pps	320 Kbps	512 pps	320 Kbps	5120 pps	3200 Kbps
7d'106	7d'001	640 pps	384 Kbps	640 pps	384 Kbps	6400 pps	3840 Kbps
7d'107	7d'001	768 pps	448 Kbps	768 pps	448 Kbps	7680 pps	4480 Kbps
7d'108	7d'002	896 pps	512 Kbps	896 pps	512 Kbps	8960 pps	5120 Kbps
7d'109	7d'002	1024 pps	576 Kbps	1024 pps	576 Kbps	10240 pps	5760 Kbps
7d'110	7d'002	1152 pps	640 Kbps	1152 pps	640 Kbps	11520 pps	6400 Kbps
7d'111	7d'002	1280 pps	704 Kbps	1280 pps	704 Kbps	12800 pps	7040 Kbps
7d'112	7d'002	1408 pps	768 Kbps	1408 pps	768 Kbps	14080 pps	7680 Kbps
7d'113	7d'003	1536 pps	832 Kbps	1536 pps	832 Kbps	15360 pps	8320 Kbps



# KSZ8795CLX

表3-18: 不同速率限制的10/100/1000 Mbps速率选择 (续)

项目	pps的bps约束 (仅传出)	10 Mbps		100 Mbps		1000 Mbps	
		pps	Kbps	pps	Kbps	pps	Kbps
7d'114	7d'003	1664 pps	896 Kbps	1664 pps	896 Kbps	16640 pps	8960 Kbps
7d'115	7d'003	1792 pps	969 Kbps	1792 pps	969 Kbps	17920 pps	9690 Kbps

各个端口“接收端”和“发送端”的速率限制是独立的。对于10BASE-T, 10 Mbps以上的速率设置表示未限制速率。

在接收端, 可通过设置传入速率控制寄存器在每个端口上对每个优先级的数据接收速率进行限制。在发送端, 可通过设置传出速率控制寄存器在每个端口上对每个队列的数据发送速率进行限制。对于bps模式, 除数据字段(从数据包DA到FCS)外, 每个帧的大小还可以选择包含最小帧间隔(Interframe Gap, IFG)或前导字节。

### 3.6.9.1 传入速率限制

对于传入速率限制, KSZ8795CLX可以有选择地从所有类型的帧中进行挑选; 通过端口速率限制控制寄存器的bit[3:2]选择多播、广播或泛洪单播帧。KSZ8795CLX对那些选定类型帧的数据速率进行计数。如果数据速率超过规定的速率限制, 则传入端口将丢弃数据包; 或者, 如果通过端口速率限制控制寄存器bit[4]使能了传入速率限制流控制, 则流控制生效且不丢弃数据包。传入速率限制支持基于端口和基于802.1p和DiffServ的优先级。基于端口的优先级固定为优先级0-3, 可通过端口控制0寄存器的bit[4:3]进行选择。基于802.1p和DiffServ的优先级可通过寄存器128和129的默认设置映射到优先级0-3。在传入速率限制中, 如果采用2队列或4队列模式, 则将寄存器135全局控制19 bit[3]置1以使能基于队列的速率限制。应通过端口控制9和控制0寄存器将所有相关传入端口和传出端口划分为2队列或4队列模式。4队列模式将通过端口寄存器传入限制控制1-4的bit[6:0]为优先级0-3使用Q0-Q3。2队列模式将通过端口传入限制控制1-2寄存器的bit[6:0]为优先级0-1使用Q0-Q1。802.1p和DiffServ数据包中的优先级可通过配置寄存器128和129设定为优先级0-3以进行重映射。

### 3.6.9.2 传出速率限制

对于传出速率限制, 将漏桶算法应用于每个输出优先队列以进行输出流量整形。延长每帧的帧间隔, 以产生光滑且无冲突的传出流量。每个输出优先级队列的吞吐量由数据速率选择表(遵循传出速率限制控制寄存器)指定的传出速率限制。

如果任何传出队列接收的流量超过指定的传出速率吞吐量, 数据包可能积累到输出队列和数据包存储器中。如果队列或端口的存储器用尽, 则会触发数据包丢弃或流控制。如果发生拥塞, 实际传出速率可能由传出端的流控制/丢弃主导, 并可能因此略小于指定传出速率。传出速率限制支持基于端口以及基于802.1p和DiffServ的优先级, 基于端口的优先级固定为优先级0-3, 可通过端口控制0寄存器的bit[4:3]进行选择。基于802.1p和DiffServ的优先级可通过寄存器128和129的默认设置映射到优先级0-3。在传出速率限制中, 如果采用2队列或4队列模式, 则将寄存器135全局控制19 bit[3]置1以使能基于队列的速率限制。应通过端口控制9和控制0寄存器将所有相关传入端口和传出端口划分为2队列或4队列模式。4队列模式将通过端口传出限制控制1-4寄存器为优先级0-3使用Q0-Q3。2队列模式将通过端口传出速率限制控制1-2寄存器为优先级0-1使用Q0-Q1。802.1p和DiffServ数据包中的优先级可通过配置寄存器128和129设定为优先级0-3以进行重映射。

当传出速率被限制时, 每个端口只使用1个队列进行传出端口速率限制。优先级数据包将基于数据速率选择表(见表3-18)。如果传出速率限制在每个端口上使用多个队列进行传出端口速率限制, 则最高优先级数据包将基于数据速率选择表中的速率限制精确数值。其他较低优先级的数据包速率将基于8:4:2:1(默认)的优先比加以限制, 此优先比基于最高优先级的速率。发送队列的优先比是可编程的。

为减少拥塞, 最好确保传出带宽超过传入带宽。



### 3.6.9.3 发送队列优先比编程

传出端口队列0-3的默认发送优先比为8:4:2:1。可通过端口控制10、11、12和13寄存器设定优先比。当发送速率超过发送队列的优先比限制时，端口控制10、11、12和13寄存器的发送队列0-3优先比会限制发送速率。最高优先级队列不受限制。将基于发送队列优先比限制其他较低优先级队列。

### 3.6.10 VLAN和地址过滤

为防止特定种类的数据包可能降低互联网协议语音（Voice Over Internet Protocol, VoIP）等应用中开关的质量，开关提供了一种机制，可使用以下MAC地址和VLAN ID对数据包进行过滤和映射。

- 自寻址数据包
- 未知单播数据包
- 未知多播数据包
- 未知VID数据包
- 未知IP多播数据包

通过全局控制18寄存器bit[6]，可使能自寻址过滤将源自开关本身的数据包滤除。自寻址过滤将过滤传出端口上的数据包；自身MAC地址在寄存器104至109 MAC地址寄存器0至5中分配。

未知单播数据包过滤可通过全局控制寄存器15 bit[5]使能，该寄存器的bit[4:0]指定转发端口映射。

未知多播数据包过滤可通过全局控制寄存器16 bit[5]使能，该寄存器的bit[4:0]指定转发端口映射。

未知VID数据包过滤可通过全局控制寄存器17 bit[5]使能，该寄存器的bit[4:0]指定转发端口映射。

未知IP多播数据包过滤可通过全局控制寄存器18 bit[5]控制，该寄存器的bit[4:0]指定转发端口映射。

上述过滤均基于全局。

### 3.6.11 基于802.1X端口的安全

IEEE 802.1x是一种基于端口的验证协议。EAPOL是一种通常由验证过程用作非受控端口的协议。通过接收并提取特殊的EAPOL帧，微处理器（CPU）可以控制传入和传出端口是否应转发数据包。如果某个用户端口需要来自其他端口的服务（验证器），则必须获得验证器的批准。通过检查帧的目标地址，KSZ8795CLX可以检测到EAPOL帧。目标地址应当是一个如IEEE 802.1x中定义的多播地址（01-80-C2-00-00-03）或可编程的保留多播地址域中使用的地址，偏移量为-00-03。检测到EAPOL帧后，会将其转发到CPU，这样CPU便可将这些帧发送到验证器服务器。最终，CPU根据请求者MAC源地址确定其是否合格，并接收或丢弃这些帧。

将KSZ8795CLX配置为验证器时，必须随后将开关的端口配置为验证用途。在初始化为验证器的端口验证中，客户端上电或插入端口，验证器端口向客户端发送一个扩展认证协议（Extensible Authentication Protocol, EAP）PDU，请求客户端进行识别。在过程的这一阶段，从物理角度看，开关上的端口已连接；不过，802.1X过程未授权该端口，因此没有任何帧通过客户端上的端口传递到开关结构中。如果与开关相连的PC不理解其正从开关接收的EDP PDU，则PC将无法发送ID，并且该端口将保持未授权状态。在这种状态下，端口将不允许任何用户流量通过，如同被禁止一样。如果客户端PC正在运行802.1X EAP，则会使用其已配置的ID响应请求。此ID可以是用户名/密码组合或者证书。

开关验证后，验证器从PC（客户端）接收ID。KSZ8795CLX随后将该ID信息传递到验证服务器（RADIUS服务器），该服务器可以验证标识信息。RADIUS服务器使用成功或失败消息响应开关。如果响应成功，则端口将获得授权，允许用户流量通过该端口，如同任何连接到访问设备的开关端口一样。如果响应失败，则端口将保持未授权状态，并因此保持未使用状态。如果服务器没有任何响应，则端口也将保持未授权状态，不会传递任何流量。

# KSZ8795CLX

---

## 3.6.11.1 验证寄存器和编程模型

端口验证控制寄存器定义基于端口的验证控制。可在这些寄存器中设定各个端口的验证。KSZ8795CLX提供了三种用于实现IEEE 802.1x功能的模式。通过将端口验证寄存器中的相应位置1，可以选择各个模式。

当模式控制位AUTHENTICATION\_MODE = 00（通过模式）时，将使能强制验证，某个端口将始终被授权，并且无需客户端或验证服务器发送任何消息。这种情况通常发生在KSZ8795CLX连接到另一个开关、路由器或服务器时，以及连接到不支持802.1X的客户端时。使能ACL时，不符合ACL规则的所有数据包都将通过，否则将应用ACL操作。

阻止模式（AUTHENTICATION\_MODE = 01）是标准的基于端口的验证模式。在该模式下，端口向客户端发送EAP数据包，并在接收到验证服务器的肯定响应后变为授权状态。对所有传入数据包进行验证前，通信被阻止；验证通过后，软件将切换到通过模式，以允许所有传入数据包通过。在该模式下，不检查传入数据包的源地址。需要配置包括EAP地址在内的整个保留多播地址的转发映射，以允许在验证前后在查找表中对其进行转发。使能ACL时，除ACL命中以外的所有数据包被阻止。

第三种模式是陷阱模式（AUTHENTICATION\_MODE = 11'b）。在该模式下，所有数据包将发送至CPU端口。如果使能ACL，这些不符合ACL规则的数据包会被转发到CPU，而不是被丢弃。基于各个端口的所有这些功能均可通过端口控制5寄存器来选择，其中bit[2]用于使能ACL，bit[1:0]用于选择模式。

## 3.6.12 ACL 过滤

通过创建访问控制列表（ACL），可以执行与协议无关的第2层MAC、第3层IP或第4层TCP/UDP ACL过滤，ACL过滤根据ACL规则表过滤传入的以太网数据包。该功能允许开关根据以太网报头中的源MAC地址、IP报头中的IP地址以及TCP报头中的端口号和协议过滤客户流量。该功能可通过MAC表和ACL规则表执行。除了使用静态表中的条目处理多播过滤外，还可针对所有路由网络协议配置ACL，以在这些协议的数据包通过开关时执行过滤。ACL可阻止特定通信进入或退出网络。

### 3.6.12.1 访问控制列表

KSZ8795CLX提供了一个基于规则的ACL规则表。ACL规则表是一个访问控制条目的有序列表。每个条目指定特定的规则（一组匹配条件和操作规则），以允许或拒绝对开关结构的数据包访问。“允许”或“拒绝”的含义依使用ACL时的环境而定。当某个接口接收到数据包时，开关将数据包中的字段与任意已应用的ACL比较，根据规则表中指定的条件验证数据包是否具有转发所需的权限。

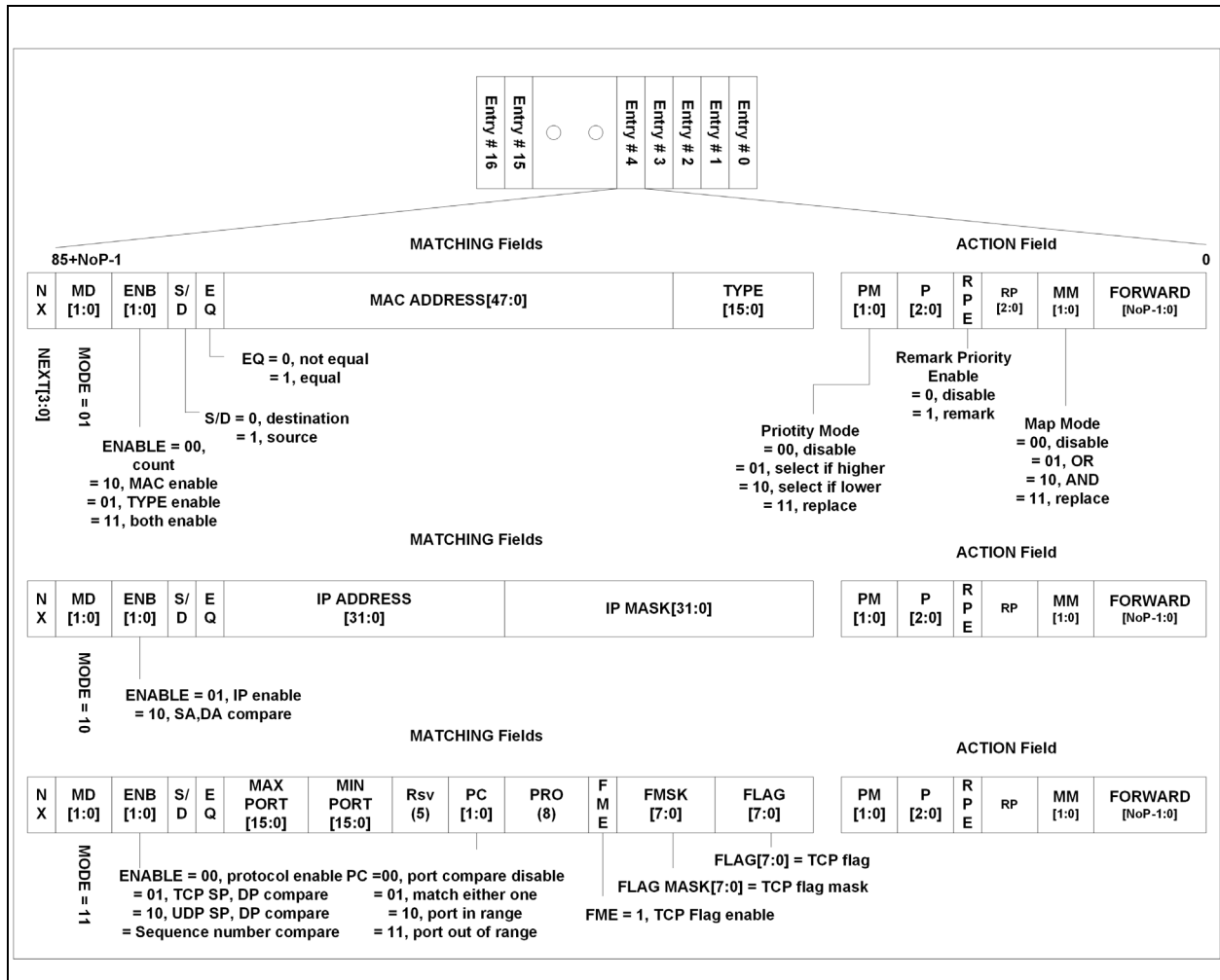
过滤器会根据规则表中的ACL条目对数据包进行逐一检验。通常由第一次匹配决定路由器是接收还是拒绝数据包。不过，允许对规则进行级联，以对传入数据包构成更可靠和/或更严格的要求。ACL允许开关根据数据包第2层报头中的源MAC地址、目标MAC地址和以太网类型，第3层报头中的源IP地址和目标IP地址以及第4层报头中的端口号和协议过滤传入流量。

每个列表由三部分组成：

- 匹配字段
- 操作字段
- 处理字段

匹配字段指定每个数据包匹配的规则，操作字段指定针对规则测试成功时所采取的操作。图3-11给出了ACL的格式以及各个字段的说明。

图3-11: ACL 格式



### 匹配字段

- MD[1:0]: MODE  
ACL 定义了 3 种工作模式。模式 0 禁止当前规则列表，模式 1 是第 2 层 MAC 报头过滤的资格规则，模式 2 用于第 3 层 IP 地址过滤，模式 3 执行第 4 层 TCP 端口号/协议过滤。选择模式 0 时，将不采取任何操作。
- ENB[1:0]: ENABLE  
使能当前列表中的不同规则。
  - MD = 01 时  
ENB = 00 时，操作字段中 PM、P、RPE、RP 和 MM 的 11 位聚合位域为与匹配字段中的 MAC 地址和类型相匹配的数据包指定一个计数值。  
计数单位在 FORWARD 位域的 MSB 中定义。此 MSB 等于 0 时，计数单位为  $\mu$ s；等于 1 时，计数单位为 ms。  
FORWARD 位的第 2 个 MSB 确定计数器终止时用于生成中断的算法。第 2 个 MSB 等于 0 时，11 位计数器将装入 ACL 列表中的计数值，并开始按每单位时间递减一次进行计数。计数结束（即，在该值指定的时间内尚未接收下一个合格的数据包）时会产生中断。  
第 2 个 MSB 等于 1 时，计数器会在每次接收到匹配数据包时递增，当终止计数达到 ACL 列表中的计数值时，将产生中断，之后计数器复位。  
ENB = 01 时，MAC 地址位域参与检验；ENB = 10 时，MAC 类型位域用于检验；ENB = 11 时，列表中的 MAC 地址和 MAC 类型位域均进行检验。
  - MD = 10 时

# KSZ8795CLX

---

如果ENB = 01, 则使能IP地址和掩码或IP协议以进行相应检验。如果ENB = 10, 则比较源地址和目标地址。根据EQ位设置做出丢弃/转发决策。

- MD = 11时  
如果ENB = 00, 则使能协议比较。  
如果ENB = 01, 则选择TCP地址比较。  
如果ENB = 10, 则选择UDP地址比较。  
如果ENB = 11, 则比较TCP的序列号。
- S/D: 源地址或目标地址选择
  - 等于0时, 目标地址/端口用于比较; 等于1时, 选择源地址。
- E/Q: 比较算法
  - 等于0时, 表示不相等时匹配。等于1时, 表示相等时匹配。
- MAC地址[47:0]
  - MAC源地址或目标地址
- TYPE[15:0]
  - MAC以太网类型。
- IP地址[31:0]
  - IP源地址或目标地址。
- IP掩码[31:0]
  - 用于组地址过滤的IP地址掩码。
- 最大端口[15:0]、最小端口[15:0]/序列号[31:0]
  - TCP端口号或序列号匹配范围。
- PC[1:0]: 端口比较
  - 等于00时, 禁止比较; 等于01时, 匹配最大或最小端口之一; 等于10时, 端口号处于最大到最小范围内时匹配; 等于11时, 端口号超出范围时匹配。
- PRO[7:0]
  - 待匹配的IP协议。
- FME
  - 标志匹配使能: 等于1时, 使能TCP标志匹配。
- FLAG[5:0]
  - 待匹配的TCP标志。

## 操作字段

- PM[1:0]: 优先级模式
  - 等于00时, 未选择任何优先级, 优先级由QoS/分类确定。等于01时, 如果P位域的值大于QoS结果, 则使用P位域中的优先级。等于10时, 如果P位域的值小于QoS结果, 则使用P位域中的优先级。等于11时, P位域将替换QoS确定的优先级。
- P[2:0]
  - 优先级。
- RPE: 备注优先级使能
  - 等于0时, 无需备注。等于1时, 带标记数据包中的VLAN优先级位将替换为列表中的RP位域。
- RP[2:0]
  - 备注优先级。
- MM[1:0]: 映射模式
  - 等于00时, 无需转发重映射。等于01时, FORWARD中的转发映射将与来自查找表的转发映射做或运算。等于10时, FORWARD中的转发映射将与来自查找表的转发映射做与运算。等于11时, FORWARD中的转发映射将替换来自查找表的转发映射。
- FORWARD bit[4:0]: 转发端口——每一位表示一个端口的转发决策。

## 处理字段

- FRN bit[3:0]: 第一条规则编号
  - 从16个条目中分配在规则集中使用哪个条目及其操作字段。
- RULESET bit[15:0]: 规则集

- 待认证的规则集，可通过2个规则集寄存器将16条规则分配给每个端口的规则集。规则表允许对规则进行级联。RTB中有16个条目。每个条目本身可作为一个规则，或者可与其他条目级联组成一个规则集。对传入数据包针对规则集中所包含规则的所有测试结果做与运算，即得到传入数据包针对此规则集的测试结果。规则集的操作将成为FRN字段中指定的第一条规则的操作。规则的优先级越高，其索引号越小。也就是说，规则0具有最高优先级，规则15具有最低优先级。当模式位设置为2'b00时，将禁止ACL规则表条目。  
规则集（RULESET）用于选择不同规则针对传入数据包的匹配结果。将对这些选定的匹配结果做与运算，以确定帧是否匹配。具有相同操作的不同规则集的条件将进行或运算以与帧字段进行比较，而CPU会将该相同操作编程到要进行或运算的这些不同规则集中。对于匹配规则集，将根据每个规则集的第一条规则编号（First Rule Number, FRN）对具有不同操作的不同规则集进行仲裁或选择。建立规则表时，将优先级高或索引号小的规则置于表的顶部。无论匹配规则集是否具有相同的操作，硬件始终会比较不同规则集的第一条规则编号，以确定最终的规则集和操作。

### 3.6.12.2 ACL的DoS攻击防护

根据分别设定为丢弃或不丢弃每种DoS数据包的规则设置，ACL可为以下拒绝服务（Denial of Service, DoS）攻击类型提供特定的检测/保护。

#### 例1

当MD = 10且ENABLE = 10时，将EQ位置1可以确定丢弃或转发具有相同源和目标IP地址的IPv4/IPv6数据包。

#### 例2

当MD = 11且ENABLE = 01/10时，将EQ位置1可以确定丢弃或转发具有相同源和目标TCP/UDP端口的IPv4/IPv6数据包。

#### 例3

当MD = 11、ENABLE = 11、序列号 = 0、FME = 1、FMSK = 00101001且FLAG = xx1x1xx1时，将EQ位置1可以确定丢弃或转发TCP序列号等于0、标志位URG = 1、PSH = 1且FIN = 1的所有数据包。

#### 例4

当MD = 11、ENABLE = 01、最大端口 = 1024、最小端口 = 0、FME = 1、FMSK = 00010010且FLAG = xxx0xx1x时，将EQ位置1可以确定丢弃或转发TCP端口号 ≤ 1024、标志位URB = 0且SYN = 1的所有数据包。

与ACL相关的寄存器有：

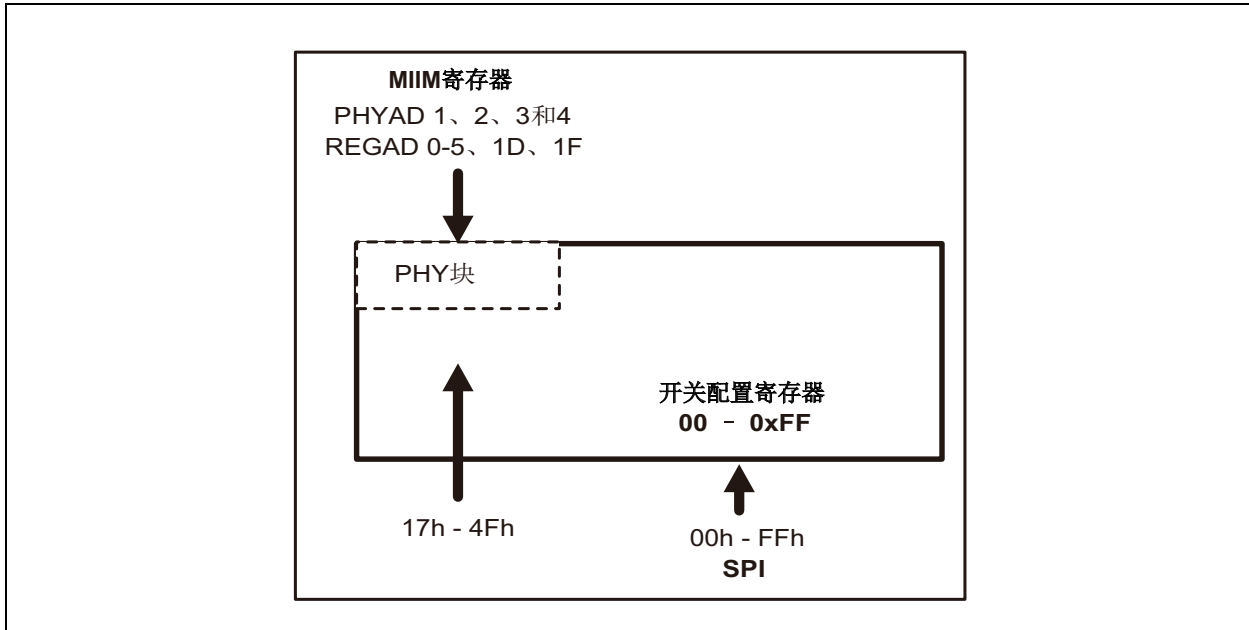
- 寄存器110（0x6E）、寄存器111（0x6F）和ACL规则表。

# KSZ8795CLX

## 4.0 器件寄存器

KSZ8795CLX 器件具有一组丰富的寄存器，可用于管理器件的功能。这些寄存器通过MIIM或SPI接口进行访问。图4-1给出了可用于访问的各个接口及其寻址范围的全局图。

图4-1: 接口和寄存器映射



线性地址空间0x00-0xFF内的寄存器均可由连接至SPI总线的CPU通过SPI接口进行访问。表4-1汇总了该线性地址空间内各功能的映射。

表4-1: 地址空间内的功能区域映射

寄存器单元	器件区域	说明
0x00 至 0xFF	开关控制和配置	用于控制开关、MAC和PHY整体功能的寄存器
0x6E 至 0x6F	间接控制寄存器	用于对器件内不同区域进行间接寻址和访问的寄存器。 - 管理信息库 (Management Information Base, MIB) 计数器 - 静态MAC地址表 - 动态MAC地址表 - VLAN表 - PME间接寄存器 - ACL间接寄存器 - EEE间接寄存器
0x70 至 0x78	间接访问寄存器	用于对器件内四个不同区域进行间接寻址和访问的寄存器。 - 管理信息库 (MIB) 计数器 - 静态MAC地址表 - 动态MAC地址表 - VLAN表
0xA0	间接字节访问寄存器	此间接字节寄存器用于访问： - PME间接寄存器 - ACL间接寄存器 - EEE间接寄存器

**表4-1: 地址空间内的功能区域映射 (续)**

寄存器单元	器件区域	说明
0x17至0x4F	PHY1至PHY4 MIIM 寄存器 映射至那些端口寄存器的地址范围	IEEE 802.3 规范中指定的PHY 寄存器。

## 4.1 寄存器映射

**表4-2: 直接寄存器**

地址	内容
0x00-0x01	系列ID、芯片ID、版本ID和启动开关寄存器
0x02-0x0D	全局控制寄存器0至11
0x0E-0x0F	全局掉电管理控制寄存器
0x10-0x14	端口1控制寄存器0至4
0x15	端口1认证控制寄存器
0x16-0x18	端口1保留(出厂测试寄存器)
0x19-0x1F	端口1控制/状态寄存器
0x20-0x24	端口2控制寄存器0至4
0x25	端口2认证控制寄存器
0x26-0x28	端口2保留(出厂测试寄存器)
0x29-0x2F	端口2控制/状态寄存器
0x30-0x34	端口3控制寄存器0至4
0x35	端口3认证控制寄存器
0x36-0x38	端口3已注册(出厂测试寄存器)
0x39-0x3F	端口3控制/状态寄存器
0x40-0x44	端口4控制寄存器0至4
0x45	端口4认证控制寄存器
0x46-0x48	端口4保留(出厂测试寄存器)
0x49-0x4F	端口4控制/状态寄存器
0x50-0x54	端口5控制寄存器0至4
0x56-0x58	端口5保留(出厂测试寄存器)
0x59-0x5F	端口5控制/状态寄存器
0x60-0x67	保留(出厂测试寄存器)
0x68-0x6D	MAC地址寄存器
0x6E-0x6F	间接访问控制寄存器
0x70-0x78	间接数据寄存器
0x79-0x7B	保留(出厂测试寄存器)
0x7C-0x7D	全局中断和屏蔽寄存器
0x7E-0x7F	ACL中断状态和控制寄存器
0x80-0x87	全局控制寄存器12至19
0x88	开关自检控制寄存器
0x89-0x8F	QM全局控制寄存器
0x90-0x9F	全局TOS优先级控制寄存器0至15
0xA0	全局间接字节寄存器
0xA0-0xAF	保留(出厂测试寄存器)
0xB0-0xBE	端口1控制寄存器

# KSZ8795CLX

表4-2: 直接寄存器 (续)

地址	内容
0xBF	保留 (出厂测试寄存器): 传输队列重映射基址寄存器
0xC0-0xCE	端口2控制寄存器
0xCF	保留 (出厂测试寄存器)
0xD0-0xDE	端口3控制寄存器
0xDF	保留 (出厂测试寄存器)
0xE0-0xEE	端口4控制寄存器
0xEF	保留 (出厂测试寄存器)
0xF0-0xFE	端口5控制寄存器
0xFF	保留 (出厂测试寄存器)

表4-3: 全局寄存器

地址	名称	说明	模式	默认值
<b>寄存器0 (0x00): 芯片ID0</b>				
7-0	系列ID	芯片系列。	RO	0x87
<b>寄存器1 (0x01): 芯片ID1/启动开关</b>				
7-4	芯片ID	0x9 = 8795	RO	0x9
3-1	版本ID	—	RO	0x0
0	启动开关	1 = 启动芯片的开关功能。 0 = 停止芯片的开关功能。	R/W	1
<b>寄存器2 (0x02): 全局控制0</b>				
7	新后退使能	专为UNH设计的新后退算法 1 = 使能 0 = 禁止	R/W	0
6	全局软复位使能	全局软件复位 1 = 使能以复位所有FSM和数据路径 (非配置)。 0 = 禁止复位。  <b>注:</b> 此复位将使通信状态下的数据包接收操作停止。所有寄存器均会保留其配置值。	R/W	0
5	刷新动态MAC表	为RSTP刷新整个动态MAC表。该位自清零 (Self-Clear, SC)。 1 = 触发刷新动态MAC表操作。 0 = 正常工作。  <b>注:</b> 与学习功能一直处于关闭状态 (学习禁止) 的端口相关的所有条目均将被刷新。如果要刷新整个表, 则必须关闭所有端口的学习功能。	R/W (SC)	0



**表4-3: 全局寄存器 (续)**

地址	名称	说明	模式	默认值
4	刷新静态MAC表	为RSTP刷新静态MAC表中的匹配条目 1 = 触发刷新静态MAC表操作。 0 = 正常工作。  <b>注:</b> 匹配条目的定义为转发端口位域中包含单个端口和单播MAC地址的条目。反过来, 该端口的学习功能一直处于关闭状态 (学习禁止)。对于每个端口, 都可以将多个条目判定为匹配条目。	R/W (SC)	0
3	保留	N/A, 不要更改	RO	1
2	保留	N/A, 不要更改	RO	1
1	UNH模式	1 = 开关将丢弃T/L字段中为0x8808或DA = 01-80-C2-00-00-01的数据包。 0 = 开关将丢弃被判定为“流控制”数据包的数据包。	R/W	0
0	链路更改老化	1 = 链路从“有链路”更改为“无链路”时, 会导致快速老化 (<800 μs), 进而加速地址表老化。经过老化周期后, 老化逻辑会恢复正常状态 (300±75秒)。  <b>注:</b> 如果任一端口被拔出, 则将自动删除所有地址。	R/W	0
<b>寄存器3 (0x03): 全局控制1</b>				
7	保留	N/A, 不要更改。	RO	0
6	2 KB数据包支持	1 = 使能2 KB数据包支持。 0 = 禁止2 KB数据包支持。	R/W	0
5	IEEE 802.3x发送流控制禁止	0 = 基于AN结果使能发送流控制。 1 = 无论AN结果如何, 都不会使能发送流控制。	R/W	0
4	IEEE 802.3x接收流控制禁止	0 = 基于AN结果使能接收流控制。 1 = 无论AN结果如何, 都不会使能接收流控制。  <b>注:</b> bit[5]和bit[4]默认值由同一个引脚进行控制, 但可单独进行编程。	R/W	0
3	帧长字段检查	1 = 检查IEEE数据包中的帧长字段。如果实际长度不匹配, 则会丢弃该数据包 (针对L/T < 1500的情况)。	R/W	0
2	老化使能	1 = 使能芯片中的老化功能。 0 = 禁止老化功能。	R/W	1
1	快速老化使能	1 = 开启快速老化 (800 μs)。	R/W	0
0	高级后退使能	1 = 在半双工模式下使能更高级别的后退算法以提升性能。该功能不包含在IEEE标准中。	R/W	0

# KSZ8795CLX

表4-3: 全局寄存器 (续)

地址	名称	说明	模式	默认值
<b>寄存器4 (0x04) : 全局控制2</b>				
7	单播端口VLAN不匹配丢弃	该功能用于端口VLAN (如端口控制1寄存器中所述)。 1 = 所有数据包均不能跨越VLAN边界。 0 = 单播数据包 (不包括未知/多播/广播) 可跨越VLAN边界。  注: 使能镜像时, 被镜像到其他端口的单目标数据包将被丢弃。	R/W	1
6	多播风暴保护禁止	1 = “广播风暴保护” 不包含多播数据包。只会调节DA = FFFFFFFF数据包。 0 = “广播风暴保护” 包含DA = FFFFFFFF和DA[40] = 1数据包。	R/W	1
5	背压模式	1 = 选择基于载波侦听的背压。 0 = 选择基于冲突的背压。	R/W	1
4	流控制和背压公平模式	1 = 选择公平模式。在该模式下, 如果流控制端口和非流控制端口与同一目标端口进行通信, 则可能会丢弃非流控制端口的数据包。这有助于防止流控制端口长时间受到流控制。 0 = 在该模式下, 如果流控制端口和非流控制端口与同一目标端口通信, 则流控制端口将受到流控制。这对于流控制端口可能不“公平”。	R/W	1
3	无冲突过量丢弃	1 = 发生16次或更多次冲突时, 开关不会丢弃数据包。 0 = 发生16次或更多次冲突时, 开关会丢弃数据包。	R/W	0
2	保留	N/A, 不要更改。	RO	0
1	最大合法数据包大小检查禁止	1 = 使能接收最大大小达1536个字节的数据包。 0 = 对于带标记的数据包, 最大为1522个字节 (不包括含有从CPU至端口1-4的STPID的数据包), 对于不带标记的数据包, 最大为1518个字节。大小超过指定值的数据包均将被丢弃。	R/W	0
0	保留	N/A	RO	0
<b>寄存器5 (0x05) : 全局控制3</b>				
7	802.1q VLAN使能	1 = 使能802.1q VLAN模式。操作前, 需要先对VLAN表进行设置。 0 = 禁止802.1q VLAN。	R/W	0
6	开关端口5 SW5-GMII/RGMII/MII/RMII的IGMP侦听使能接口	1 = 使能IGMP侦听。所有IGMP数据包均将通过开关端口5 GMII/RGMII/MII/RMII接口转发至处理器。 0 = 禁止IGMP侦听。	R/W	0
5-1	保留	N/A, 不要更改。	RO	0000

**表4-3: 全局寄存器 (续)**

地址	名称	说明	模式	默认值
0	嗅探模式选择	1 = 使能 Rx AND Tx 嗅探 (要求源端口和目标端口均匹配)。 0 = 使能 Rx OR Tx 嗅探 (要求源端口或目标端口至少有一个匹配)。  注: 默认值用于实现仅接收嗅探。	R/W	0
<b>寄存器6 (0x06): 全局控制4</b>				
7	开关 SW5-MII/RMII 背压使能	1 = 使能开关 MII/RMII 接口的半双工背压。 0 = 禁止开关 MII 接口的背压。	R/W	0
6	开关 SW5-MII/RMII 半双工模式	1 = 使能 MII/RMII 接口半双工模式。 0 = 使能 MII/RMII 接口全双工模式。	R/W	0
5	开关 SW5-MII/RMII 流控制使能	1 = 使能开关 MII/RMII 接口的全双工流控制。 0 = 禁止开关 MII/RMII 接口的全双工流控制。	R/W	0
4	开关 SW5-MII/RMII 速度	1 = 开关 SW5-MII/RMII 处于 10 Mbps 模式。 0 = 开关 SW5-MII/RMII 处于 100 Mbps 模式。	R/W	0
3	空 VID 替换	1 = 使用端口 VID (12 位) 替换空 VID。 0 = 不替换空 VID。	R/W	0
2-0	广播风暴保护率 bit[10:8]	该寄存器与下一个寄存器共同决定预设时间内输入端口上允许的“64 字节块”数据包数据的数量。对于 100BT, 预设时间为 50 ms; 对于 10BT, 预设时间为 500 ms。默认值为 1%。	R/W	000
<b>寄存器7 (0x07): 全局控制5</b>				
7-0	广播风暴保护率 bit[7:0]	该寄存器与上一个寄存器共同决定预设时间内输入端口上允许的“64 字节块”数据包数据的数量。对于 100BT, 预设时间为 50 ms; 对于 10BT, 预设时间为 500 ms。默认值为 1%。  注: $148,800 \text{ 帧/s} \times 50 \text{ ms/间隔} \times 1\% = 74 \text{ 帧/间隔}$ (近似值) = 0x4A。	R/W	0x4A
<b>寄存器8 (0x08): 全局控制6 MIB 控制</b>				
7	刷新计数器	1 = 已使能端口的所有 MIB 计数器均将复位为 0。操作完成后, 该位会自清零。 0 = 不复位 MIB 计数器。	R/W (SC)	0
6	冻结计数器	1 = 使能的端口会停止计数。 0 = 使能的端口不会停止计数。	R/W	0
5	保留	N/A, 不要更改。	RO	0
4-0	控制使能	1 = 针对各个端口使能刷新和冻结功能。 bit[4] 用于端口 5 刷新 + 冻结。 bit[3] 用于端口 4 刷新 + 冻结。 bit[2] 用于端口 3 刷新 + 冻结。 bit[1] 用于端口 2 刷新 + 冻结。 bit[0] 用于端口 1 刷新 + 冻结。 0 = 禁止刷新和冻结。	R/W	0

# KSZ8795CLX

表4-3: 全局寄存器 (续)

地址	名称	说明	模式	默认值															
<b>寄存器9 (0x09) : 全局控制7</b>																			
7-0	出厂测试	N/A, 不要更改	RO	0x40															
<b>寄存器10 (0x0A) : 全局控制8</b>																			
7-0	出厂测试	N/A, 不要更改	RO	0x00															
<b>寄存器11 (0x0B) : 全局控制9</b>																			
7	保留	N/A, 不要更改	RO	0															
6	端口5 SW5-RMII参考时钟边沿选择	选择SW5-RMII参考时钟的数据采样边沿:  1 = 在REFCLK负边沿进行数据采样。 0 = 在REFCLK正边沿进行数据采样 (默认值)。	R/W	0															
5-4	LED模式	可编程LED输出, 使用控制寄存器的其中2位指示端口的活动/状态。输出为低电平时, LED点亮 (有效); 输出为高电平时, LED熄灭 (无效)。  <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Control Bits [5:4]</th> <th>00</th> <th>01</th> <th>10</th> <th>11</th> </tr> </thead> <tbody> <tr> <td>LEDx_1</td> <td>Speed</td> <td>ACT</td> <td>Duplex</td> <td>Duplex</td> </tr> <tr> <td>LEDx_0</td> <td>Link/ACT</td> <td>Link</td> <td>Link/ACT</td> <td>Link</td> </tr> </tbody> </table> LINK = LED点亮; ACT = LED闪烁; LINK/ACT = LED点亮/闪烁。 速度 = LED点亮 (100BT); LED熄灭 (10BT); LED闪烁 (1000BT保留)。 双工 = LED点亮 (全双工); LED熄灭 (半双工)。	Control Bits [5:4]	00	01	10	11	LEDx_1	Speed	ACT	Duplex	Duplex	LEDx_0	Link/ACT	Link	Link/ACT	Link	R/W	00
Control Bits [5:4]	00	01	10	11															
LEDx_1	Speed	ACT	Duplex	Duplex															
LEDx_0	Link/ACT	Link	Link/ACT	Link															
3	保留	N/A, 不要更改。	RO	0															
2	保留	N/A, 不要更改。	RO	0															
1	REFCLKO使能	1 = 使能REFCLKO引脚时钟输出。 0 = 禁止REFCLKO引脚时钟输出。  配置脚输入选项: LED2_0 PU = 使能REFCLK_O (25 MHz) (默认)。 PD = 禁止REFCLK_O。  <b>注:</b> 该时钟为附加时钟。如果系统需要该时钟源, 则可以节省一个振荡器; 如果系统不需要该25 MHz时钟源, 则将其禁止。	R/W	0															
0	SPI读取采样时钟边沿选择	选择用于采样SPI读数据的SPI时钟边沿。  1 = 在SPI时钟的上升沿触发 (针对高速SPI)。 0 = 在SPI时钟的下降沿触发。	R/W	0															
<b>寄存器12 (0x0C) : 全局控制10</b>																			
7-6	保留	保留	RO	01															
5-2	保留	N/A, 不要更改。	RO	0001															

**表4-3: 全局寄存器 (续)**

地址	名称	说明	模式	默认值
1	尾部标记使能	尾部标记功能仅适用于端口5。 1 = 在FCS之前插入1个字节的数据。 0 = 不插入数据。	R/W	0
0	传送流控制数据包	1 = 开关不会过滤802.3x“流控制”数据包。 0 = 开关会过滤802.3x“流控制”数据包。	R/W	0
<b>寄存器 13 (0x0D) : 全局控制 11</b>				
7-0	出厂测试	N/A, 不要更改。	RO	00000000
<b>寄存器 14 (0x0E) : 掉电管理控制 1</b>				
7-6	保留	N/A, 不要更改。	RO	00
5	PLL 掉电	PLL掉电使能: 1 = 使能 0 = 禁止  注: 在电能检测模式 (EDPD 模式) 下使用	R/W	0
4-3	电源管理 模式选择	电源管理模式: 00 = 正常模式 (D0) 01 = 电能检测模式 (D2) 10 = 软掉电模式 (D3) 11 = 保留  注: RC 表示读取清零。	R/W (RC)	00
2-0	保留	N/A, 不要更改。	RO	000
<b>寄存器 15 (0x0F) : 掉电管理控制 2</b>				
7-0	Go_Sleep_Time [7:0]	电能检测模式开启时, 该值用于控制器件必须至少持续多长时间未检测到电能事件才进入低功耗状态。单位值为20 ms。go_sleep时间的默认值为1.6秒 (80 (十进制) × 20 ms)。	R/W	01010000

## 4.2 端口寄存器

以下寄存器用于使能基于端口分配的功能。寄存器位分配对于所有端口均相同, 但各个端口的地址会有所不同, 如下文所述。

**表4-4: 端口寄存器**

地址	名称	说明	模式	默认值
<b>寄存器 16 (0x10) : 端口 1 控制 0</b>				
<b>寄存器 32 (0x20) : 端口 2 控制 0</b>				
<b>寄存器 48 (0x30) : 端口 3 控制 0</b>				
<b>寄存器 64 (0x40) : 端口 4 控制 0</b>				
<b>寄存器 80 (0x50) : 端口 5 控制 0</b>				
7	广播风暴保护使能	1 = 为端口上的传入数据包使能广播风暴保护。 0 = 禁止广播风暴保护。	R/W	0

# KSZ8795CLX

表4-4: 端口寄存器 (续)

地址	名称	说明	模式	默认值
6	DiffServ 优先级分类使能	1 = 为端口上的传入数据包使能DiffServ优先级分类。 0 = 禁止DiffServ功能。	R/W	0
5	802.1p 优先级分类使能	1 = 为端口上的传入数据包使能802.1p优先级分类。 0 = 为端口上的传入数据包禁止802.1p优先级分类。	R/W	0
4 – 3	基于端口的优先级分类使能	00 = 如果“Diffserv”或“802.1p”分类未使能或分类失败，则端口上的传入数据包将被归类为优先级0队列。 01 = 如果“Diffserv”或“802.1p”分类未使能或分类失败，则端口上的传入数据包将被归类为优先级1队列。 10 = 如果“Diffserv”或“802.1p”分类未使能或分类失败，则端口上的传入数据包会被归类为优先级2队列。 11 = 如果“Diffserv”或“802.1p”分类未使能或分类失败，则端口上的传入数据包会被归类为优先级3队列。  注：可同时使能“DiffServ”、“802.1p”和端口优先级。802.1p与DSCP的逻辑或运算结果会覆盖端口优先级。	R/W	00
2	标记插入	1 = 当端口上有数据包输出时，如果接收到的数据包不含802.1q标记，则开关会为其添加802.1q标记。此开关不会为带标记的数据包添加标记。插入的标记为传入端口的“端口VID”。 0 = 禁止标记插入。	R/W	0
1	标记删除	1 = 当端口上有数据包输出时，如果接收到的数据包带802.1q标记，则开关会将802.1q标记删除。如果接收到的数据包不带标记，则开关不会对其进行修改。 0 = 禁止标记删除。	R/W	0
0	2队列划分使能	对于端口1-5，寄存器16/32/48/64/80中的bit[0]应与寄存器177/193/209/225/241 bit[1]结合使用。这将选择划分成1、2还是4个队列：  对于端口1、寄存器177 bit[1]和寄存器16 bit[0]：  11 = 保留 10 = 端口输出队列被划分为四个优先级队列（或将802.1p映射到优先级0-3模式）。 01 = 端口输出队列被划分为两个优先级队列（或将802.1p映射到优先级0-3模式）。 00 = 端口中存在单个输出队列。不存在优先级差异，即使数据包被分为高优先级或低优先级亦如此。	R/W	0

**表4-4: 端口寄存器 (续)**

地址	名称	说明	模式	默认值
<b>寄存器 17 (0x11): 端口 1 控制 1</b> <b>寄存器 33 (0x21): 端口 2 控制 1</b> <b>寄存器 49 (0x31): 端口 3 控制 1</b> <b>寄存器 65 (0x41): 端口 4 控制 1</b> <b>寄存器 81 (0x51): 端口 5 控制 1</b>				
7	嗅探器端口	1 = 端口被指定为嗅探器端口, 并且将发送受监视的数据包。 0 = 端口为常规端口。	R/W	0
6	接收嗅探	1 = 端口上接收到的所有数据包均会被标记为“受监视的数据包”, 并会被转发至指定的“嗅探器端口”。 0 = 无接收监视。	R/W	0
5	发送嗅探	1 = 端口上发送的所有数据包均会被标记为“受监视的数据包”, 并会被转发至指定的“嗅探器端口”。 0 = 无发送监视。	R/W	0
4 - 0	端口 VLAN 成员	定义端口的端口 VLAN 成员。 bit[4] 表示端口 5, bit[3] 表示端口 4, bit[2] 表示端口 3, bit[1] 表示端口 2, bit[0] 表示端口 1。 该端口只能在成员内进行通信。1 表示端口包含在成员中; 0 表示端口不包含在成员中。	R/W	0x1f
<b>寄存器 18 (0x12): 端口 1 控制 2</b> <b>寄存器 34 (0x22): 端口 2 控制 2</b> <b>寄存器 50 (0x32): 端口 3 控制 2</b> <b>寄存器 66 (0x42): 端口 4 控制 2</b> <b>寄存器 82 (0x52): 端口 5 控制 2</b>				
7	用户优先级上限	1 = 如果数据包的“用户优先级字段”高于端口默认标记寄存器中的“用户优先级字段”, 则将数据包的“用户优先级字段”替换为端口默认标记寄存器控制 3 中的“用户优先级字段”。 0 = 不将数据包的优先级字段替换为端口控制 3 寄存器 bit[7:5] 的端口默认标记优先级字段。	R/W	0
6	传入 VLAN 过滤	1 = 开关会丢弃 VLAN 表 bit[11:7] 中 VID 端口成员不包含传入端口的数据包。 0 = 无传入 VLAN 过滤。	R/W	0
5	丢弃非 PVID 数据包	1 = 开关会丢弃 VID 与传入端口默认 VID 不匹配的数据包。 0 = 不会丢弃任何数据包。	R/W	0
4	强制流控制	1 = 无论 AN 结果如何, 都会使能端口的接收与发送流控制。 0 = 基于 AN 结果使能流控制 (默认值)。	R/W	0
3	背压使能	1 = 使能端口半双工背压。 0 = 禁止端口半双工背压。	R/W	0
2	发送使能	1 = 使能端口上的数据包发送。 0 = 禁止端口上的数据包发送。	R/W	1

# KSZ8795CLX

表4-4: 端口寄存器 (续)

地址	名称	说明	模式	默认值
1	接收使能	1 = 使能端口上的数据包接收。 0 = 禁止端口上的数据包接收。	R/W	1
0	学习禁止	1 = 禁止开关地址学习功能。 0 = 使能开关地址学习功能。	R/W	0
<b>寄存器 19 (0x13): 端口 1 控制 3</b> <b>寄存器 35 (0x23): 端口 2 控制 3</b> <b>寄存器 51 (0x33): 端口 3 控制 3</b> <b>寄存器 67 (0x43): 端口 4 控制 3</b> <b>寄存器 83 (0x53): 端口 5 控制 3</b>				
7-0	默认标记 [15:8]	端口的默认标记, 其中包含: 7-5: 用户优先级位 4: CFI 位 3-0: VID [11:8]	R/W	0
<b>寄存器 20 (0x14): 端口 1 控制 4</b> <b>寄存器 36 (0x24): 端口 2 控制 4</b> <b>寄存器 52 (0x34): 端口 3 控制 4</b> <b>寄存器 68 (0x44): 端口 4 控制 4</b> <b>寄存器 84 (0x54): 端口 5 控制 4</b>				
7-0	默认标记 [7:0]	默认端口 1 的标记, 其中包含: 7-0: VID [7:0]	R/W	1
寄存器 19 和 20 (以及对应于其他端口的寄存器) 有两个用途: ——与传入无标记数据包相关, 用于传出标记。 ——传入无标记数据包或空 VID 带标记数据包的默认 VID, 用于地址查找。				
<b>寄存器 21 (0x15): 端口 1 控制 5</b> <b>寄存器 37 (0x25): 端口 2 控制 5</b> <b>寄存器 53 (0x35): 端口 3 控制 5</b> <b>寄存器 69 (0x45): 端口 4 控制 5</b> <b>寄存器 85 (0x55): 端口 5 控制 5</b>				
7-3	保留	N/A, 不要更改。	RO	00000
2	ACL 使能	1 = 使能 ACL 0 = 禁止 ACL	R/W	0
1-0	AUTHENTICATION_ MODE	这些位控制基于端口的验证: 00 和 10 = 禁止验证, 允许所有通信 (强制已认证); 如果使能 ACL, 则 ACL 丢失时所有通信都会通过 01 = 使能验证, 阻止所有通信; 如果使能 ACL, 则 ACL 丢失时通信会受阻 11 = 使能验证, 所有通信都会受限于 CPU 端口; 如果使能 ACL, 则只有在 ACL 丢失时通信才会受限于端口 5 CPU 端口。	R/W	00
<b>寄存器 22 (0x16): 保留</b> <b>寄存器 38 (0x26): 保留</b> <b>寄存器 54 (0x36): 保留</b> <b>寄存器 70 (0x46): 保留</b> <b>寄存器 86 (0x56): 端口 5 接口控制 6</b>				
7	RMII_CLK_SEL	端口 5 SW5-RMII 模式选择 1 = RMII 使用内部时钟 (时钟模式) 0 = RMII 使用外部时钟 (正常模式) 配置脚输入选项: LED2_1 PU = SW5-RMII 处于时钟模式下 (默认)。 PD = SW5-RMII 处于正常模式下。  注: 该引脚具有内部上拉	R/W	1



**表4-4: 端口寄存器 (续)**

地址	名称	说明	模式	默认值
6	Is_1Gbps	<p>1 = 在 GMII/RGMII 模式下, 为端口 5 选择 1 Gbps。                      0 = 在 GMII/RGMII 模式下, 为端口 5 选择 10/100 Mbps。                      配置脚输入选项: LED1_0                      PU = SW5-GMII/RGMII 模式下为 1 Gbps (默认)                      PD = SW5-GMII/RGMII 模式下为 10/100 Mbps</p> <p><b>注:</b> 该引脚具有内部上拉。                      在 10/100 Mbps 模式下, 使用寄存器 6 (全局控制 4) 的 bit[4] 针对速度 10 或 100 进行设置。</p>	R/W	1
5	保留	N/A, 不要更改。	RO	1
4	RGMII 内部延时 (Internal Delay, ID) 传入使能	<p>使能传入 RGMII-ID 模式                      1 = 使能传入 RGMII-ID。将向传入时钟输入中添加内部延时。                      0 = 不添加延时, 仅应用时钟到数据偏移。  <b>注:</b> 如果 RGMII 连接伙伴在标准规范值 <math>\pm 0.5</math> ns 的时间内发送数据到时钟偏移 (未在 PCB 上插入延时), 则将 bit[4] 置 1 会添加一段传入延时, 从而满足输入偏移最小值 1 ns 到最大值 2.6 ns 的要求 (时钟走线应与 PCB 布局中的数据走线等长)。</p>	R/W	0
3	RGMII 内部延时 (ID) 传出使能	<p>使能传出 RGMII-ID 模式                      1 = 使能传出 RGMII-ID。将向传出时钟输出中添加内部延时。                      0 = 不添加延时, 仅应用时钟到数据偏移。  <b>注:</b> 如果将 bit[3] 置 1, RGMII 发送时钟会添加一段内部传出延时, 从而为接收端添加最短 1 ns 的数据到时钟偏移, 接收端可能会也可能不会添加任何内部延时来满足其自身的接收时序要求。(如果无需额外的外部时钟偏移, 则时钟走线应与 PCB 布局中的数据走线等长)。</p>	R/W	1
2	GMII/MII 模式选择	<p>端口 5 GMAC5 SW5-GMII/MII 模式选择                      1 = GMII/MII 处于 GMAC/MAC 模式 (默认情况)。                      0 = GMII/MII 处于 GPHY/PHY 模式。                      配置脚选项: LED2_1                      PU = GMII/MII 处于 GMAC/MAC 模式 (默认情况)。                      PD = GMII/MII 处于 GPHY/PHY 模式。</p> <p><b>注:</b> 将 GMAC5 SW5-GMII 设为 GPHY 模式时, CRS 和 COL 引脚会从输入更改为输出。                      将 SW5-MII 设为 PHY 模式时, CRS、COL、RXC 和 TXC 引脚会从输入更改为输出。</p>	R/W	1

# KSZ8795CLX

表4-4: 端口寄存器 (续)

地址	名称	说明	模式	默认值
1-0	接口模式选择  注: 这适用于端口5 SW5-GMII/RGMII/MII/RMII	这些位用于为开关端口5 (SW5) 选择接口类型和模式。 端口5模式选择: 00 = MII 01 = RMII 10 = GMII 11 = RGMII 配置脚输入选项: LED3[1:0] 00 = MII 01 = RMII 10 = GMII 11 = RGMII (默认)  注: 这些引脚具有内部上拉。	R/W	11
寄存器23 (0x17): 端口1控制7 寄存器39 (0x27): 端口2控制7 寄存器55 (0x37): 端口3控制7 寄存器71 (0x47): 端口4控制7 寄存器87 (0x57): 保留		(注4-1)		
7-6	保留	N/A, 不要更改。	RO	00
5-4	Advertised_Flow_Control_Capability	这些位指示KSZ8795CLX已实现IEEE条款31和附录31B中定义的可选MAC控制子层和暂停功能(这两个功能可实现独立于速率和介质的全双工操作)。  00 = 无暂停 01 = 对称暂停 10 = 面向链路伙伴的非对称暂停 11 = 面向本地器件的对称暂停和非对称暂停 bit[5]指示支持非对称暂停。当bit[5]置1时, bit[4]的值用于指示链路中流支持的暂停帧的方向。非对称暂停配置可单独使能IEEE附录31B中定义的暂停接收和暂停发送功能。	R/W	11
3	通告的100BT全双工功能	1 = 通告100BT全双工功能。 0 = 阻止100BT全双工功能发送至链路伙伴。	R/W	1
2	通告的100BT半双工功能	1 = 通告100BT半双工功能。 0 = 阻止100BT半双工功能发送至链路伙伴。	R/W	1
1	通告的10BT全双工功能	1 = 通告10BT全双工功能。 0 = 阻止10BT全双工功能发送至链路伙伴。	R/W	1
0	通告的10BT半双工功能	1 = 通告10BT半双工功能。 0 = 阻止10BT半双工功能发送至链路伙伴。	R/W	1

**表4-4: 端口寄存器 (续)**

地址	名称	说明	模式	默认值
<b>寄存器 24 (0x18): 端口1状态0</b> <b>寄存器 40 (0x28): 端口2状态0</b> <b>寄存器 56 (0x38): 端口3状态0</b> <b>寄存器 72 (0x48): 端口4状态0</b> <b>寄存器 88 (0x58): 保留</b>				
7-6	保留	N/A, 不要更改。	RO	0000
5-4	Partner_Flow_Control_Capable	这些位指示伙伴是否具备IEEE条款31和附录31B中定义的可选MAC控制子层和暂停功能(这两个功能可实现独立于速率和介质的全双工操作)。 00 = 无暂停 01 = 对称暂停 10 = 面向链路伙伴的非对称暂停 11 = 面向本地器件的对称暂停和非对称暂停	RO	00
3	伙伴 100BT 全双工功能	1 = 链路伙伴具备 100BT 全双工功能。 0 = 链路伙伴不具备 100BT 全双工功能。	RO	0
2	伙伴 100BT 半双工功能	1 = 链路伙伴具备 100BT 半双工功能。 0 = 链路伙伴不具备 100BT 半双工功能。	RO	0
1	伙伴 10BT 全双工功能	1 = 链路伙伴具备 10BT 全双工功能。 0 = 链路伙伴不具备 10BT 全双工功能。	RO	0
0	伙伴 10BT 半双工功能	1 = 链路伙伴具备 10BT 半双工功能。 0 = 链路伙伴不具备 10BT 半双工功能。	RO	0
<b>寄存器 25 (0x19): 端口1状态1</b> <b>寄存器 41 (0x29): 端口2状态1</b> <b>寄存器 57 (0x39): 端口3状态1</b> <b>寄存器 73 (0x49): 端口4状态1</b> <b>寄存器 89 (0x59): 保留</b>		(注4-1)		
7	HP_MDIX	1 = HP 自动MDI/MDI-X 模式 0 = Microchip 自动MDI/MDI-X 模式	R/W	1
6	出厂测试	N/A, 不要更改。	RO	0
5	Polrvs	1 = 极性是相反的 0 = 极性不是相反的	RO	0
4	发送流控制使能	1 = 发送流控制功能有效 0 = 发送流控制功能无效	RO	0
3	接收流控制使能	1 = 接收流控制功能有效 0 = 接收流控制功能无效	RO	0
2	工作速度	1 = 链路速度为 100 Mbps 0 = 链路速度为 10 Mbps	RO	0
1	双工模式	1 = 链路为全双工 0 = 链路为半双工	RO	0
0	保留	N/A, 不要更改。	RO	0

# KSZ8795CLX

表4-4: 端口寄存器 (续)

地址	名称	说明	模式	默认值
寄存器 26 (0x1A) : 端口 1 PHY 控制 8 寄存器 42 (0x2A) : 端口 2 PHY 控制 8 寄存器 58 (0x3A) : 端口 3 PHY 控制 8 寄存器 74 (0x4A) : 端口 4 PHY 控制 8 寄存器 90 (0x5A) : 保留 (注 4-1)				
7	CDT 短于 10m 注: CDT 的全称为电缆诊断测试	1 = 短于 10 米	RO	0
6 – 5	CDT_Result	00 = 正常情况 01 = 检测到电缆中存在开路情况 10 = 检测到电缆中存在短路情况 11 = 电缆诊断测试失败	RO	00
4	CDT_Enable	1 = 使能电缆诊断测试。完成 CDT 测试后, 该位会自清零。 0 = 指示电缆诊断测试 (如果已使能) 已完成。	R/W (SC)	0
3	Force_Link	1 = 强制链路传递 0 = 正常工作	R/W	0
2	Pwrsave	1 = 使能节能功能 0 = 禁止节能功能	R/W	0
1	远程环回	1 = 执行远程环回, 端口 1 的环回如下: 端口 1 (寄存器 26, bit[1] = 1) 开始: RXP1/RXM1 (端口 1) 环回: 端口 1 PHY 的 PMD/PMA 结束: TXP1/TXM1 (端口 1) 将寄存器 42、58 和 74 的 bit[1] 置 1 会对端口 2、3 和 4 执行远程环回。 0 = 正常工作。	R/W	0
0	CDT_Fault_Count[8]	CDT 故障计数的 bit[8] 到故障的距离。 约为 $0.4 \times \text{CDT\_Fault\_Count}[8:0]$ 。	RO	0
寄存器 27 (0x1B) : 端口 1 LinkMD 结果 寄存器 43 (0x2B) : 端口 2 LinkMD 结果 寄存器 59 (0x3B) : 端口 3 LinkMD 结果 寄存器 75 (0x4B) : 端口 4 LinkMD 结果 寄存器 91 (0x5B) : 保留				
7 – 0	CDT_Fault_Count[7:0]	CDT 故障计数的 bit[7:0] 到故障的距离。约为 $0.4m \times \text{CDT\_Fault\_Count}[8:0]$	RO	0x00

**表4-4: 端口寄存器 (续)**

地址	名称	说明	模式	默认值
寄存器 28 (0x1C) : 端口1控制9 寄存器 44 (0x2C) : 端口2控制9 寄存器 60 (0x3C) : 端口3控制9 寄存器 76 (0x4C) : 端口4控制9 寄存器 92 (0x5C) : 保留		(注4-1)		
7	禁止自动协商	1 = 禁止自动协商。速度和双工模式由同一寄存器的bit[6:5]决定。 0 = 使能自动协商。	R/W	0
6	强制速度	1 = 如果禁止自动协商 (bit[7])，则强制100BT。 0 = 如果禁止自动协商 (bit[7])，则强制10BT。	R/W	1
5	强制双工模式	1 = 如果(1) AN被禁止或(2) AN被使能但失败，则强制全双工。 0 = 如果(1) AN被禁止或(2) AN被使能但失败，则强制半双工 (默认)。	R/W	0
4 - 0	保留	N/A, 不要更改。	RO	0x1f
寄存器 29 (0x1D) : 端口1控制10 寄存器 45 (0x2D) : 端口2控制10 寄存器 61 (0x3D) : 端口3控制10 寄存器 77 (0x4D) : 端口4控制10 寄存器 93 (0x5D) : 保留		(注4-1)		
7	LED熄灭	1 = 熄灭所有端口的LED (LEDx_2、LEDx_1和LEDx_0引脚，其中“x”为端口编号)。 如果该位置1，则这些引脚会被驱动为高电平。 0 = 正常工作。	R/W	0
6	TXIDS	1 = 禁止端口的发送器。 0 = 正常工作。	R/W	0
5	重新启动AN	1 = 重新启动自动协商。 0 = 正常工作。	R/W (SC)	0
4	保留	N/A, 不要更改	RO	0
3	掉电	1 = 掉电。 0 = 正常工作。	R/W	0
2	禁止自动MDI/MDI-X	1 = 禁止自动MDI/MDIX功能。 0 = 使能自动MDI/MDIX功能。	R/W	0
1	强制MDI	1 = 如果禁止自动MDI/MDIX，则会强制PHY进入MDI模式。 0 = MDI-X模式。	R/W	0
0	MAC环回	1 = 执行MAC环回。环回路径如下： 例如，设置端口1 MAC环回 (寄存器29, bit[0] = 1)，则会将端口2用作监视端口。数据包将进行传送。 开始：端口2接收 (也可开始接收端口3、4和5的数据包)。 环回：端口1的MAC。 结束：端口2发送 (也可分别在端口3、4和5结束)。 将寄存器45、61、77和93的bit[0]置1会分别对端口2、3、4和5执行MAC环回。 0 = 正常工作。	R/W	0

# KSZ8795CLX

表4-4: 端口寄存器 (续)

地址	名称	说明	模式	默认值
寄存器 30 (0x1E): 端口1状态2 寄存器 46 (0x2E): 端口2状态2 寄存器 62 (0x3E): 端口3状态2 寄存器 78 (0x4E): 端口4状态2 寄存器 94 (0x5E): 保留		(注4-1)		
7	MDIX 状态	1 = MDI。 0 = MDI-X。	RO	0
6	自动协商已完成	1 = 自动协商已完成。 0 = 自动协商尚未完成。	RO	0
5	链路良好	1 = 链路良好。 0 = 链路不良。	RO	0
4-0	保留	N/A, 不要更改。	RO	00000
寄存器 31 (0x1F): 端口1控制11和状态3 寄存器 47 (0x2F): 端口2控制11和状态3 寄存器 63 (0x3F): 端口3控制11和状态3 寄存器 79 (0x4F): 端口4控制11和状态3 寄存器 95 (0x5F): 保留			(注4-1)	
7	PHY 环回	1 = 执行PHY环回。环回路径如下: 示例- 设置端口1 PHY环回 (寄存器31, bit[7] = 1) 将端口2用作监视端口。数据包将进行传送。 开始: 端口2接收 (也可从端口3、4和5开始)。 环回: 端口1 PHY的PMD/PMA 结束: 端口2发送 (也可分别在端口3、4和5结束)。 将寄存器47、63、79和95的bit[7]置1会分别对端口2、3、4和5执行PHY环回。 0 = 正常工作。	R/W	0
6	保留	N/A, 不要更改	RO	0
5	PHY 隔离	1 = PHY与内部MII和TX+/TX-电气隔离。 0 = 正常工作。	R/W	0
4	软复位	1 = PHY软复位。该位自清零。 0 = 正常工作。	R/W (SC)	0
3	强制链路	1 = 强制链路处于PHY模式。 0 = 正常工作	R/W	0
2-0	端口工作模式指示	指示端口工作模式的当前状态: 000 = 保留 001 = 仍处于自动协商状态 010 = 10BASE-T半双工 011 = 100BASE-TX半双工 100 = 保留 101 = 10BASE-T全双工 110 = 100BASE-TX全双工 111 = 保留	RO	001

注4-1 端口控制7至11和端口状态1至3的内容可由MDC/MDIO接口通过标准MIIM寄存器进行访问。

### 4.3 高级控制寄存器

寄存器 104 至 109 定义开关引擎的 MAC 地址。此 48 位地址用作 MAC 暂停控制帧的源地址。

**表 4-5: 高级控制寄存器 104 至 109**

地址	名称	说明	模式	默认值
<b>寄存器 104 (0x68) : MAC 地址寄存器 0</b>				
7 - 0	MACA[47:40]	—	R/W	0x00
<b>寄存器 105 (0x69) : MAC 地址寄存器 1</b>				
7 - 0	MACA[39:32]	—	R/W	0x10
<b>寄存器 106 (0x6A) : MAC 地址寄存器 2</b>				
7 - 0	MACA[31:24]	—	R/W	0xA1
<b>寄存器 107 (0x6B) : MAC 地址寄存器 3</b>				
7 - 0	MACA[23:16]	—	R/W	0xff
<b>寄存器 108 (0x6C) : MAC 地址寄存器 4</b>				
7 - 0	MACA[15:8]	—	R/W	0xff
<b>寄存器 109 (0x6D) : MAC 地址寄存器 5</b>				
7 - 0	MACA[7:0]	—	R/W	0xff

使用寄存器 110 和 111 对静态 MAC 地址表、VLAN 表、动态地址表、PME 寄存器、ACL 表、EEE 寄存器和 MIB 计数器进行数据读写。

**表 4-6: 高级控制寄存器 110 至 111**

地址	名称	说明	模式	默认值
<b>寄存器 110 (0x6E) : 间接访问控制 0</b>				
7 - 5	EEE/ACL/ PME 间接寄 存器功能 选择	000 = 间接模式用于 bit[3:2] 中的表选择。当这些位不等于 000 时，bit[3:2] 用于 2 个附加的 MSB 地址位。 001 = 选择全局和基于端口的 EEE 寄存器，端口计数在 4 个 MSB 间接地址位中指定，8 位寄存器指针在 8 个 LSB 间接地址位中指定。 010 = 选择基于端口的 ACL 寄存器，端口计数在 4 个 MSB 间接地址位中指定，寄存器指针在 8 个 LSB 间接地址位中指定。 011 = 保留 100 = 选择 PME 控制寄存器。 101 = 使用 LinkMD 电缆诊断（请参见“LinkMD 电缆诊断”小节中的示例）。	R/W	000
4	读高电平 写低电平	1 = 读周期。 0 = 写周期。	R/W	0

# KSZ8795CLX

**表4-6: 高级控制寄存器 110 至 111 (续)**

地址	名称	说明	模式	默认值
3 - 2	表选择或间接地址 [11:10]	<p>如果 bit[6:5] = 00, 则                      00 = 选择静态 MAC 地址表。                      01 = 选择 VLAN 表。                      10 = 选择动态地址表。                      11 = 选择 MIB 计数器。</p> <p>如果 bit[6:5] 不等于 00, 则表示间接地址 [11:10] (间接地址的 MSB), 间接地址的 bit[11:8] 可用作端口地址, 而 bit[7:0] 可用作寄存器地址。</p> <p><b>注:</b>                      1. 寄存器 110 的 bit[3:0] 用于间接地址 bit[11:8] 的 4 个 MSB 位, 这四个位也用于端口间接寄存器。                      0000 = 全局间接寄存器                      0001 = 端口 1 间接寄存器                      0010 = 端口 2 间接寄存器                      0011 = 端口 3 间接寄存器                      0100 = 端口 4 间接寄存器                      0101 = 端口 5 间接寄存器                      2. 寄存器 111 bit[7:0] 用于间接地址位的 8 个 LSB, 这些位用于间接寄存器地址空间。</p>	R/W	00
1 - 0	间接地址 [9:8]	间接地址的 bit[9:8]。	R/W	00
<b>寄存器 111 (0x6F): 间接访问控制 1 (注 4-2)</b>				
7 - 0	间接地址 [7:0]	间接地址的 bit[7:0]。	R/W	00000000

**注 4-2** 写入寄存器 111 会触发一条命令, 具体是读访问还是写访问由寄存器 110 的 bit[4] 决定。

间接数据寄存器 112 至 120 用于静态表、VLAN、动态表、PME、EEE、ACL 和 MIB 计数器。

**表4-7: 高级控制寄存器 112 至 120**

地址	名称	说明	模式	默认值
<b>寄存器 112 (0x70): 间接数据寄存器 8</b>				
7 - 0	间接数据 [71:64]	间接数据的 bit[71:64]。	R/W	00000000
<b>寄存器 113 (0x71): 间接数据寄存器 7</b>				
7 - 0	间接数据 [63:56]	间接数据的 bit[63:56]。	R/W	00000000
<b>寄存器 114 (0x72): 间接数据寄存器 6</b>				
7 - 0	间接数据 [55:48]	间接数据的 bit[55:48]。	R/W	00000000
<b>寄存器 115 (0x73): 间接数据寄存器 5</b>				
7 - 0	间接数据 [47:40]	间接数据的 bit[47:40]。	R/W	00000000
<b>寄存器 116 (0x74): 间接数据寄存器 4</b>				
7 - 0	间接数据 [39:32]	间接数据的 bit[39:32]。	R/W	00000000



**表4-7: 高级控制寄存器 112 至 120 (续)**

地址	名称	说明	模式	默认值
<b>寄存器 117 (0x75): 间接数据寄存器 3</b>				
7 - 0	间接数据 [31:24]	间接数据的 bit[31:24]。	R/W	00000000
<b>寄存器 118 (0x76): 间接数据寄存器 2</b>				
7 - 0	间接数据 [23:6]	间接数据的 bit[23:16]。	R/W	00000000
<b>寄存器 119 (0x77): 间接数据寄存器 1</b>				
7 - 0	间接数据 [15:8]	间接数据的 bit[15:8]。	R/W	00000000
<b>寄存器 120 (0x78): 间接数据寄存器 0</b>				
7 - 0	间接数据 [7:0]	间接数据的 bit[7:0]。	R/W	00000000

所谓的间接字节寄存器即为仅用于 PME/ACL/EEE 间接寄存器访问的直接寄存器。间接字节寄存器 160 (0xA0) 用于读取/写入所有 PME、EEE 和 ACL 间接寄存器。

**表4-8: 高级控制寄存器 160、124 至 127**

地址	名称	说明	模式	默认值
<b>寄存器 160 (0xA0): 间接字节寄存器 (针对 PME、EEE 和 ACL 寄存器)</b>				
7 - 0	间接字节 [7:0]	间接访问的字节数据。	R/W	00000000
<b>寄存器 124 (0x7C): 中断状态寄存器</b>				
7 - 5	保留	N/A, 不要更改。	RO	000
4	PME 中断状态	1 = PME 中断请求 0 = 正常  <b>注:</b> 该位反映了 PME 控制寄存器, 可通过写入 PME 控制寄存器来清零 该位在 PME 置为有效时置 1。写 1 以清零该位 (WC)	RO	0
3	端口 4 中断状态	1 = 端口 4 中断请求 0 = 正常  <b>注:</b> 端口 4 链路发生变化时, 该位会置 1。写 1 以清零该位 (WC)	R/WC	0
2	端口 3 中断状态	1 = 端口 3 中断请求 0 = 正常  <b>注:</b> 端口 3 链路发生变化时, 该位会置 1。写 1 以清零该位 (WC)	R/WC	0
1	端口 2 中断状态	1 = 端口 2 中断请求 0 = 正常  <b>注:</b> 端口 2 链路发生变化时, 该位会置 1。写 1 以清零该位 (WC)	R/WC	0
0	端口 1 中断状态	1 = 端口 1 中断请求 0 = 正常  <b>注:</b> 端口 1 链路发生变化时, 该位会置 1。写 1 以清零该位 (WC)	R/WC	0

# KSZ8795CLX

表4-8: 高级控制寄存器 160、124至 127 (续)

地址	名称	说明	模式	默认值
<b>寄存器 125 (0x7D) : 中断屏蔽寄存器</b>				
7 - 5	保留	N/A, 不要更改。	RO	000
4	PME 中断屏蔽	1 = 允许PME中断。 0 = 正常	R/W	0
3	端口4 中断屏蔽	1 = 允许端口4中断。 0 = 正常	R/W	0
2	端口3 中断屏蔽	1 = 允许端口3中断。 0 = 正常	R/W	0
1	端口2 中断屏蔽	1 = 允许端口2中断。 0 = 正常	R/W	0
0	端口1 中断屏蔽	1 = 允许端口1中断。 0 = 正常	R/W	0
<b>寄存器 126 (0x7E) : ACL 中断状态寄存器</b>				
7 - 5	保留	N/A, 不要更改。	RO	000
4 - 0	ACL_INT_ STATUS	ACL 中断状态, 每个端口对应1位 1 = 检测到ACL中断。 0 = 未检测到ACL中断。	RO	00000
<b>寄存器 127 (0x7F) : ACL 中断控制寄存器</b>				
7 - 5	保留	N/A, 不要更改。	RO	000
4 - 0	ACL_INT_ ENABLE	ACL 中断允许, 每个端口对应1位 1 = 允许ACL中断。 0 = 禁止ACL中断。	R/W	00000

寄存器 128 和 129 可用于从 802.1p 优先级字段 0 至 7 映射到开关的四个优先级队列 0 至 3。0x3 表示最高优先级队列（优先级 3），0x0 表示最低优先级队列（优先级 0）。

表4-9: 高级控制寄存器 128 至 129

地址	名称	说明	模式	默认值
<b>寄存器 128 (0x80) : 全局控制 12</b>				
7 - 6	Tag_0x3	IEEE 802.1p 映射。如果 IEEE 802.1p 标记的值为 0x3, 则该字段的值被用作帧的优先级。	R/W	0x1
5 - 4	Tag_0x2	IEEE 802.1p 映射。如果 IEEE 802.1p 标记的值为 0x2, 则该字段的值被用作帧的优先级。	R/W	0x1
3 - 2	Tag_0x1	IEEE 802.1p 映射。如果 IEEE 802.1p 标记的值为 0x1, 则该字段的值被用作帧的优先级。	R/W	0x0
1 - 0	Tag_0x0	IEEE 802.1p 映射。如果 IEEE 802.1p 标记的值为 0x0, 则该字段的值被用作帧的优先级。	R/W	0x0
<b>寄存器 129 (0x81) : 全局控制 13</b>				
7 - 6	Tag_0x7	IEEE 802.1p 映射。如果 IEEE 802.1p 标记的值为 0x7, 则该字段的值被用作帧的优先级。	R/W	0x3

**表4-9: 高级控制寄存器 128 至 129 (续)**

地址	名称	说明	模式	默认值
5 - 4	Tag_0x6	IEEE 802.1p映射。如果IEEE 802.1p标记的值为0x6, 则该字段的值被用作帧的优先级。	R/W	0x3
3 - 2	Tag_0x5	IEEE 802.1p映射。如果IEEE 802.1p标记的值为0x5, 则该字段的值被用作帧的优先级。	R/W	0x2
1 - 0	Tag_0x4	IEEE 802.1p映射。如果IEEE 802.1p标记的值为0x4, 则该字段的值被用作帧的优先级。	R/W	0x2

**表4-10: 高级控制寄存器 130 至 135**

地址	名称	说明	模式	默认值
<b>寄存器 130 (0x82) : 全局控制 14</b>				
7 - 6	Pri_2Q[1:0]	选择二队列配置时, 这些Pri_2Q[1:0]位用于将IEEE 802.1p的2位结果从寄存器128/129 (或对于四队列, 将TOS/DiffServ的2位结果从寄存器144-159进行映射) 映射到两个队列低/高优先级。  IEEE 802.1p或TOS/DiffServ的2位结果 00 (0) = 映射到低优先级队列 01 (1) = Prio_2Q[0]映射到低/高优先级队列 10 (2) = Prio_2Q[1]映射到低/高优先级队列 11 (3) = 映射到高优先级队列  Pri_2Q[1:0]: 00 = 结果0、1和2为低优先级。3为高优先级。 01 = 不受支持, 应加以避免 10 = 结果0和1为低优先级。2和3为高优先级 (默认)。 11 = 结果0为低优先级。1、2和3为高优先级。	R/W	10
5 - 0	保留	N/A, 不要更改。	RO	001000
<b>寄存器 131 (0x83) : 全局控制 15</b>				
7 - 6	保留	N/A, 不要更改。	RO	10
5	未知单播数据包转发	1 = 使能支持未知单播数据包转发 0 = 禁止	R/W	0
4 - 0	未知单播数据包转发端口映射	00000 = 过滤未知单播数据包 00001 = 将未知单播数据包转发至端口1 00011 = 将未知单播数据包转发至端口1和端口2 00111 = 将未知单播数据包转发至端口1、端口2和端口3 01111 = 将未知单播数据包转发至端口1、端口2、端口3和端口4 11111 = 将未知单播数据包广播给所有端口	R/W	00000
<b>寄存器 132 (0x84) : 全局控制 16</b>				
7 - 6	保留	N/A, 不要更改。	RO	01

# KSZ8795CLX

表4-10: 高级控制寄存器130至135 (续)

地址	名称	说明	模式	默认值
5	未知多播数据包转发 (不包括IP多播数据包)	1 = 使能支持未知多播数据包转发 0 = 禁止	R/W	0
4 - 0	未知多播数据包转发端口映射	00000 = 过滤未知多播数据包 00001 = 将未知多播数据包转发至端口1 00011 = 将未知多播数据包转发至端口1和端口2 00111 = 将未知多播数据包转发至端口1、端口2和端口3 01111 = 将未知多播数据包转发至端口1、端口2、端口3和端口4 11111 = 将未知多播数据包广播给所有端口	R/W	00000
<b>寄存器133 (0x85) : 全局控制17</b>				
7 - 6	保留	N/A, 不要更改。	RO	00
5	未知VID数据包转发	1 = 使能支持未知VID数据包转发 0 = 禁止	R/W	0
4 - 0	未知VID数据包转发端口映射	00000 = 过滤未知VID数据包 00001 = 将未知VID数据包转发至端口1 00011 = 将未知VID数据包转发至端口1和端口2 00111 = 将未知VID数据包转发至端口1、端口2和端口3 01111 = 将未知VID数据包转发至端口1、端口2、端口3和端口4 11111 = 将未知VID数据包广播给所有端口	R/W	00000
<b>寄存器134 (0x86) : 全局控制18</b>				
7	保留	N/A, 不要更改。	RO	0
6	自寻址过滤使能	1 = 使能过滤自寻址单播和多播数据包 0 = 不过滤任何自寻址数据包  注: 自寻址过滤会过滤传出端口上的数据包, 自身MAC地址在寄存器104至109中分配。	R/W	0
5	未知IP多播数据包转发	1 = 使能支持未知IP多播数据包转发 0 = 禁止支持未知IP多播数据包转发	R/W	0

**表4-10: 高级控制寄存器 130至 135 (续)**

地址	名称	说明	模式	默认值
4 - 0	未知IP多播数据包转发端口映射	00000 = 过滤未知IP多播数据包 00001 = 将未知IP多播数据包转发至端口1 00011 = 将未知IP多播数据包转发至端口1和端口2 00111 = 将未知IP多播数据包转发至端口1、端口2和端口3 01111 = 将未知IP多播数据包转发至端口1、端口2、端口3和端口4 11111 = 将未知IP多播数据包广播给所有端口	R/W	00000
<b>寄存器 135 (0x87) : 全局控制 19</b>				
7 - 6	保留	N/A, 不要更改。	RO	00
5 - 4	传入速率限制周期	用于计算传入速率限制的单位周期: 00 = 16 ms 01 = 64 ms 1x = 256 ms	R/W	01
3	使能基于队列的传出速率限制	使能基于队列的传出速率限制 0 = 基于端口的传出速率限制 (默认) 1 = 基于队列的传出速率限制	R/W	0
2	插入源端口PVID标记选择使能	1 = 针对每个源端口PVID在传出端口上使能源端口PVID标记插入或非插入选项 (基于端口的控制8寄存器)。 0 = 禁止, 来自任何传入端口的所有数据包均将插入PVID (基于端口控制0寄存器bit[2])。	R/W	0
1 - 0	保留	N/A, 不要更改。	RO	00

Ipv4/Ipv6 TOS优先级控制寄存器实现了一个完全解码的64位差分服务代码点 (DSCP) 寄存器, 可用于从IP报头的6位TOS字段确定优先级。TOS字段的6个最高有效位被完全解码为64种可能, 产生的异常代码会被映射到DSCP寄存器的相应位中。

**表4-11: 高级控制寄存器 144至 159**

地址	名称	说明	模式	默认值
<b>寄存器 144 (0x90) : TOS优先级控制寄存器0</b>				
7 - 6	DSCP[7:6]	Ipv4和Ipv6映射 如果帧IP OS/DiffServ/通信类别值的bit[7:2]为0x03, 则该字段中的值被用作帧的优先级。	R/W	00
5 - 4	DSCP[5:4]	Ipv4和Ipv6映射 如果帧IP OS/DiffServ/通信类别值的bit[7:2]为0x02, 则该字段中的值被用作帧的优先级。	R/W	00
3 - 2	DSCP[3:2]	Ipv4和Ipv6映射 如果帧IP OS/DiffServ/通信类别值的bit[7:2]为0x01, 则该字段中的值被用作帧的优先级。	R/W	00

# KSZ8795CLX

表4-11: 高级控制寄存器 144 至 159 (续)

地址	名称	说明	模式	默认值
1 - 0	DSCP[1:0]	Ipv4 和 Ipv6 映射 如果帧 IP OS/DiffServ/通信类别值的 bit[7:2] 为 0x00, 则该字段中的值被用作帧的优先级。	R/W	00
<b>寄存器 145 (0x91) : TOS 优先级控制寄存器 1</b>				
7 - 6	DSCP[15:14]	Ipv4 和 Ipv6 映射 _ 用于值 0x07	R/W	00
5 - 4	DSCP[13:12]	Ipv4 和 Ipv6 映射 _ 用于值 0x06	R/W	00
3 - 2	DSCP[11:10]	Ipv4 和 Ipv6 映射 _ 用于值 0x05	R/W	00
1 - 0	DSCP[9:8]	Ipv4 和 Ipv6 映射 _ 用于值 0x04	R/W	00
<b>寄存器 146 (0x92) : TOS 优先级控制寄存器 2</b>				
7 - 6	DSCP[23:22]	Ipv4 和 Ipv6 映射 _ 用于值 0x0B	R/W	00
5 - 4	DSCP[21:20]	Ipv4 和 Ipv6 映射 _ 用于值 0x0A	R/W	00
3 - 2	DSCP[19:18]	Ipv4 和 Ipv6 映射 _ 用于值 0x09	R/W	00
1 - 0	DSCP[17:16]	Ipv4 和 Ipv6 映射 _ 用于值 0x08	R/W	00
<b>寄存器 147 (0x93) : TOS 优先级控制寄存器 3</b>				
7 - 6	DSCP[31:30]	Ipv4 和 Ipv6 映射 _ 用于值 0x0F	R/W	00
5 - 4	DSCP[29:28]	Ipv4 和 Ipv6 映射 _ 用于值 0x0E	R/W	00
3 - 2	DSCP[27:26]	Ipv4 和 Ipv6 映射 _ 用于值 0x0D	R/W	00
1 - 0	DSCP[25:24]	Ipv4 和 Ipv6 映射 _ 用于值 0x0C	R/W	00
<b>寄存器 148 (0x94) : TOS 优先级控制寄存器 4</b>				
7 - 6	DSCP[39:38]	Ipv4 和 Ipv6 映射 _ 用于值 0x13	R/W	00
5 - 4	DSCP[37:36]	Ipv4 和 Ipv6 映射 _ 用于值 0x12	R/W	00
3 - 2	DSCP[35:34]	Ipv4 和 Ipv6 映射 _ 用于值 0x11	R/W	00
1 - 0	DSCP[33:32]	Ipv4 和 Ipv6 映射 _ 用于值 0x10	R/W	00
<b>寄存器 149 (0x95) : TOS 优先级控制寄存器 5</b>				
7 - 6	DSCP[47:46]	Ipv4 和 Ipv6 映射 _ 用于值 0x17	R/W	00
5 - 4	DSCP[45:44]	Ipv4 和 Ipv6 映射 _ 用于值 0x16	R/W	00
3 - 2	DSCP[43:42]	Ipv4 和 Ipv6 映射 _ 用于值 0x15	R/W	00
1 - 0	DSCP[41:40]	Ipv4 和 Ipv6 映射 _ 用于值 0x14	R/W	00
<b>寄存器 150 (0x96) : TOS 优先级控制寄存器 6</b>				
7 - 6	DSCP[55:54]	Ipv4 和 Ipv6 映射 _ 用于值 0x1B	R/W	00
5 - 4	DSCP[53:52]	Ipv4 和 Ipv6 映射 _ 用于值 0x1A	R/W	00
3 - 2	DSCP[51:50]	Ipv4 和 Ipv6 映射 _ 用于值 0x19	R/W	00
1 - 0	DSCP[49:48]	Ipv4 和 Ipv6 映射 _ 用于值 0x18	R/W	00
<b>寄存器 151 (0x97) : TOS 优先级控制寄存器 7</b>				
7 - 6	DSCP[63:62]	Ipv4 和 Ipv6 映射 _ 用于值 0x1F	R/W	00
5 - 4	DSCP[61:60]	Ipv4 和 Ipv6 映射 _ 用于值 0x1E	R/W	00
3 - 2	DSCP[59:58]	Ipv4 和 Ipv6 映射 _ 用于值 0x1D	R/W	00
1 - 0	DSCP[57:56]	Ipv4 和 Ipv6 映射 _ 用于值 0x1C	R/W	00
<b>寄存器 152 (0x98) : TOS 优先级控制寄存器 8</b>				
7 - 6	DSCP[71:70]	Ipv4 和 Ipv6 映射 _ 用于值 0x23	R/W	00
5 - 4	DSCP[69:68]	Ipv4 和 Ipv6 映射 _ 用于值 0x22	R/W	00
3 - 2	DSCP[67:66]	Ipv4 和 Ipv6 映射 _ 用于值 0x21	R/W	00
1 - 0	DSCP[65:64]	Ipv4 和 Ipv6 映射 _ 用于值 0x20	R/W	00

**表4-11: 高级控制寄存器 144 至 159 (续)**

地址	名称	说明	模式	默认值
<b>寄存器 153 (0x99) : TOS 优先级控制寄存器 9</b>				
7 - 6	DSCP[79:78]	Ipv4 和 Ipv6 映射 _ 用于值 0x27	R/W	00
5 - 4	DSCP[77:76]	Ipv4 和 Ipv6 映射 _ 用于值 0x26	R/W	00
3 - 2	DSCP[75:74]	Ipv4 和 Ipv6 映射 _ 用于值 0x25	R/W	00
1 - 0	DSCP[73:72]	Ipv4 和 Ipv6 映射 _ 用于值 0x24	R/W	00
<b>寄存器 154 (0x9A) : TOS 优先级控制寄存器 10</b>				
7 - 6	DSCP[87:86]	Ipv4 和 Ipv6 映射 _ 用于值 0x2B	R/W	00
5 - 4	DSCP[85:84]	Ipv4 和 Ipv6 映射 _ 用于值 0x2A	R/W	00
3 - 2	DSCP[83:82]	Ipv4 和 Ipv6 映射 _ 用于值 0x29	R/W	00
1 - 0	DSCP[81:80]	Ipv4 和 Ipv6 映射 _ 用于值 0x28	R/W	00
<b>寄存器 155 (0x9B) : TOS 优先级控制寄存器 11</b>				
7 - 6	DSCP[95:94]	Ipv4 和 Ipv6 映射 _ 用于值 0x2F	R/W	00
5 - 4	DSCP[93:92]	Ipv4 和 Ipv6 映射 _ 用于值 0x2E	R/W	00
3 - 2	DSCP[91:90]	Ipv4 和 Ipv6 映射 _ 用于值 0x2D	R/W	00
1 - 0	DSCP[89:88]	Ipv4 和 Ipv6 映射 _ 用于值 0x2C	R/W	00
<b>寄存器 156 (0x9C) : TOS 优先级控制寄存器 12</b>				
7 - 6	DSCP [103:102]	Ipv4 和 Ipv6 映射 _ 用于值 0x33	R/W	00
5 - 4	DSCP [101:100]	Ipv4 和 Ipv6 映射 _ 用于值 0x32	R/W	00
3 - 2	DSCP[99:98]	Ipv4 和 Ipv6 映射 _ 用于值 0x31	R/W	00
1 - 0	DSCP[97:96]	Ipv4 和 Ipv6 映射 _ 用于值 0x30	R/W	00
<b>寄存器 157 (0x9D) : TOS 优先级控制寄存器 13</b>				
7 - 6	DSCP [111:110]	Ipv4 和 Ipv6 映射 _ 用于值 0x37	R/W	00
5 - 4	DSCP [109:108]	Ipv4 和 Ipv6 映射 _ 用于值 0x36	R/W	00
3 - 2	DSCP [107:106]	Ipv4 和 Ipv6 映射 _ 用于值 0x35	R/W	00
1 - 0	DSCP [105:104]	Ipv4 和 Ipv6 映射 _ 用于值 0x34	R/W	00
<b>寄存器 158 (0x9E) : TOS 优先级控制寄存器 14</b>				
7 - 6	DSCP [119:118]	Ipv4 和 Ipv6 映射 _ 用于值 0x3B	R/W	00
5 - 4	DSCP [117:116]	Ipv4 和 Ipv6 映射 _ 用于值 0x3A	R/W	00
3 - 2	DSCP [115:114]	Ipv4 和 Ipv6 映射 _ 用于值 0x39	R/W	00
1 - 0	DSCP [113:112]	Ipv4 和 Ipv6 映射 _ 用于值 0x38	R/W	00
<b>寄存器 159 (0x9F) : TOS 优先级控制寄存器 15</b>				
7 - 6	DSCP [127:126]	Ipv4 和 Ipv6 映射 _ 用于值 0x3F	R/W	00
5 - 4	DSCP [125:124]	Ipv4 和 Ipv6 映射 _ 用于值 0x3E	R/W	00
3 - 2	DSCP [123:122]	Ipv4 和 Ipv6 映射 _ 用于值 0x3D	R/W	00

# KSZ8795CLX

表4-11: 高级控制寄存器144至159 (续)

地址	名称	说明	模式	默认值
1 - 0	DSCP [121:120]	Ipv4和Ipv6映射_用于值0x3C	R/W	00

表4-12: 高级控制寄存器163至164

地址	名称	说明	模式	默认值
<b>寄存器163 (0xA3): 全局控制20</b>				
7	保留	N/A, 不要更改。	RO	0
6 - 4	GMII/RGMI 高速驱动强度	GMII和RGMI的高速接口驱动强度 000 = 2 mA 001 = 4 mA 010 = 8 mA 011 = 12 mA 100 = 16 mA 101 = 20 mA 110 = 24 mA (默认) 111 = 28 mA	R/W	110
3	保留	N/A, 不要更改。	RO	0
2 - 0	MII/RMII 低速驱动强度	MII和RMII的低速接口驱动强度 000 = 2 mA 001 = 4 mA 010 = 8 mA (默认) 011 = 12 mA 100 = 16 mA 101 = 20 mA 110 = 24 mA 111 = 28 mA	R/W	010
<b>寄存器164 (0xA4): 全局控制21</b>				
7 - 4	保留	N/A, 不要更改。	RO	0x2
3	IPv6 MLD 侦听选项	IPv6 MLD侦听选项 1 = 使能 0 = 禁止	R/W	0
2	IPv6 MLD 侦听使能	IPv6 MLD侦听使能 1 = 使能 0 = 禁止	R/W	0
1 - 0	保留	N/A, 不要更改。	RO	10



**表4-13: 其他高级控制寄存器 (注4-1)**

地址	名称	说明	模式	默认值
寄存器 176 (0xB0): 端口 1 控制 12 寄存器 192 (0xC0): 端口 2 控制 12 寄存器 208 (0xD0): 端口 3 控制 12 寄存器 224 (0xE0): 端口 4 控制 12 寄存器 240 (0xF0): 端口 5 控制 12				
7	保留	—	RO	1
6	通过所有帧	基于端口使能以通过所有帧 1 = 使能 0 = 禁止  <b>注:</b> 该位用于仅接收嗅探的端口镜像。	R/W	0
5 - 4	保留	—	RO	00
3	在编号最高的传出端口插入无标记数据包目标的源端口 PVID	寄存器 176: 在传出端口 5 插入无标记帧的源端口 1 PVID 寄存器 192: 在传出端口 5 插入无标记帧的源端口 2 PVID 寄存器 208: 在传出端口 5 插入无标记帧的源端口 3 PVID 寄存器 224: 在传出端口 5 插入无标记帧的源端口 4 PVID 寄存器 240: 在传出端口 4 插入无标记帧的源端口 5 PVID  <b>注:</b> 通过寄存器 135 的 bit[2] 使能。	R/W	0
2	在编号第二高的传出端口插入无标记数据包目标的源端口 PVID	寄存器 176: 在传出端口 4 插入无标记帧的源端口 1 PVID 寄存器 192: 在传出端口 4 插入无标记帧的源端口 2 PVID 寄存器 208: 在传出端口 4 插入无标记帧的源端口 3 PVID 寄存器 224: 在传出端口 3 插入无标记帧的源端口 4 PVID 寄存器 240: 在传出端口 3 插入无标记帧的源端口 5 PVID  <b>注:</b> 通过寄存器 135 的 bit[2] 使能。	R/W	0
1	在编号第二低的传出端口插入无标记数据包目标的源端口 PVID	寄存器 176: 在传出端口 3 插入无标记帧的源端口 1 PVID 寄存器 192: 在传出端口 3 插入无标记帧的源端口 2 PVID 寄存器 208: 在传出端口 2 插入无标记帧的源端口 3 PVID 寄存器 224: 在传出端口 2 插入无标记帧的源端口 4 PVID 寄存器 240: 在传出端口 2 插入无标记帧的源端口 5 PVID  <b>注:</b> 通过寄存器 135 的 bit[2] 使能。	R/W	0

# KSZ8795CLX

表4-13: 其他高级控制寄存器 (注4-1) (续)

地址	名称	说明	模式	默认值
0	在编号最低的传出端口插入无标记数据包目标的源端口 PVID	寄存器 176: 在传出端口 2 插入无标记帧的源端口 1 PVID 寄存器 192: 在传出端口 1 插入无标记帧的源端口 2 PVID 寄存器 208: 在传出端口 1 插入无标记帧的源端口 3 PVID 寄存器 224: 在传出端口 1 插入无标记帧的源端口 4 PVID 寄存器 240: 在传出端口 1 插入无标记帧的源端口 5 PVID  注: 通过寄存器 135 的 bit[2] 使能。	R/W	0
寄存器 177 (0xB1): 端口 1 控制 13 寄存器 193 (0xC1): 端口 2 控制 13 寄存器 209 (0xD1): 端口 3 控制 13 寄存器 225 (0xE1): 端口 4 控制 13 寄存器 241 (0xF1): 端口 5 控制 13				
7 - 2	保留	—	RO	000000
1	4 队列划分使能	该位与寄存器 16/32/48/64/80 bit[0] 共同用于选择划分成 1、2 还是 4 个队列:  { 寄存器 177 bit[1] 和寄存器 16 bit[0] = }:  11 = 保留。 10 = 端口输出队列被划分为四个优先级队列 (或将 802.1p 映射到优先级 0-3 模式)。 01 = 端口输出队列被划分为两个优先级队列 (或将 802.1p 映射到优先级 0-3 模式)。 00 = 端口中存在单个输出队列。不存在优先级差异, 即使数据包被分为较高和较低优先级亦如此。	R/W	0
0	使能丢弃标记	0 = 禁止带标记数据包丢弃 1 = 使能带标记数据包丢弃	R/W	0
寄存器 178 (0xB2): 端口 1 控制 14 寄存器 194 (0xC2): 端口 2 控制 14 寄存器 210 (0xD2): 端口 3 控制 14 寄存器 226 (0xE2): 端口 4 控制 14 寄存器 242 (0xF2): 端口 5 控制 14				
7	使能端口发送队列 3 比率	0 = 严格遵循优先级, 先发送该优先级队列 3 的所有数据包, 然后再发送较低优先级队列。 1 = bit[6:0] 反映在特定时间内允许从该优先级队列 3 中发送的数据包数。	R/W	1
6 - 0	端口发送队列 3 比率 [6:0]	四队列模式下最高优先级数据包中用于发送队列 3 的数据包数。	R/W	0001000

**表4-13: 其他高级控制寄存器 (注4-1) (续)**

地址	名称	说明	模式	默认值
<b>寄存器 179 (0xB3): 端口 1 控制 15</b> <b>寄存器 195 (0xC3): 端口 2 控制 15</b> <b>寄存器 211 (0xD3): 端口 3 控制 15</b> <b>寄存器 227 (0xE3): 端口 4 控制 15</b> <b>寄存器 243 (0xF3): 端口 5 控制 15</b>				
7	使能端口发送队列 2 比率	0 = 严格遵循优先级, 先发送该优先级队列 2 的所有数据包, 然后再发送较低优先级队列。 1 = bit[6:0] 反映在特定时间内允许从该优先级队列 1 中发送的数据包数。	R/W	1
6 - 0	端口发送队列 2 比率 [6:0]	四队列模式下高/低优先级数据包中用于发送队列 2 的数据包数。	R/W	0000100
<b>寄存器 180 (0xB4): 端口 1 控制 16</b> <b>寄存器 196 (0xC4): 端口 2 控制 16</b> <b>寄存器 212 (0xD4): 端口 3 控制 16</b> <b>寄存器 228 (0xE4): 端口 4 控制 16</b> <b>寄存器 244 (0xF4): 端口 5 控制 16</b>				
7	使能端口发送队列 1 比率	0 = 严格遵循优先级, 先发送该优先级队列 1 的所有数据包, 然后再发送较低优先级队列。 1 = bit[6:0] 反映在特定时间内允许从该优先级队列 1 中发送的数据包数。	R/W	1
6 - 0	端口发送队列 1 比率 [6:0]	四队列模式下低/高优先级数据包 (或两队列模式下高优先级数据包) 中用于发送队列 1 的数据包数。	R/W	0000010
<b>寄存器 181 (0xB5): 端口 1 控制 17</b> <b>寄存器 197 (0xC5): 端口 2 控制 17</b> <b>寄存器 213 (0xD5): 端口 3 控制 17</b> <b>寄存器 229 (0xE5): 端口 4 控制 17</b> <b>寄存器 245 (0xF5): 端口 5 控制 17</b>				
7	使能端口发送队列 0 比率	0 = 严格遵循优先级, 先发送该优先级队列 0 的所有数据包, 然后再发送较低优先级队列。 1 = bit[6:0] 反映在特定时间内允许从该优先级队列 0 中发送的数据包数。	R/W	1
6 - 0	端口发送队列 0 比率 [6:0]	四队列模式下最低优先级数据包 (或两队列模式下低优先级数据包) 中用于发送队列 0 的数据包数。	R/W	0000001
<b>寄存器 182 (0xB6): 端口 1 速率限制控制</b> <b>寄存器 198 (0xC6): 端口 2 速率限制控制</b> <b>寄存器 214 (0xD6): 端口 3 速率限制控制</b> <b>寄存器 230 (0xE6): 端口 4 速率限制控制</b> <b>寄存器 246 (0xF6): 端口 5 速率限制控制</b>				
7	保留	—	RO	0
6	传入限制基于端口/基于优先级的选择	1 = 传入速率限制基于端口 0 = 传入速率限制基于优先级	R/W	0

# KSZ8795CLX

表4-13: 其他高级控制寄存器 (注4-1) (续)

地址	名称	说明	模式	默认值
5	传入限制位/ 数据包模式 选择	1 = 速率限制基于数据包数进行计数。 0 = 速率限制基于位数进行计数。	R/W	0
4	传入速率限制 流控制使能	1 = 超出端口的接收速率时会流控制置为有效。 0 = 超出端口的接收速率时不会流控制置为有效。	R/W	0
3 - 2	限制模式	传入限制模式 这些位决定对哪些类型的帧进行限制并在传入速率 限制下进行计数。  00 = 对所有帧进行限制并计数。 01 = 对广播、多播和泛洪单播帧进行限制并计数。 10 = 仅对广播和多播帧进行限制并计数。 11 = 仅对广播帧进行限制并计数。	R/W	00
1	计数IFG	计数IFG字节 1 = 每个帧的最小帧间间隔。(IFG)字节(每帧 12个)包含在传入和传出速率限制计算中。 0 = 不对IFG字节进行计数。	R/W	0
0	计数前导	计数前导字节 1 = 每个帧的前导字节(每帧8个)均包含在传入和 传出速率限制计算中。 0 = 不对前导字节进行计数。	R/W	0
寄存器 183 (0xB7): 端口1 优先级0 传入限制控制1 寄存器 199 (0xC7): 端口2 优先级0 传入限制控制1 寄存器 215 (0xD7): 端口3 优先级0 传入限制控制1 寄存器 231 (0xE7): 端口4 优先级0 传入限制控制1 寄存器 247 (0xF7): 端口5 优先级0 传入限制控制1				
7	保留	—	RO	0
6 - 0	基于端口的 优先级0 传入限制	优先级0 帧的传入数据速率限制 根据“速率限制支持”小节中的表18对该端口的传 入流量进行整形。	R/W	0000000
寄存器 184 (0xB8): 端口1 优先级1 传入限制控制2 寄存器 200 (0xC8): 端口2 优先级1 传入限制控制2 寄存器 216 (0xD8): 端口3 优先级1 传入限制控制2 寄存器 232 (0xE8): 端口4 优先级1 传入限制控制2 寄存器 248 (0xF8): 端口5 优先级1 传入限制控制2				
7	保留	—	RO	0
6 - 0	基于端口的 优先级1 传入限制	优先级1 帧的传入数据速率限制 根据“速率限制支持”小节中的表18对该端口的传 入流量进行整形。	R/W	0000000
寄存器 185 (0xB9): 端口1 优先级2 传入限制控制3 寄存器 201 (0xC9): 端口2 优先级2 传入限制控制3 寄存器 217 (0xD9): 端口3 优先级2 传入限制控制3 寄存器 233 (0xE9): 端口4 优先级2 传入限制控制3 寄存器 249 (0xF9): 端口5 优先级2 传入限制控制3				
7	保留	—	RO	0

**表4-13: 其他高级控制寄存器 (注4-1) (续)**

地址	名称	说明	模式	默认值
6 - 0	基于端口的 优先级2 传入限制	优先级2帧的传入数据速率限制 根据“速率限制支持”小节中的表18对该端口的传入流量进行整形。	R/W	0000000
<b>寄存器 186 (0xBA) : 端口1优先级3传入限制控制4</b> <b>寄存器 202 (0xCA) : 端口2优先级3传入限制控制4</b> <b>寄存器 218 (0xDA) : 端口3优先级3传入限制控制4</b> <b>寄存器 234 (0xEA) : 端口4优先级3传入限制控制4</b> <b>寄存器 250 (0xFA) : 端口5优先级3传入限制控制4</b>				
7	基于端口的 传入速率限制使能	针对各优先级的传入数据速率限制设置生效 根据优先级传入限制控制, 触发用于所有优先级队列的端口传入速率限制引擎使其生效。  <b>注:</b> 对该寄存器进行的任何写操作都会根据优先级传入限制控制, 触发用于所有优先级队列的端口传入速率限制引擎使其生效。要使端口优先级0至3传入速率限制控制生效, 需要在配置端口传入限制控制1至4寄存器的bit[6:0]后分别将端口1、2、3、4和5对应的寄存器186、202、218、234和250的bit[7]置1。	R/W	0
6 - 0	基于端口的 优先级3 传入限制	优先级3帧的传入数据速率限制 根据“速率限制支持”小节中的表18对该端口的传入流量进行整形。	R/W	0000000
<b>寄存器 187 (0xBB) : 端口1队列0传出限制控制1</b> <b>寄存器 203 (0xCB) : 端口2队列0传出限制控制1</b> <b>寄存器 219 (0xDB) : 端口3队列0传出限制控制1</b> <b>寄存器 235 (0xEB) : 端口4队列0传出限制控制1</b> <b>寄存器 251 (0xFB) : 端口5队列0传出限制控制1</b>				
7	保留	—	RO	0
6 - 0	端口队列0 传出限制	优先级0帧的传出数据速率限制 根据“速率限制支持”小节中的表18对该端口的传出流量进行整形。 在四队列模式下, 为最低优先级。 在二队列模式下, 为低优先级。	R/W	0000000
<b>寄存器 188 (0xBC) : 端口1队列1传出限制控制2</b> <b>寄存器 204 (0xCC) : 端口2队列1传出限制控制2</b> <b>寄存器 220 (0xDC) : 端口3队列1传出限制控制2</b> <b>寄存器 236 (0xEC) : 端口4队列1传出限制控制2</b> <b>寄存器 252 (0xFC) : 端口5队列1传出限制控制2</b>				
7	保留	—	RO	0
6 - 0	端口队列1 传出限制	优先级1帧的传出数据速率限制 根据“速率限制支持”小节中的表18对该端口的传出流量进行整形。 在四队列模式下, 为低/高优先级。 在二队列模式下, 为高优先级。	R/W	0000000
<b>寄存器 189 (0xBD) : 端口1队列2传出限制控制3</b> <b>寄存器 205 (0xCD) : 端口2队列2传出限制控制3</b> <b>寄存器 221 (0xDD) : 端口3队列2传出限制控制3</b> <b>寄存器 237 (0xED) : 端口4队列2传出限制控制3</b> <b>寄存器 253 (0xFD) : 端口5队列2传出限制控制3</b>				
7	保留	—	RO	0

# KSZ8795CLX

表4-13: 其他高级控制寄存器 (注4-1) (续)

地址	名称	说明	模式	默认值
6-0	端口队列2 传出限制	优先级2帧的传出数据速率限制 根据“速率限制支持”小节中的表18对该端口的传出流量进行整形。 在四队列模式下, 为高/低优先级。	R/W	0000000
<b>寄存器 190 (0xBE): 端口1队列3传出限制控制4</b> <b>寄存器 206 (0xCE): 端口2队列3传出限制控制4</b> <b>寄存器 222 (0xDE): 端口3队列3传出限制控制4</b> <b>寄存器 238 (0xEE): 端口4队列3传出限制控制4</b> <b>寄存器 254 (0xFE): 端口5队列3传出限制控制4</b>				
7	保留	—	RO	0
6-0	端口队列3 传出限制	优先级3帧的传出数据速率限制 根据“速率限制支持”小节中的表18对该端口的传出流量进行整形。 在四队列模式下, 为最高优先级。	R/W	0000000

**注4-1** 在端口优先级0至3传入速率限制模式下, 需要将所有相关的传出端口设为二队列或四队列模式。  
在端口队列0至3传出速率限制模式下, 如果每个端口使用多个传出队列, 则最高优先级会基于速率选择表获取准确的速率限制, 而其他优先级数据包速率则基于端口控制14/15/16/17寄存器的比率。

表4-14: 高级控制寄存器191至255

地址	名称	说明	模式	默认值
<b>寄存器 191 (0xBF): 测试寄存器</b>				
7-0	保留	N/A, 不要更改。	RO	0x80
<b>寄存器 207 (0xCF): 保留控制寄存器</b>				
7-0	保留	N/A, 不要更改。	RO	0x15
<b>寄存器 223 (0xDF): 测试寄存器2</b>				
7-0	保留	N/A, 不要更改。	RO	0x0C
<b>寄存器 239 (0xEF): 测试寄存器3</b>				
7-0	保留	N/A, 不要更改。	RO	0x32
<b>寄存器 255 (0xFF): 测试寄存器4</b>				
7-0	保留	N/A, 不要更改。	RO	0x00

表4-15: 间接寄存器说明

控制	间接地址	内容
直接地址0x6E, 功能选择bit[7-5] = 000, Table_select bit[3-2] = 00	0x000至0x01F	静态MAC地址表条目0至31
直接地址0x6E, 功能选择bit[7-5] = 000, Table_select bit[3-2] = 01	0x000至0x1FF	VLAN表桶0至1023 (每个桶4个条目)
直接地址0x6E, 功能选择bit[7-5] = 000, Table_select bit[3-2] = 10	0x000至0x1FF	动态MAC地址表条目0至1023

**表4-15: 间接寄存器说明 (续)**

控制	间接地址	内容
直接地址 0x6E, 功能选择 bit[7-5] = 000, Table_select bit[3-2] = 11	0x000 至 0x08F 和 0x100 至 0x109	0x000 至 0x01F 端口 1 MIB 计数器 0x020 至 0x03F 端口 2 MIB 计数器 0x040 至 0x05F 端口 3 MIB 计数器 0x060 至 0x07F 端口 4 MIB 计数器 0x080 至 0x09F 端口 5 MIB 计数器 0x100 至 0x113 总字节和丢弃的 MIB 计数器
直接地址 0x6E, 功能选择 bit[7-5] = 001, bit[3-0] = 间接地址 bit[11-8] = MSB 间接地址 = 端口间接寄存器地址 0xn	{0xn, 6h00} 至 {0xn, 6h05}	基于端口的 16 位 EEE 控制寄存器 0 至 5 n = 端口号 使用间接字节寄存器 (0xA0)
直接地址 0x6E, 功能选择 bit[7-5] = 010, bit[3-0] = 间接地址 bit[11-8] = MSB 间接地址 = 端口间接寄存器地址 0xn	{0xn, 6h00} 至 {0xn, 6h1F}	ACL 条目 0 至 15, 例如 6h00 和 6h01 用于条目 0 n = 端口号 使用间接字节寄存器 (0xA0)
直接地址 0x6E, 功能选择 bit[7-5] = 011, bit[3-0] = 间接地址 bit[11-8] = MSB 间接地址 = 端口间接寄存器地址 0xn	{0xn, 8h00} 至 {0xn, 8h4FF}	保留为出厂设置。
直接地址 0x6E, 功能选择 bit[7-5] = 100, bit[3-0] = 间接地址 bit[11-8] = MSB 间接地址 = 端口间接寄存器地址 0xn	{0xn, 8h00} 至 {0xn, 8h4FF}	配置寄存器和 PME 等 n = 0 (全局) n = 1 至 4 (端口号) 使用间接字节寄存器 (0xA0)
直接地址 0x6E, 功能选择 bit[7-5] = 101, bit[3-0] = 间接地址 bit[11-8] = MSB 间接地址 = 端口间接寄存器地址 0xn	{0xn, 8h00} 至 {0xn, 8h4FF}	保留为出厂设置。

## 4.4 静态 MAC 地址表

KSZ8795CLX 包含一个静态地址表和一个动态地址表。请求 DA 查找时，会搜索这两个表，以便做出数据包转发决策。请求 SA 查找时，只会出于老化、移植和学习的目的搜索动态表。静态 DA 查找结果的优先级高于动态 DA 查找结果。如果两个表中均存在 DA 匹配，则会使用静态表中的结果。静态表只能由外部 SPI 主器件（通常为处理器）进行访问和控制。静态表中的条目不会被 KSZ8795CLX 删除。所有添加、修改和删除操作均由外部器件执行。

**注：**静态 MAC 表读与静态 MAC 表写的寄存器位分配有所不同，如下表所示。

**表4-16: 静态 MAC 地址表**

地址	名称	说明	模式	默认值
<b>静态 MAC 表读格式 (32 个条目)</b>				
63 - 57	FID	过滤器 VLAN ID, 表示 128 个有效 VLAN 中的一个。	RO	0000000
56	使用 FID	1 = 使用 (FID+MAC) 在静态表中进行查找。 0 = 仅使用 MAC 在静态表中进行查找。	RO	0
55	保留	—	RO	0

# KSZ8795CLX

表4-16: 静态MAC地址表 (续)

地址	名称	说明	模式	默认值
54	改写	1 = 改写成生成树“发送使能 = 0”或“接收使能 = 0”设置。该位用于实现生成树。 0 = 无改写。	RO	0
53	有效性	1 = 该条目有效, 将使用查找结果。 0 = 该条目无效。	RO	0
52 - 48	转发端口	这5个位用于控制转发端口。 例如: 00001 = 转发至端口1 00010 = 转发至端口2 00100 = 转发至端口3 01000 = 转发至端口4 10000 = 转发至端口5 00110 = 转发至端口2和端口3 11111 = 广播 (不包括传入端口)	RO	00000
47 - 0	MAC地址 (DA)	48位MAC地址。	RO	0x0
<b>静态MAC表写格式 (32个条目)</b>				
62 - 56	FID	过滤器VLAN ID, 表示128个有效VLAN中的一个。	W	0000000
55	使用FID	1 = 使用 (FID+MAC) 在静态表中进行查找。 0 = 仅使用MAC在静态表中进行查找。	W	0
54	改写	1 = 改写成生成树“发送使能 = 0”或“接收使能 = 0”设置。该位用于实现生成树。 0 = 无改写。	W	0
53	有效性	1 = 该条目有效, 将使用查找结果。 0 = 该条目无效。	W	0
52 - 48	转发端口	这5个位用于控制转发端口。 例如, 00001 = 转发至端口1 00010 = 转发至端口2 00100 = 转发至端口3 01000 = 转发至端口4 10000 = 转发至端口5 00110 = 转发至端口2和端口3 11111 = 广播 (不包括传入端口)	W	00000
47 - 0	MAC地址 (DA)	48位MAC地址。	W	0

示例:

- 静态地址表读 (读取第二个条目)  
将0x10写入寄存器110 (读取所选静态表)  
将0x1写入寄存器111 (触发读操作)

则

- 读寄存器 113 (63:56)
- 读寄存器 114 (55:48)
- 读寄存器 115 (47:40)
- 读寄存器 116 (39:32)



读寄存器 117 (31:24)

读寄存器 118 (23:16)

读寄存器 119 (15:8)

读寄存器 120 (7:0)

## 2. 静态地址表写（写入第八个条目）

写寄存器 113 (62:56)

写寄存器 114 (55:48)

写寄存器 115 (47:40)

写寄存器 116 (39:32)

写寄存器 117 (31:24)

写寄存器 118 (23:16)

写寄存器 119 (15:8)

写寄存器 120 (7:0)

将0x00写入寄存器110（写入所选静态表）

将0x7写入寄存器111（触发写操作）

## 4.5 VLAN表

VLAN表用于VLAN表查找。如果使能802.1q VLAN模式（寄存器5 bit[7] = 1），则该表用于检索与传入数据包相关的VLAN信息。VLAN表中有三个字段，分别用于FID（过滤器ID）、有效性和VLAN成员。使用该表之前，必须初始化这三个字段。不存在VID字段，因为4096个VID用作访问1024x52位存储空间中的专用存储器地址索引。每个条目有四个VLAN。每个VLAN包含13个位。四个VLAN需要52个位。总共有1024个条目，通过专用存储器地址和数据位支持4096个VLAN ID。FID有7个位用于支持128个有效VLAN。

表4-17: VLAN表

地址	名称	说明	模式	建议的初始值
<b>静态VLAN表的格式（最多支持4096个VLAN ID条目和128个有效VLAN）</b>				
12	有效性	1 = 该条目有效。 0 = 该条目无效。	R/W	0
11 - 7	成员	指定VLAN的成员端口。 如果DA查找失败（静态表和动态表中均无匹配），则与该VLAN相关的数据包将被转发至该字段中指定的端口。 例如，11001表示端口5、4和1在该VLAN中。	R/W	111111
6 - 0	FID	过滤器ID。KSZ8795CLX支持128个有效VLAN（由该七位位域表示）。FID为映射的ID。如果使能802.1q VLAN，则会基于FID+DA和FID+SA进行查找。	R/W	0

如果使能802.1q VLAN模式，则KSZ8795CLX会在数据包无标记或标记含空VID时为每个传入数据包分配一个VID，为数据包分配的VID为传入端口的默认端口VID。如果数据包标记含非空VID，则会使用标记中的VID。查找过程始于基于VID编号及其专用存储器地址和数据位进行的VLAN表查找。如果VLAN表中的条目无效，则会丢弃数据包并且不会进行地址学习。如果条目有效，则会检索FID。在MAC表中执行FID+DA和FID+SA查找。FID+DA查找用于确定转发端口。如果在MAC表中进行的FID+DA查找失败，则会基于VLAN表将数据包广播至所有成员或指定成员（不包括传入端口）。如果FID+SA失败，则会学习FID+SA。要在不同的有效VLAN之间进行通信，需设置同一FID；否则，设置不同的FID。

# KSZ8795CLX

VLAN表配置分为1024个VLAN组，每个VLAN组包含四个VLAN条目，最多支持4096个VLAN条目。每个VLAN组共有60个位，相邻条目之间插入三个反转位。实际上，有52个位用于VLAN组，这些位应在间接地址指定的时刻同时进行读取或写入。

VLAN组中的VLAN条目按如下所述映射到间接数据寄存器：

- Entry0[12:0]映射到VLAN组bit[12:0] {寄存器119[4:0]，寄存器120[7:0]}
- Entry1[12:0]映射到VLAN组bit[28:16] {寄存器117[4:0]，寄存器118[7:0]}
- Entry2[12:0]映射到VLAN组bit[44:32] {寄存器115[4:0]，寄存器116[7:0]}
- Entry3[12:0]映射到VLAN组bit[60:48] {寄存器113[4:0]，寄存器114[7:0]}

要读取一个VLAN条目，应先读取VLAN组，然后才能提取特定的VLAN条目信息。要更新任何VLAN条目，应先读取VLAN组，然后仅更新所需VLAN条目并写回整个VLAN组。VLAN表中的FID为7位，因此VLAN表支持独特的128个流VLAN组。在间接地址寄存器110和111中，每个VLAN组地址为10位长（最大为1024），其中VLAN组地址的bit[9:8]对应于寄存器110的bit[1:0]，VLAN组地址的bit[7:0]对应于寄存器111的bit[7:0]。每个读写操作最多可访问四个连续VLAN条目。

示例：

## 1. VLAN表读（读取VID = 2条目）

先写入间接控制和地址寄存器

将0x14写入寄存器110（0x6E）（读取所选VLAN表）

将0x0写入寄存器111（0x6F）（触发针对VID = 0、1、2和3条目的读操作）

然后针对VID = 2条目读取间接数据寄存器bit[38:26]

读寄存器115（0x73）（寄存器115 [4:0]为VLAN VID = 2条目的bit[12:8]）

读寄存器116（0x74）（寄存器116 [7:0]为VLAN VID = 2条目的bit[7:0]）

## 2. VLAN表写（写入VID = 10条目）

读取包含VID = 8、9、10和11的VLAN组。

将0x14写入寄存器110（0x6E）（读取所选VLAN表）

将0x02写入寄存器111（0x6F）（触发读操作和VID = 8、9、10和11间接地址）

先通过间接数据寄存器113、114、115、116、117、118、119和120读取VLAN组。

通过寄存器115 bit[4:0]和寄存器116 bit[7:0]修改间接数据寄存器bit[44:32]，具体如下：

写入寄存器115（0x73）（寄存器115 [4:0]为VLAN VID = 10条目的bit[12:8]）

写入寄存器116（0x74）（寄存器116 [7:0]为VLAN VID = 10条目的bit[7:0]）

然后写入间接控制和地址寄存器

将0x04写入寄存器110（0x6E）（写入所选VLAN表）

将0x02写入寄存器111（0x6F）（触发写操作和VID = 8、9、10和11间接地址）

表4-18给出了间接地址/数据寄存器和VLAN ID之间的关系。

表4-18: VLAN ID和间接寄存器

VLAN组的间接地址 高/低bit[9-0]	每个VLAN条目的间 接数据寄存器位	VID编号	VLAN标记中的VID bit[12-2]	VLAN标记中的 VID bit[1-0]
0	bit[12:0]	0	0	0
0	bit[28:16]	1	0	1
0	bit[44:32]	2	0	2
0	bit[60:48]	3	0	3
1	bit[12:0]	4	1	0
1	bit[28:16]	5	1	1
1	bit[44:32]	6	1	2
1	bit[60:48]	7	1	3
2	bit[12:0]	8	2	0

**表4-18: VLAN ID和间接寄存器 (续)**

VLAN组的间接地址 高/低 bit[9-0]	每个VLAN条目的间 接数据寄存器位	VID 编号	VLAN标记中的VID bit[12-2]	VLAN标记中的 VID bit[1-0]
2	bit[28:16]	9	2	1
2	bit[44:32]	10	2	2
2	bit[60:48]	11	2	3
:	:	:	:	:
:	:	:	:	:
:	:	:	:	:
1023	bit[12:0]	4092	1023	0
1023	bit[28:16]	4093	1023	1
1023	bit[44:32]	4095	1023	2
1023	bit[60:48]	4095	1023	3

## 4.6 动态MAC地址表

表4-19是只读的。

**表4-19: 动态MAC地址表**

地址	名称	说明	模式	默认值
<b>动态MAC地址表的格式 (1000个条目)</b>				
71	MAC为空	1 = 表格中不存在有效条目。 0 = 表格中存在有效条目。	RO	1
70 - 61	有效条目数	指示表格中的有效条目数。 0x3ff表示 1000个条目 0x1和 bit[71] = 0: 表示2个条目 0x0和 bit[71] = 0: 表示1个条目 0x0和 bit[71] = 1: 表示0个条目	RO	0
60 - 59	时间戳	用于内部老化的2位计数器	RO	—
58 - 56	源端口	用于学习FID+MAC的源端口。 000 = 端口1 001 = 端口2 010 = 端口3 011 = 端口4 100 = 端口5	RO	0x0
55	数据就绪	1 = 条目未就绪, 请重试, 直至该位被置为0。 0 = 条目已就绪。	RO	—
54 - 48	FID	过滤器ID	RO	0x0
47 - 0	MAC地址	48位MAC地址	RO	0x0

示例:

- 动态MAC地址表读 (读取第一个条目) 和检索MAC表大小  
 将0x18写入寄存器110 (读取所选动态表)  
 将0x0写入寄存器111 (触发读操作), 然后  
 读寄存器112 (71:64)  
 读寄存器113 (63:56); //以上两个寄存器显示条目数  
 读寄存器114 (55:48)//如果bit[55]为1, 则从该寄存器重新开始 (重新读取)

# KSZ8795CLX

读寄存器 115 (47:40)

读寄存器 116 (39:32)

读寄存器 117 (31:24)

读寄存器 118 (23:16)

读寄存器 119 (15:8)

读寄存器 120 (7:0)

## 2. 动态 MAC 地址表读（读取第 257 个条目），不检索条目数信息

将 0x19 写入寄存器 110（读取所选动态表）

将 0x1 写入寄存器 111（触发读操作），然后

读寄存器 112 (71:64)

读寄存器 113 (63:56)

读寄存器 114 (55:48)//如果 bit[55] 为 1，则从该寄存器重新开始（重新读取）

读寄存器 115 (47:40)

读寄存器 116 (39:32)

读寄存器 117 (31:24)

读寄存器 118 (23:16)

读寄存器 119 (15:8)

读寄存器 120 (7:0)

## 4.7 PME 间接寄存器

PME 寄存器基于全局和各个端口提供。这些寄存器通过间接存储器访问进行读/写，如表 4-20 所示。

表 4-20: PME 间接寄存器

地址	名称	说明	模式	默认值
<b>全局 PME 控制寄存器</b> 寄存器 110 (0x6E) bit[7:5] = 100 (用于 PME)，寄存器 110 bit[3:0] = 0x0 (用于间接全局寄存器)， 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。 偏移量: 0x00 (bit[31:24])、0x01 (bit[23:16])、0x02 (bit[15:8]) 和 0x03 (bit[7:0])。 位置: (100 PME) -> {0x0, 偏移量} -> 0xA0 保存数据。				
31 - 2	保留	—	RO	全 0
1	PME 输出使能	1 = 使能 PME 输出引脚。 0 = 禁止 PME 输出引脚。	R/W	0
0	PME 输出极性	1 = PME 输出引脚为高电平有效。 0 = PME 输出引脚为低电平有效。	R/W	0
<b>端口 PME 控制状态寄存器</b> 寄存器 110 (0x6E) bit[7:5] = 100 (用于 PME)，寄存器 110 bit[3:0] = 0xn (用于间接端口寄存器，其中 n = 1、2、3 或 4)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。 偏移量: 0x00 (bit[31:24])、0x01 (bit[23:16])、0x02 (bit[5:8]) 和 0x03 (bit[7:0])。 位置: (100 PME) -> {0xn, 偏移量} -> 0xA0 保存数据。				
31 - 3	保留	—	RO	全 0
2	魔术包检测	1 = 在任一端口处检测到魔术包 (写 1 以清零)。 0 = 未检测到魔术包。	R/W W1C	0
1	链路连通检测	1 = 在任一端口处检测到链路连通 (写 1 以清零)。 0 = 未检测到链路连通。	R/W W1C	0
0	电能检测	1 = 在任一端口处检测到电能 (写 1 以清零)。 0 = 未检测到电能。	R/W W1C	0

**表4-20: PME间接寄存器 (续)**

地址	名称	说明	模式	默认值
<b>端口PME控制掩码寄存器</b> 寄存器 110 (0x6E) bit[7:5] = 100 (用于PME)，寄存器 110 bit[3:0] = 0xn (用于端口，其中n = 1、2、3或4)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x04 (bit[31:24])、0x05 (bit[23:16])、0x06 (bit[15:8]) 和0x07 (bit[7:0])。 位置: (100 PME) -> {0xn, 偏移量} -> 0xA0保存数据。				
31 - 3	保留	—	RO	全0
2	魔术包检测使能	1 = 在主机QMU处检测到魔术包时，会将PME引脚置为有效。 0 = 检测到魔术包时不会将PME引脚置为有效。	R/W	0
1	链路连通检测使能	1 = 在任一端口处检测到链路连通时，会将PME引脚置为有效。 0 = 检测到链路连通时不会将PME引脚置为有效。	R/W	0
0	电能检测使能	1 = 在任一端口处检测到线路电能时，会将PME引脚置为有效。 0 = 检测到电能时不会将PME引脚置为有效。	R/W	0

**编程示例**

*读操作*

1. 使用间接访问控制寄存器选择要读取的寄存器以及读取全局PME控制寄存器。  
将0x90写入寄存器 110 (0x6E) //PME选择和读操作，端口号的4个MSB (寄存器110 bit[3:0]) = 0，用于全局PME寄存器。
2. 将0x03写入寄存器 111 (0x6F) //触发针对全局PME控制寄存器的bit[7:0]的读操作。
3. 读取间接字节寄存器 160 (0xA0) //获取全局PME控制寄存器的值。

*写操作*

1. 将0x80写入寄存器 110 (0x6E) //PME选择和写操作，端口号的4个MSB = 0，用于全局PME寄存器。
2. 将0x03写入寄存器 111 (0x6F) //选择写入全局PME控制地址寄存器的bit[7:0]。
3. 将新值写入间接字节寄存器 160 bit[7:0] (0xA0) //通过分配的间接数据寄存器地址将值写入间接PME数据寄存器的全局PME控制寄存器。

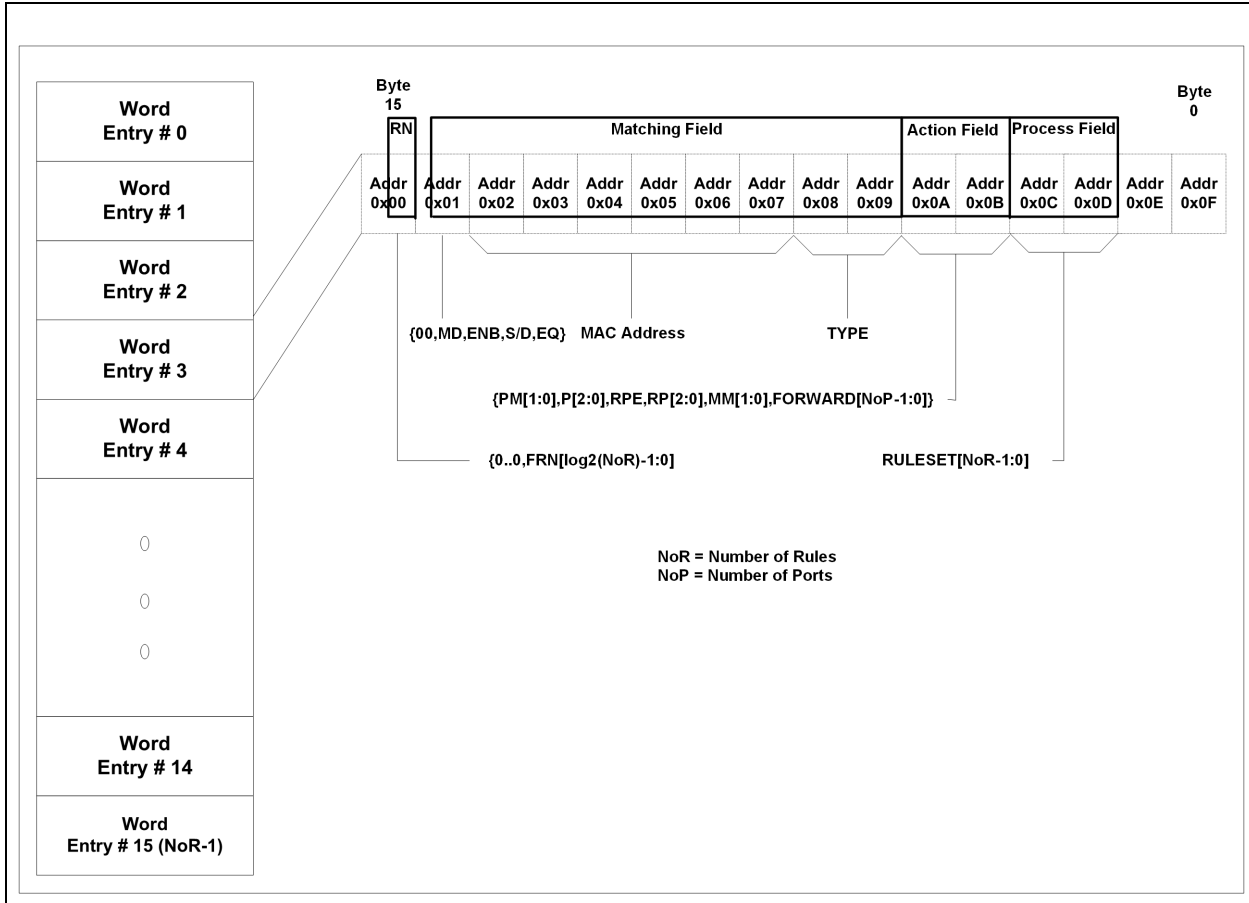
# KSZ8795CLX

## 4.8 ACL 规则表和 ACL 间接寄存器

### 4.8.1 ACL 寄存器和编程模型

ACL 寄存器由单片机通过串行接口进行访问。基于端口的寄存器组通过间接寻址机制进行访问。ACL 条目采用下图所示的格式进行存储。每个 ACL 规则列表最多可基于每个端口输入 16 个条目，共有五个 ACL 规则列表可针对五个端口进行设置。

图4-2: ACL 表访问



要更新任何基于端口的 ACL 寄存器，建议针对间接地址寄存器寻址的各个 128 位（使用了 112 个位）条目执行读修改写序列，以确保控制内容的完整性。对于每个 ACL 条目读访问，至少需要两个间接控制写操作和两个间接控制读操作（之后应进行间接数据读操作）；对于每个 ACL 条目写访问，至少需要一个间接控制读操作和三个间接控制写操作。每个 112 位基于端口的 ACL 字条目（ACL 字）通过一系列间接访问控制 0 寄存器 110（0x6E）访问来实现，具体方法是指定间接访问控制 1 寄存器 111（0x6F）中的 bit[3:0] 4 位端口号（间接地址[11:8]）和 8 位间接寄存器地址（间接地址[7:0]）。地址编号 0x00-0x0d 用于指定各个条目的字节位置（见上图），地址 0x00 指示各个 128 位条目的字节 15（MSB），地址 0x01 指示字节 14 等，地址 0x0E 和 0x0F 的字节保留供将来使用。地址 0x10 和 0x11 针对各个条目保持按位字节使能。地址 0x12 用作控制和状态寄存器。ACL 间接寄存器小节中定义了这些寄存器的格式。

### 4.8.2 ACL 间接寄存器

表 4-21 用于实现 ACL 模式选择和基于各个端口进行过滤。

**表4-21: 14字节ACL规则的ACL间接寄存器**

地址	名称	说明	模式	默认值
<b>Port_ACL_0</b> <b>ACL 端口寄存器 0 (0x00)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x00 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 <b>处理字段</b>				
7 - 4	保留	—	RO	0x0
3 - 0	FRN[3:0]	第一条规则编号 该位域用于规则集的第一条规则编号。在 ACL 规则表中, 每个端口共有 16 个条目。每条规则均可通过 ACL 端口寄存器 12 (0x0c) 和寄存器 13 (0x0d) 使用规则集的其他规则进行设置。 无论是单个规则还是规则集, 都必须按照 FRN[3:0] 指定一个将要使用哪个操作字段的条目。	R/W	0000
<b>Port_ACL_1</b> <b>ACL 端口寄存器 1 (0x01)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x01 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 <b>匹配字段</b>				
7 - 6	保留	—	RO	00
5 - 4	MD[1:0]	<b>MODE</b> 00 = 禁止当前规则列表, 不执行任何操作 01 = 限定规则以用于第 2 层 MAC 报头过滤 10 = 用于第 3 层 IP 地址过滤 11 = 执行第 4 层 TCP 端口号/协议过滤	R/W	00

# KSZ8795CLX

表4-21: 14字节ACL规则的ACL间接寄存器(续)

地址	名称	说明	模式	默认值
3 - 2	ENB[1:0]	<p>ENABLE</p> <p>MD = 01时: 00 = 操作字段中PM、P、REP和MM的11个位用于为与匹配字段中的MAC地址和TYPE匹配的数据包指定计数值。 计数单位在FORWARD字段bit[4]中定义: bit[4] = 0, 将使用µs。 bit[4] = 1, 将使用ms。</p> <p>FORWARDED字段bit[3]确定用于在计数器终止时产生中断的算法。bit[3] = 0, 11位计数器会装载列表中的计数值, 并开始每隔一个单位时间递减计数一次。计数结束(即, 在该值指定的时间内尚未接收下一个合格的数据包)时会产生中断。 bit[3] = 1, 每次接收到匹配的数据包时计数器都会递增, 并会在计数结束时产生中断, 之后计数会重置。</p> <p>01 = MAC地址位域正在进行测试。 10 = MAC TYPE位域用于测试。 11 = 会针对该列表中的这些位域对MAC地址和TYPE进行测试。</p> <p>MD = 10时: 00 = 保留。 01 = 使能IP地址和掩码或IP协议以进行相应的测试。 10 = 比较SA和DA; 基于E/Q位设置做出丢弃/转发决策。 11 = 保留</p> <p>MD = 11时: 00 = 使能协议比较。 01 = 选择TCP/UDP地址比较。 10 = 同01 11 = 比较TCP的序列号。</p>	R/W	00
1	S_D	<p>源/目标地址</p> <p>0 = 使用DA进行比较。 1 = 使用SA进行比较。</p>	R/W	0
0	EQ	<p>相等比较</p> <p>0 = 不等时匹配。 1 = 相等时匹配。</p>	R/W	0
<p><b>Port_ACL_2</b>  <b>ACL端口寄存器2 (0x02)</b>            寄存器110 (0x6E) bit[7:5] = 010 (用于ACL), 寄存器110 bit[3:0] = 0xn (用于端口1、2、3、4和5)。            寄存器111 (0x6F) bit[7:0] = 偏移量0x02 (用于访问间接字节寄存器0xA0)。            位置: (010 ACL) -&gt; {0xn, 偏移量} -&gt; 0xA0保存数据。  <b>第2层的匹配字段</b></p>				
7 - 0	MAC_ADDR [47:40]	MAC地址	R/W	00000000



**表4-21: 14字节ACL规则的ACL间接寄存器 (续)**

地址	名称	说明	模式	默认值
<b>Port_ACL_3</b> <b>ACL 端口寄存器 3 (0x03)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x03 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 2 层的匹配字段				
7 - 0	MAC_ADDR [39:32]	MAC 地址	R/W	00000000
<b>Port_ACL_4</b> <b>ACL 端口寄存器 4 (0x04)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x04 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 2 层的匹配字段				
7 - 0	MAC_ADDR [31:24]	MAC 地址	R/W	00000000
<b>Port_ACL_5</b> <b>ACL 端口寄存器 5 (0x05)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x05 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 2 层的匹配字段				
7 - 0	MAC_ADDR [23:16]	MAC 地址	R/W	00000000
<b>Port_ACL_6</b> <b>ACL 端口寄存器 6 (0x06)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x06 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 2 层的匹配字段				
7 - 0	MAC_ADDR [15:8]	MAC 地址	R/W	00000000
<b>Port_ACL_7</b> <b>ACL 端口寄存器 7 (0x07)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x07 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 2 层的匹配字段				
7 - 0	MAC_ADDR [7:0]	MAC 地址	R/W	00000000
<b>Port_ACL_8</b> <b>ACL 端口寄存器 8 (0x08)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x08 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 2 层的匹配字段				
7 - 0	TYPE[15:8]	以太网类型	R/W	00000000

# KSZ8795CLX

表4-21: 14字节ACL规则的ACL间接寄存器(续)

地址	名称	说明	模式	默认值
<b>Port_ACL_9</b> <b>ACL端口寄存器9(0x09)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x09(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第2层的匹配字段				
7-0	TYPE[7:0]	以太网类型	R/W	00000000
注: 匹配字段中的第2层、第3层和第4层应位于不同的条目中。相同层应位于同一条目中。				
<b>Port_ACL_2</b> <b>ACL端口寄存器2(0x02)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x02(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第3层的匹配字段				
7-0	IP_ADDR [31:24]	IP地址	R/W	00000000
<b>Port_ACL_3</b> <b>ACL端口寄存器3(0x03)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x03(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第3层的匹配字段				
7-0	IP_ADDR [23:16]	IP地址	R/W	00000000
<b>Port_ACL_4</b> <b>ACL端口寄存器4(0x04)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x04(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第3层IP的匹配字段				
7-0	IP_ADDR [15:8]	IP地址	R/W	00000000
<b>Port_ACL_5</b> <b>ACL端口寄存器5(0x05)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x05(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第3层的匹配字段				
7-0	IP_ADDR [7:0]	IP地址	R/W	00000000
<b>Port_ACL_6</b> <b>ACL端口寄存器6(0x06)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x06(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第3层的匹配字段				
7-0	IP_Mask [31:24]	IP掩码	R/W	00000000

表4-21: 14字节ACL规则的ACL间接寄存器(续)

地址	名称	说明	模式	默认值
<b>Port_ACL_7</b> <b>ACL 端口寄存器 7 (0x07)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x07 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 3 层的匹配字段				
7 - 0	IP_Mask [23:16]	IP 掩码	R/W	00000000
<b>Port_ACL_8</b> <b>ACL 端口寄存器 8 (0x08)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x08 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 3 层的匹配字段				
7 - 0	IP_Mask [15:8]	IP 掩码	R/W	00000000
<b>Port_ACL_9</b> <b>ACL 端口寄存器 9 (0x09)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x09 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 3 层的匹配字段				
7 - 0	IP_Mask [7:0]	IP 掩码	R/W	00000000
<b>注: 匹配字段中的第 2 层、第 3 层和第 4 层应位于不同的条目中。相同层应位于同一条目中。</b>				
<b>Port_ACL_2</b> <b>ACL 端口寄存器 2 (0x02)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x02 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 4 层的匹配字段				
7 - 0	MAX 端口 [15:8]	用于 TCP 端口号或序列号匹配的范围	R/W	00000000
<b>Port_ACL_3</b> <b>ACL 端口寄存器 3 (0x03)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x03 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 4 层的匹配字段				
7 - 0	MIN 端口 [7:0]	用于 TCP 端口号或序列号匹配的范围	R/W	00000000
<b>Port_ACL_4</b> <b>ACL 端口寄存器 4 (0x04)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x04 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。 第 4 层的匹配字段				
7 - 3	保留	—	RO	00000

# KSZ8795CLX

表4-21: 14字节ACL规则的ACL间接寄存器(续)

地址	名称	说明	模式	默认值
2 - 1	PC[1:0]	00 = 禁止端口比较。 01 = 匹配MAX或MIN之一。 10 = 端口号位于MIN和MAX之间时匹配。 11 = 端口号超出范围时匹配。	R/W	00
0	PRO[7]	IP协议 用于要匹配的IP协议	—	0
<b>Port_ACL_5</b> <b>ACL端口寄存器5(0x05)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x05(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第4层的匹配字段				
7 - 1	PRO[6:0]	IP协议 用于要匹配的IP协议	R/W	0000000
0	FME	标志匹配使能 0 = 禁止TCP FLAG匹配 1 = 使能TCP FLAG匹配	R/W	0
<b>Port_ACL_6</b> <b>ACL端口寄存器6(0x06)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x06(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第4层的匹配字段				
7 - 0	FMSK[7:0]	TCP FLAG掩码	R/W	00000000
<b>Port_ACL_7</b> <b>ACL端口寄存器7(0x07)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x07(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 第4层的匹配字段				
7 - 0	FLAG[7:0]	TCP FLAG	R/W	00000000
<b>Port_ACL_8</b> <b>ACL端口寄存器8(0x08)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x08(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。				
7 - 0	保留	—	RO	00000000
<b>Port_ACL_9</b> <b>ACL端口寄存器9(0x09)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL), 寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x09(用于访问间接字节寄存器0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。				
7 - 0	保留	—	RO	00000000
注: 匹配字段中的第2层、第3层和第4层应位于不同的条目中。相同层应位于同一条目中。				

**表4-21: 14字节ACL规则的ACL间接寄存器 (续)**

地址	名称	说明	模式	默认值
<b>Port_ACL_A</b>				
<b>ACL 端口寄存器 10 (0x0A)</b>				
寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。				
寄存器 111 (0x6F) bit[7:0] = 偏移量 0x0A (用于访问间接字节寄存器 0xA0)。				
位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。				
操作字段				
7 - 6	PM[1:0]	优先级模式 00 = 未选择任何优先级; 带标记的数据包中使用由 QoS/类别确定的优先级。 01 = 如果 P [2:0] 位域中的优先级高于接收的带标记数据包的 3 位优先级字段中的 QoS 结果, 则使用该位域中的优先级。 10 = 如果 P [2:0] 位域中的优先级低于接收的带标记数据包的 3 位优先级字段中的 QoS 结果, 则使用该位域中的优先级。 11 = P [2:0] 位域将替代接收的带标记数据包的 3 位优先级字段。	R/W	00
5 - 3	P[2:0]	优先级 注: 要使用的 3 位优先级值取决于 bit[7:6] 中的 PM [1:0] 设置。	R/W	000
2	RPE	备注优先级使能 0 = 无需备注。 1 = 数据包中的 VLAN 优先级位由列表中下一个位域 RP[2:1] 替代。	R/W	0
1 - 0	RP[2:1]	备注优先级 00 = 优先级 0 01 = 优先级 1 10 = 优先级 2 11 = 优先级 3	R/W	00
<b>Port_ACL_B</b>				
<b>ACL 端口寄存器 11 (0x0B)</b>				
寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。				
寄存器 111 (0x6F) bit[7:0] = 偏移量 0x0B (用于访问间接字节寄存器 0xA0)。				
位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。				
操作字段				
7	RP[0]	备注优先级	R/W	0
6 - 5	MM[1:0]	映射模式 00 = 无需转发重映射。请勿使用 FORWARD 字段中的转发映射; 仅使用查找表中的转发映射。 01 = 将 FORWARD 字段中的转发映射与查找表中的转发映射进行逻辑或运算。 10 = 将 FORWARD 字段中的转发映射与查找表中的转发映射进行逻辑与运算。 11 = FORWARD 字段中的转发映射会替代查找表中的转发映射。	R/W	00

# KSZ8795CLX

表4-21: 14字节ACL规则的ACL间接寄存器(续)

地址	名称	说明	模式	默认值
4 - 0	FORWARD [4:0]	端口映射 每个位都用于指示一个端口的转发决策。 bit[0] = 端口1 bit[1] = 端口2 bit[2] = 端口3 bit[3] = 端口4 bit[4] = 端口5  MD = 01且ENB = 00时, bit[4]用作计数单位: 0 = μs 1 = ms bit[3]用于选择计数模式: 0 = 11位计数器从操作字段PM、P、RPE、RP和MM中分配的值进行递减计数,计数结束后会产生中断。 1 = 11位计数器针对接收到的每个匹配数据包进行递增计数,直至达到操作字段PM、P、RPE、RP和MM中分配的值,随即会产生中断。  注:有关详细信息,请参见ENB字段说明。	R/W	—
<b>Port_ACL_C</b> <b>ACL端口寄存器12(0x0C)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL),寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x0C(用于访问间接字节寄存器0xA0)。 位置:(010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 处理字段				
7 - 0	RULESET [15:8]	规则集 每个位用于指示bit0至bit16中对应的条目(规则列表中共16个条目)是否可分配用于按端口级联的规则中使用的规则集。	R/W	00000000
<b>Port_ACL_D</b> <b>ACL端口寄存器13(0x0D)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL),寄存器110 bit[3:0] = 0xn(用于端口1、2、3、4和5)。 寄存器111(0x6F) bit[7:0] = 偏移量0x0D(用于访问间接字节寄存器0xA0)。 位置:(010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。 处理字段				
7 - 0	RULESET [7:0]	规则集 每个位用于指示bit0至bit16中对应的条目(规则列表中共16个条目)是否可分配用于按端口级联的规则中使用的规则集。	R/W	00000000

表4-22: 14字节ACL规则的临时存储

地址	名称	说明	模式	默认值
<b>Port_ACL_BYTE_ENB_MSB</b> <b>ACL端口寄存器14(0x10)</b> 寄存器110(0x6E) bit[7:5] = 010(用于ACL),寄存器110 bit[3:0] = 0xn(用于端口1、2、3和4)。 寄存器111(0x6F) bit[7:0] = 偏移量0x10(用于访问间接字节寄存器0xA0)。 位置:(010 ACL) -> {0xn, 偏移量} -> 0xA0保存数据。				
7 - 6	保留	—	RO	00

**表4-22: 14字节ACL规则的临时存储 (续)**

地址	名称	说明	模式	默认值
5 - 0	BYTE_ENB [13:8]	<p>ACL表中的字节使能；每条目14字节</p> <p>1 = 选择字节进行读/写 0 = 未选择字节</p> <p>BYTE_ENB[13:0]的bit[0]用于ACL表条目中的字节地址0x0D, BYTE_ENB[13:0]的bit[1]用于ACL表条目中的字节地址0x0C等。 BYTE_ENB[13:0]的bit[13]用于ACL表条目中的字节地址0x00。</p>	R/W	0
<p><b>Port_ACL_BYTE_ENB_LSB</b>  <b>ACL端口寄存器15 (0x11)</b>            寄存器110 (0x6E) bit[7:5] = 010 (用于ACL)，寄存器110 bit[3:0] = 0xn (用于端口1、2、3、4和5)。            寄存器111 (0x6F) bit[7:0] = 偏移量0x11 (用于访问间接字节寄存器0xA0)。            位置: (010 ACL) -&gt; {0xn, 偏移量} -&gt; 0xA0保存数据。</p>				
7 - 0	BYTE_ENB [7:0]	<p>ACL表中的字节使能；每条目14字节</p> <p>1 = 选择字节进行读/写 0 = 未选择字节</p> <p>BYTE_ENB[13:0]的bit[0]用于ACL表条目中的字节地址0x0D, BYTE_ENB[13:0]的bit[1]用于ACL表条目中的字节地址0x0C等。 BYTE_ENB[13:0]的bit[13]用于ACL表条目中的字节地址0x00。</p>	R/W	0x00

**表4-23: ACL读/写控制**

地址	名称	说明	模式	默认值
<p><b>Port_ACL_ACCESS_CONTROL1</b>  <b>ACL端口寄存器16 (0x12)</b>            寄存器110 (0x6E) bit[7:5] = 010 (用于ACL)，寄存器110 bit[3:0] = 0xn (用于端口1、2、3、4和5)。            寄存器111 (0x6F) bit[7:0] = 偏移量0x12 (用于访问间接字节寄存器0xA0)。            位置: (010 ACL) -&gt; {0xn, 偏移量} -&gt; 0xA0保存数据。</p>				
7	保留	—	RO	0
6	WRITE_STATUS	<p>写操作状态</p> <p>1 = 已完成写操作 0 = 正在进行写操作</p>	RO	1
5	READ_STATUS	<p>读操作状态</p> <p>1 = 已完成读操作 0 = 正在进行读操作</p>	RO	1
4	WRITE_READ	<p>请求类型</p> <p>1 = 写 0 = 读</p>	R/W	0

# KSZ8795CLX

表4-23: ACL读/写控制 (续)

地址	名称	说明	模式	默认值
3 - 0	ACL_ENTRY_ADDRESS	ACL 条目地址 0000 = 条目 0。 0001 = 条目 1。 ..... 1111 = 条目 15。	R/W	0000
<b>Port_ACL_ACCESS_CONTROL2</b> <b>ACL 端口寄存器 17 (0x13)</b> 寄存器 110 (0x6E) bit[7:5] = 010 (用于 ACL), 寄存器 110 bit[3:0] = 0xn (用于端口 1、2、3、4 和 5)。 寄存器 111 (0x6F) bit[7:0] = 偏移量 0x13 (用于访问间接字节寄存器 0xA0)。 位置: (010 ACL) -> {0xn, 偏移量} -> 0xA0 保存数据。				
7 - 1	保留	—	RO	0000000
0	强制 DLR 未命中	1 = DLR 过滤使用单个 ACL 条目。与 ACL 条目匹配的 DLR 数据包将被视为未命中 0 = DLR 过滤使用多个 ACL 条目。与 DLR 数据包的规则集匹配的 DLR 数据包将被视为命中。 <b>注:</b> DLR 的全称为器件级冗余。	R/W	0

ACL 寄存器可通过以下读/写示例进行编程:

示例:

读操作

1. 设置字节使能寄存器以从 ACL 表条目的 0x00 至 0x0d 中选择 ACL 字的所有字节的相关步骤:

使用间接访问控制寄存器选择要读取的寄存器。要读取端口 1 的第一个条目 Entry0:

将 0x41 写入寄存器 110 (0x6E) // 选择 ACL 并写入端口 1 (对于端口 2、3、4 和 5, 分别写入 0x42、0x43、0x44 和 0x45)

将 0x10 写入寄存器 111 (0x6F) // 在 ACL 端口寄存器 14 (字节使能 MSB 寄存器) 地址中触发对端口 1 的写操作。

针对字节使能字的 MSB 将 0x3F 写入间接字节寄存器 160 (0xA0)。

将 0x41 写入寄存器 110 (0x6E) // 选择写入端口 1。

将 0x11 写入寄存器 111 (0x6F) // 在 ACL 端口寄存器 15 (字节使能 LSB 寄存器) 地址中触发对端口 1 的写操作。(上述 2 个写操作都可能属于突发操作)。

针对字节使能字的 LSB 将 0xFF 写入间接字节寄存器 160 (0xA0)。

将 0x41 写入寄存器 110 (0x6E) // 选择 ACL 和对端口 1 的写操作。

将 0x12 写入寄存器 111 (0x6F) // 将 ACL 读/写控制寄存器地址 0x12 写入寄存器 111 中的间接地址, 以在 ACL 端口寄存器 16 (ACL 访问控制寄存器) 中触发对端口 1 的读操作, 从而读取条目 0。

将 0x00 写入间接字节寄存器 160 (0xA0) // ACL 端口寄存器 16 (0x12) bit[4] = 0 用于读取 ACL, bit[3:0] = 0x0 用于读取条目 0。

2. 设置 ACL 控制寄存器以读取 ACL 条目字 0 的相关步骤:

将 0x51 写入寄存器 110 (0x6E) // 选择 ACL 并读取端口 1 (对于端口 2、3、4 和 5, 分别写入 0x52、0x53、0x54 和 0x55)。

将 0x12 写入寄存器 111 (0x6F) // 在 ACL 端口寄存器 16 (ACL 访问控制 1) 中触发对端口 1 的读操作。

读取间接字节寄存器 160 (0xA0) 以获取数据 (如果 bit[5] 置 1, 则读操作会在 ACL 端口寄存器 16 [0x12] 中完成, 然后会转至下一步。否则, 将重复上述查询步骤)。

将 0x51 写入寄存器 110 (0x6E) // 选择读取端口 1。



将0x00写入寄存器111 (0x6F) //由端口1 ACL访问寄存器0 (0x00) 基于字节使能寄存器设置触发读/突发读操作。对间接字节寄存器160 (0xA0) 进行读/突发读操作//以在每次字节地址写入寄存器111 (0x6F) 后, 获取ACL条目字0的数据、将0x00写入0x0D间接地址并读取寄存器160 (0xA0)。

## 写操作

1. 设置字节使能寄存器以在ACL字中选择奇地址字节的相关步骤:

使用间接访问控制寄存器选择要写入的寄存器。要写入端口5第15个条目的偶字节数:

将0x55写入寄存器110 (0x6E) //选择ACL并读取端口5。

将0x12写入寄存器111 (0x6F) //针对端口5 ACL访问控制寄存器读取触发读操作。

读取间接字节寄存器160 (0xA0) 以获取数据 (如果bit[6]置1, 则会完成之前的写操作, 并转至下一步。否则, 将重复上述查询步骤)。

将0x45写入寄存器110 (0x6E) //选择ACL并写入端口5。

将0x00写入寄存器111 (0x6F) //为端口5 ACL端口寄存器0设置偏移地址。

针对ACL端口寄存器0、1、2、...、13从0x00至0x0D对间接字节寄存器160 (0xA0) 进行写/突发写操作 (将端口5 ACL访问寄存器0、1、...、13的偶字节写入/突发写入保持缓冲区)。

将0x45写入寄存器110 (0x6E) //选择ACL并写入端口5。

将0x10写入寄存器111 (0x6F) //在ACL端口寄存器14 (字节使能MSB寄存器) 中触发对端口5的写操作。

针对字节使能字的MSB将0x15写入间接字节寄存器160 (0xA0), 以使能奇字节地址0x01、0x03和0x05。

2. 设置ACL控制寄存器以从保持缓冲区写入ACL条目字15的相关步骤:

将0x45写入寄存器110 (0x6E) //选择写入端口5。

将0x11写入寄存器111 (0x6F) //在ACL端口寄存器15 (字节使能LSB寄存器) 中触发端口5的写操作。

针对字节使能字的LSB将0x55写入间接字节寄存器160 (0xA0), 以使能奇字节地址0x07、0x09、0x0B和0x0D。

将0x45写入寄存器110 (0x6E) //选择写入端口5。

将0x12写入寄存器111 (0x6F) //将端口ACL访问控制寄存器地址 (0x12) 写入间接地址寄存器111, 以在ACL端口寄存器16中触发对端口5的写操作, 从而写入条目15的字节1、3、5、...、13。

将0x1F写入间接字节寄存器160 (0xA0) //以在ACL端口寄存器16 (0x12) 中触发对第15个条目的写操作 (bit4 = 1用于写入ACL, bit[3:0] = 0xF用于写入条目15)。

上述示例的位分配假设在ACL端口寄存器1 (0x01) 中采用MODE = 01的第2层规则, 对于MODE = 10和11的情况, 请参见ACL格式。

## 4.9 EEE 间接寄存器

EEE功能仅适用于所有铜端口。EEE寄存器基于全局和各个端口提供。这些寄存器通过间接存储器访问进行读/写, 如下表所示: LPI的全称为低功耗空闲。

表4-24: EEE全局寄存器

地址	名称	说明	模式	默认值
<b>EEE全局寄存器0</b>				
全局EEE QM缓冲区控制寄存器				
寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0x0 (用于间接全局寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。				
偏移量: 0x30 (bit[15:8]) 和 0x31 (bit[7:0])				
位置: (001 EEE) -> {0x0, 偏移量} -> 0xA0 保存数据。				
15 - 8	保留	—	RO	0x40

# KSZ8795CLX

表4-24: EEE全局寄存器 (续)

地址	名称	说明	模式	默认值
7	LPI 由输入通信 使能终止	1 = 如果检测到输入通信, 则会停止LPI请求。 0 = LPI请求不会被输入通信终止。	R/W	0
6 - 0	保留	—	RO	0x10
<b>EEE全局寄存器1</b>				
全局空TXQ到LPI等待时间控制寄存器 寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0x0 (用于间接全局寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x32 (bit[15:8]) 和 0x33 (bit[7:0]) 位置: (001 EEE) -> {0x0, 偏移量} -> 0xA0 保存数据。				
15 - 0	空TXQ到LPI 等待时间	该寄存器用于指定一段时间, 当TXQ为空的时间超过该配置时间后会生成LPI请求。这只有在使能EEE 100BT后才有效。该设置将适用于所有端口。单位值为1.3 ms。默认值为1.3s (取值范围为1.3 ms至86s)	R/W	0x10
<b>EEE全局寄存器2</b>				
全局EEE PCS诊断寄存器 寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0x0 (用于间接全局寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x34 (bit[15:8]) 和 0x35 (bit[7:0]) 位置: (001 EEE) -> {0x0, 偏移量} -> 0xA0 保存数据。				
15 - 12	保留	—	RO	0x6
11 - 8	保留	—	RO	0x8
7 - 4	保留	—	RO	0x0
3	端口4下一页 使能	1 = 自动协商期间使能下一页交换。 0 = 自动协商期间跳过下一页交换。	R/W	1
2	端口3下一页 使能	1 = 自动协商期间使能下一页交换。 0 = 自动协商期间跳过下一页交换。	R/W	1
1	端口2下一页 使能	1 = 自动协商期间使能下一页交换。 0 = 自动协商期间跳过下一页交换。	R/W	1
0	端口1下一页 使能	1 = 自动协商期间使能下一页交换。 0 = 自动协商期间跳过下一页交换。	R/W	1
<b>EEE全局寄存器3</b>				
返回至空闲控制寄存器之前的全局EEE最小LPI周期 寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0x0 (用于间接全局寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x36 (bit[15:8]) 和 0x37 (bit[7:0]) 位置: (001 EEE) -> {0x0, 偏移量} -> 0xA0 保存数据。				
15 - 0	保留	—	RO	0x0000

**表4-24: EEE全局寄存器 (续)**

地址	名称	说明	模式	默认值
<b>EEE全局寄存器4</b> 全局EEE唤醒错误阈值控制寄存器 寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0x0 (用于间接全局寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x38 (bit[15:8]) 和 0x39 (bit[7:0]) 位置: (001 EEE) -> {0x0, 偏移量} -> 0xA0 保存数据。				
15 - 0	EEE唤醒阈值	该值指定允许的最长PHY唤醒时间。 如果唤醒时间超过该值, 则EEE唤醒错误计数会递增。 注: 这是EEE标准, 不得更改。	RO	0x0201
<b>EEE全局寄存器5</b> 全局EEE PCS诊断控制寄存器 寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0x0 (用于间接全局寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x3A (bit[15:8]) 和 0x3B (bit[7:0]) 位置: (001 EEE) -> {0x0, 偏移量} -> 0xA0 保存数据。				
15 - 0	保留	—	RO	0x0001

**表4-25: EEE端口寄存器**

地址	名称	说明	模式	默认值
<b>EEE端口寄存器0</b> 端口自动协商扩展状态寄存器 寄存器110 (0x6E) bit[7:5] = 001 (用于EEE), 寄存器110 bit[3:0] = 0xn (其中n = 1-4, 用于间接端口寄存器), 寄存器111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器0xA0)。 偏移量: 0x0C (bit[15:8]) 和 0x0D (bit[7:0]) 位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。				
15 - 7	保留	—	RO	9h000
6	接收下一页位置功能	1 = 接收的下一页存储位置由bit[6:5]指定 0 = 接收的下一页存储位置不由bit[6:5]指定	RO	1
5	接收的下一页存储位置	1 = 链路伙伴下一页存储在MIIM寄存器8h中 (附加下一页) 0 = 链路伙伴下一页存储在MIIM寄存器5h中	RO	1
4	并行检测故障	1 = 通过并行检测功能检测到了故障。 0 = 未通过并行检测功能检测到故障。  此位在读取后清零。	R/LH	0
3	链路伙伴下一页功能	1 = 链路伙伴具备下一页功能 0 = 链路伙伴不具备下一页功能	RO	0
2	下一页功能	1 = 本地设备具备下一页功能 0 = 本地设备不具备下一页功能	RO	1
1	页接收	1 = 已接收到新页 0 = 尚未接收到新页	R/LH	0

# KSZ8795CLX

表4-25: EEE 端口寄存器 (续)

地址	名称	说明	模式	默认值
0	链路伙伴自动协商功能	1 = 链路伙伴具备自动协商功能 0 = 链路伙伴不具备自动协商功能	RO	0
<b>EEE 端口寄存器 1</b> 端口自动协商下一页发送寄存器 寄存器 110 (0x6E) bit[7:5] = 001 (用于 EEE), 寄存器 110 bit[3:0] = 0xn (其中 n = 1-4, 用于间接端口寄存器), 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。 偏移量: 0x0E (bit[15:8]) 和 0x0F (bit[7:0]) 位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。 如果 EEE 端口寄存器 5 bit[7] = 1, 默认为自动执行 EEE 功能, 则无需设置该寄存器。				
15	下一页	下一页 (Next Page, NP) 供下一页功能用于指示此页是否为要发送的最后一个下一页。NP 应按如下进行设置:  1 = 后续还有要发送的下一页。 0 = 最后一页。	R/W	0
14	保留	—	RO	0
13	消息页	消息页 (Message Page, MP) 供下一页功能用于区分消息页与无格式页。MP 应按如下进行设置:  1 = 消息页 0 = 无格式页	R/W	1
12	应答 2	应答 2 (Ack2) 供下一页功能用于指示器件能否对消息做出应答。Ack2 应按如下进行设置:  1 = 将对消息做出应答。 0 = 不能对消息做出应答。	R/W	0
11	翻转	翻转 (T) 供仲裁功能用于确保在下一页交换期间与链路伙伴同步。该位应始终采用之前交换的链路码字中翻转位的相反值。发送的第一个下一页中翻转位的初始值为基本链路码字中 bit[11] 的相反值, 因此可假定为逻辑值 1 或 0。翻转位应按如下进行设置:  1 = 发送的链路码字的前一个值等于逻辑 0。 0 = 发送的链路码字的前一个值等于逻辑 1。	RO	0
10 - 0	消息/无格式代码字段	消息/无格式代码字段 bit[10:0]	R/W	1

**表4-25: EEE 端口寄存器 (续)**

地址	名称	说明	模式	默认值
<b>EEE 端口寄存器 2</b> 端口自动协商链路伙伴下一页接收寄存器 寄存器 110 (0x6E) bit[7:5] = 001 (用于 EEE), 寄存器 110 bit[3:0] = 0xn (其中 n = 1-4, 用于间接端口寄存器), 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。 偏移量: 0x10 (bit[15:8]) 和 0x11 (bit[7:0]) 位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。				
15	下一页	下一页 (NP) 供下一页功能用于指示此页是否是要发送的最后一个下一页。NP 应按如下进行设置:  1 = 后续还有要发送的下一页。 0 = 最后一页。	RO	0
14	应答	应答 (Ack) 供自动协商功能用于指示器件已成功接收其链路伙伴的链路码字。无论选择器字段的值或链路码字编码如何, 应答位均在 bit D14 中进行编码。如果没有要发送的下一页信息, 则在连续接收到至少三个一致的 FLP 突发后, 该位应在链路码字中被置为逻辑 1 (忽略应答位值)。	RO	0
13	消息页	消息页 (MP) 供下一页功能用于区分消息页与无格式页。MP 应按如下进行设置:  1 = 消息页 0 = 无格式页	RO	0
12	应答 2	应答 2 (Ack2) 供下一页功能用于指示器件能否对消息做出应答。Ack2 应按如下进行设置:  1 = 将对消息做出应答。 0 = 不能对消息做出应答。	RO	0
11	翻转	翻转 (T) 供仲裁功能用于确保在下一页交换期间与链路伙伴同步。该位应始终采用之前交换的链路码字中翻转位的相反值。发送的第一个下一页中翻转位的初始值为基本链路码字中 bit[11] 的相反值, 因此可假定为逻辑值 1 或 0。翻转位应按如下进行设置:  1 = 发送的链路码字的前一个值等于逻辑 0。 0 = 发送的链路码字的前一个值等于逻辑 1。	RO	0
10 - 0	消息/无格式代码字段	消息/无格式代码字段 bit[10:0]	RO	0

# KSZ8795CLX

表4-25: EEE 端口寄存器 (续)

地址	名称	说明	模式	默认值
<b>EEE 端口寄存器 3</b>				
链路伙伴 EEE 功能状态和本地设备 EEE 功能通告寄存器				
寄存器 110 (0x6E) bit[7:5] = 001 (用于 EEE), 寄存器 110 bit[3:0] = 0xn (其中 n = 1-4, 用于间接端口寄存器), 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。				
偏移量: 0x28 (bit[15:8]) 和 0x29 (bit[7:0])				
位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。				
15	保留	—	RO	0
14	LP 10GBASE- KR EEE	1 = 支持 10GBASE-KR EEE 0 = 不支持 10GBASE-KR EEE  注: LP = 链路伙伴	RO	0
13	LP 10GBASE- KX4 EEE	1 = 支持 10GBASE-KX4 EEE 0 = 不支持 10GBASE-KX4 EEE	RO	0
12	LP 1000BASE- KX EEE	1 = 支持 1000BASE-KX EEE 0 = 不支持 1000BASE-KX EEE	RO	0
11	LP 10GBASE-T EEE	1 = 支持 10GBASE-T EEE 0 = 不支持 10GBASE-T EEE	RO	0
10	LP 1000BASE-T EEE	1 = 支持 1000BASE-T EEE 0 = 不支持 1000BASE-T EEE	RO	0
9	LP 100BASE-TX EEE	1 = 支持 100BASE-TX EEE 0 = 不支持 100BASE-TX EEE	RO	0
8 - 2	保留	—	RO	7h'0
1	本地 100BASE-TX EEE	1 = 支持 100BASE-TX EEE 0 = 不支持 100BASE-TX EEE  注: 这是用于使本地端口支持 EEE 功能	R/W	1
0	保留	—	RO	0
<b>EEE 端口寄存器 4</b>				
端口 EEE 唤醒错误计数寄存器				
寄存器 110 (0x6E) bit[7:5] = 001 (用于 EEE), 寄存器 110 bit[3:0] = 0xn (其中 n = 1-4, 用于间接端口寄存器), 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。				
偏移量: 0x2A (bit[15:8]) 和 0x2B (bit[7:0])				
位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。				
15 - 0	EEE 唤醒错 误计数器	如果从 LPI 到空闲状态的唤醒时间超过 EEE 全局寄存器 4 中指定的唤醒错误阈值时间, 则该计数会递增 1 次。唤醒错误阈值时间的默认值为 20.5 μs。该寄存器会在读取时清零。	RO	0x0000

**表4-25: EEE 端口寄存器 (续)**

地址	名称	说明	模式	默认值
<b>EEE 端口寄存器 5</b> 端口 EEE 控制寄存器 寄存器 110 (0x6E) bit[7:5] = 001 (用于 EEE), 寄存器 110 bit[3:0] = 0xn (其中 n = 1-4, 用于间接端口寄存器), 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。 偏移量: 0x2C (bit[15:8]) 和 0x2D (bit[7:0]) 位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。				
15	10BT EEE 禁止	1 = 禁止 10BT EEE 模式 0 = 使能 10BT EEE 模式 注: 10BT EEE 模式仅通过降低信号幅值来实现节能。	R/W	1
14 - 8	保留	—	RO	7h'0
7	基于硬件的 EEE NP 自动协商使能	1 = 硬件会通过下一页交换自动与链路伙伴执行 EEE 功能交换。EEE 100BT 使能 (该寄存器的 bit[0])。如果 EEE 功能匹配, 则会由硬件置 1。 0 = 禁止基于硬件的 EEE 功能交换。EEE 功能交换由软件实现。	R/W	1
6	硬件 100BT EEE 使能状态	1 = 通过基于硬件的 NP 交换使能 100BT EEE 0 = 禁止 100BT EEE	R	0
5	TX LPI 接收	1 = 指示自上次读取寄存器后, 发送 PCS 已接收到一次或多次低功耗空闲信号。 0 = 指示 PCS 尚未接收到低功耗空闲信号。 此位在读取后清零。	R/RC	0
4	TX LPI 指示	1 = 指示发送 PCS 当前正在接收低功耗空闲信号。 0 = 指示 PCS 当前未接收低功耗空闲信号。	R	0
3	RX LPI 接收	1 = 指示自上次读取寄存器后, 接收 PCS 已接收到一次或多次低功耗空闲信号。 0 = 指示 PCS 尚未接收到低功耗空闲信号。 此位在读取后清零。	R/RC	0
2	RX LPI 指示	1 = 指示接收 PCS 当前正在接收低功耗空闲信号。 0 = 指示 PCS 当前未接收低功耗空闲信号。	R	0
1	EEE 软件模式使能	1 = 通过软件将该寄存器的 bit[0] 置 1 来使能 EEE。 0 = 通过硬件自动协商来使能 EEE。	R/W	0
0	EEE SW 100BT 使能	1 = 使能 EEE 100BT 0 = 禁止 EEE 100BT  注: 如果启用基于硬件的 EEE 下一页自动协商使能, 则该位可能由软件或硬件置 1。	R/W	0

# KSZ8795CLX

表4-25: EEE 端口寄存器 (续)

地址	名称	说明	模式	默认值
<b>EEE 端口寄存器 6</b>				
端口 EEE LPI 恢复时间寄存器				
寄存器 110 (0x6E) bit[7:5] = 001 (用于 EEE), 寄存器 110 bit[3:0] = 0xn (其中 n = 1-4, 用于间接端口寄存器), 寄存器 111 (0x6F) bit[7:0] = 偏移量 (用于访问间接字节寄存器 0xA0)。				
偏移量: 0x2E (bit[15:8]) 和 0x2F (bit[7:0])				
位置: (001 EEE) -> {0xn, 偏移量} -> 0xA0 保存数据。				
15 - 8	保留	—	RO	1
7 - 0	LPI 恢复计数器	该寄存器指定 MAC 设备在能够开始发出数据包之前必须等待的时间。该值应是本地设备与远程设备二者的 LPI 恢复时间的较大值。单位值为 640 ns。默认值约为 25 μs = 39 (0x27) × 640 ns 注: 对于要快速从 EEE LPI 模式发出的数据包, 如果 PHY 恢复时间小于标准值 20.5 μs, 则可对该值进行调整。	R/W	0x27

编程示例:

读操作

1. 使用间接访问控制寄存器选择要读取的寄存器以及读取 EEE 全局寄存器 0 (全局 EEE QM 缓冲区控制寄存器)。
2. 将 0x30 写入寄存器 110 (0x6E) // EEE 选择和读操作, 端口号的 4 个 MSB = 0 用于全局寄存器。
3. 将 0x30 写入间接寄存器 111 (0x6F) // 触发读操作, 并准备好读取 EEE 全局寄存器 0 bit[15:8]。
4. 读取间接字节寄存器 160 (0xA0) // 获取 EEE 全局寄存器 0 的 bit[15:8] 值。

写操作

1. 将 0x20 写入寄存器 110 (0x6E) // EEE 选择和写操作, 端口号的 4 个 MSB = 0 用于全局寄存器。
2. 将 0x31 写入寄存器 111 (0x6F) // 选择偏移地址, 准备好写入 EEE 全局寄存器 0 bit[7:0]。
3. 将新值写入间接字节寄存器 160 (0xA0) bit[7:0]。

## 4.10 管理信息库 (MIB) 计数器

MIB 计数器基于各个端口提供。这些计数器通过间接存储器访问进行读取, 如表 4-26 所示。

表 4-26: 端口 MIB 计数器间接存储器偏移量

偏移量	计数器名称	说明
0x0	RxHiPriorityByte	Rx 高优先级八位字节计数 (包括不良数据包)。
0x1	RxUndersizePkt	Rx 过小数据包, CRC 良好。
0x2	RxFragments	Rx 碎片数据包, CRC 不良, 包含符号错误或对齐错误。
0x3	RxOversize	Rx 过大数据包, CRC 良好 (最大值: 1536 或 1522 个字节)。
0x4	RxJabbers	长度大于 1522 个字节的 Rx 数据包 (包含 CRC 错误、对齐错误或符号错误, 具体取决于最大数据包大小设置) 或仅为长度大于 1916 个字节的 Rx 数据包。
0x5	RxSymbolError	包含无效数据符号以及合法前导和数据包大小的 Rx 数据包。
0x6	RxCRCError	字节数为介于 64 和 1522 之间的整数且 CRC 不良的 Rx 数据包 (上限取决于最大数据包大小设置)。
0x7	RxAlignmentError	字节数为介于 64 和 1522 之间的非整数且 CRC 不良的 Rx 数据包 (上限取决于最大数据包大小设置)。



**表4-26: 端口MIB计数器间接存储器偏移量 (续)**

偏移量	计数器名称	说明
0x8	RxControl8808Pkts	EtherType (以太网类型) 字段中为88-08h的端口接收到的MAC控制帧数。
0x9	RxPausePkts	端口接收到的暂停帧数。暂停帧由EtherType (88-08h)、DA、控制操作码 (00-01)、数据长度 (最少64个字节) 以及有效CRC限定。
0xA	RxBroadcast	Rx良好广播数据包 (不包括错误的广播数据包或有效的多播数据包)。
0xB	RxMulticast	Rx良好多播数据包 (不包括MAC控制帧、错误的多播数据包或有效的广播数据包)。
0xC	RxUnicast	Rx良好单播数据包。
0xD	Rx64Octets	总长度为64个八位字节的全部Rx数据包 (包括不良数据包)。
0xE	Rx65to127Octets	总长度介于65和127个八位字节之间的全部Rx数据包 (包括不良数据包)。
0xF	Rx128to255Octets	总长度介于128和255个八位字节之间的全部Rx数据包 (包括不良数据包)。
0x10	Rx256to511Octets	总长度介于256和511个八位字节之间的全部Rx数据包 (包括不良数据包)。
0x11	Rx512to1023Octets	总长度介于512和1023个八位字节之间的全部Rx数据包 (包括不良数据包)。
0x12	Rx1024to1522Octets	总长度介于1024和1522个八位字节之间的全部Rx数据包 (包括不良数据包)。
0x13	Rx1523to2000Octets	总长度介于1523和2000个八位字节之间的全部Rx数据包 (包括不良数据包)。
0x14	Rx2001toMax-1Octets	总长度介于2001和 (最大值减1) 个八位字节之间的全部Rx数据包 (包括不良数据包, 上限取决于最大数据包大小减1)。
0x15	TxHiPriorityByte	Tx高优先级良好八位字节计数 (包括暂停数据包)。
0x16	TxLateCollision	在512个位时间的数据包发送后检测到的冲突次数。
0x17	TxPausePkts	端口发送的暂停帧数。
0x18	TxBroadcastPkts	Tx良好广播数据包 (不包括错误的广播或有效的多播数据包)。
0x19	TxMulticastPkts	Tx良好多播数据包 (不包括错误的多播数据包或有效的广播数据包)。
0x1A	TxUnicastPkts	Tx良好单播数据包。
0x1B	TxDeferred	端口第一次尝试发送时因介质繁忙而延时的发送数据包数。
0x1C	TxTotalCollision	Tx总冲突, 仅限半双工。
0x1D	TxExcessiveCollision	因冲突过多而导致Tx失败的帧计数。
0x1E	TxSingleCollision	仅由一个冲突导致禁止Tx的端口的成功Tx帧计数。
0x1F	TxMultipleCollision	由多个冲突导致禁止Tx的端口的成功Tx帧计数。

# KSZ8795CLX

表4-27: 基于端口的MIB计数器格式

地址	名称	说明	模式	默认值
对于端口2, 基址为0x20, 相同的偏移量定义 (0x20-0x3f) 对于端口3, 基址为0x40, 相同的偏移量定义 (0x40-0x5f) 对于端口4, 基址为0x60, 相同的偏移量定义 (0x60-0x7f) 对于端口5, 基址为0x80, 相同的偏移量定义 (0x80-0x9f)				
38	溢出	1 = 计数器溢出。 0 = 无计数器溢出。	RO	0
37	计数有效	1 = 计数器值有效。 0 = 计数器值无效。	RO	0
36 - 30	保留	—	RO	全0
29 - 0	计数器值	计数器值	RO	0

表4-28: 所有端口丢弃的数据包MIB计数器

偏移量	计数器名称	说明
0x100	端口1 Rx总字节数	端口1 Rx总八位字节计数, 包括不良数据包。
0x101	端口1 Tx总字节数	端口1 Tx总良好八位字节计数, 包括暂停数据包。
0x102	端口1 Rx丢弃数据包	因缺少资源而丢弃的端口1 Rx数据包。
0x103	端口1 Tx丢弃数据包	因缺少资源而丢弃的端口1 Tx数据包。
0x104	端口2 Rx总字节数	端口2 Rx总八位字节计数, 包括不良数据包。
0x105	端口2 Tx总字节数	端口2 Tx总良好八位字节计数, 包括暂停数据包。
0x106	端口2 Rx丢弃数据包	因缺少资源而丢弃的端口2 Rx数据包。
0x107	端口2 Tx丢弃数据包	因缺少资源而丢弃的端口2 Tx数据包。
0x108	端口3 Rx总字节数	端口3 Rx总八位字节计数, 包括不良数据包。
0x109	端口3 Tx总字节数	端口3 Tx总良好八位字节计数, 包括暂停数据包。
0x10A	端口3 Rx丢弃数据包	因缺少资源而丢弃的端口3 Rx数据包。
0x10B	端口3 Tx丢弃数据包	因缺少资源而丢弃的端口3 Tx数据包。
0x10C	端口4 Rx总字节数	端口4 Rx总八位字节计数, 包括不良数据包。
0x10D	端口4 Tx总字节数	端口4 Tx总良好八位字节计数, 包括暂停数据包。
0x10E	端口4 Rx丢弃数据包	因缺少资源而丢弃的端口4 Rx数据包。
0x10F	端口4 Tx丢弃数据包	因缺少资源而丢弃的端口4 Tx数据包。
0x110	端口5 Rx总字节数	端口5 Rx总八位字节计数, 包括不良数据包。
0x111	端口5 Tx总字节数	端口5 Tx总良好八位字节计数, 包括暂停数据包。
0x112	端口5 Rx丢弃数据包	因缺少资源而丢弃的端口5 Rx数据包。
0x113	端口5 Tx丢弃数据包	因缺少资源而丢弃的端口5 Tx数据包。

表4-29: 基于端口的RX/TX总字节MIB计数器(表4-28中)的格式

地址	名称	说明	模式	默认值
38	溢出	1 = 计数器溢出。 0 = 无计数器溢出。	RO	0
37	计数有效	1 = 计数器值有效。 0 = 计数器值无效。	RO	0
36	保留	—	RO	0
35 - 0	计数器值	计数器值	RO	0

**表4-30: 所有丢弃的数据包MIB计数器（表4-28中）的格式**

地址	名称	说明	模式	默认值
38	溢出	1 = 计数器溢出。 0 = 无计数器溢出。	RO	0
37	计数有效	1 = 计数器值有效。 0 = 计数器值无效。	RO	0
36 - 16	保留	—	RO	全0
15 - 0	计数器值	计数器值	RO	0

请注意，所有基于端口的MIB计数器均会在读取时清零。

KSZ8795CLX还具备统计控制功能，可通过全局寄存器8控制MIB，从而按端口刷新计数器或冻结计数器。

KSZ8795CLX总共按端口提供了36个MIB计数器。这些计数器用于监视端口活动，从而实现网络管理和维护。这些MIB计数器通过间接存储器访问进行读取，可参考以下示例。

1. MIB计数器读操作（读取端口1 Rx64Octets计数器）

将0x1c写入寄存器110（读取所选MIB计数器）

将0xd写入寄存器111（触发读操作）

则：

读寄存器116（计数器值[39:32]）

//如果bit[38] = 1，则发生了计数器溢出

读寄存器117（计数器值[31:24]）

读寄存器118（计数器值[23:16]）

读寄存器119（计数器值[15:8]）

读寄存器120（计数器值[7:0]）

2. MIB计数器读操作（读取端口2 Rx64Octets计数器）

将0x1c写入寄存器110（读取所选MIB计数器）

将0x2d写入寄存器111（触发读操作）

则：

读寄存器116（计数器值[39:32]）

//如果bit[38] = 1，则发生了计数器溢出

读寄存器117（计数器值[31:24]）

读寄存器118（计数器值[23:16]）

读寄存器119（计数器值[15:8]）

读寄存器120（计数器值[7:0]）

3. MIB计数器读操作（读取端口1 TX丢弃数据包）

将0x1d写入寄存器110

将0x03写入寄存器111

则：

读寄存器116（计数器值[39:32]）

//如果bit[38] = 1，则发生了计数器溢出

读寄存器119（计数器值[15:8]）

读寄存器120（计数器值[7:0]）

# KSZ8795CLX

要读出所有计数器，SPI总线上的最佳性能为  $(160+3) \times 8 \times 20 = 26 \mu\text{s}$ ，其中160为寄存器数，3为开销，8为每次访问的时钟数（频率为50 MHz）。在最繁重的情况下，字节计数器会在2分钟内溢出。建议软件至少每30秒读取一次所有计数器。所有端口MIB计数器均设计为“读取时清零”的形式。

## 4.11 MIIM 寄存器

本节中定义的所有寄存器还可通过SPI接口进行访问。

请注意，MIIM和SPI采用不同的映射机制。对于IEEE中定义的“PHYAD”，针对端口1分配为“0x1”；针对端口2分配为“0x2”；针对端口3分配为“0x3”；针对端口4分配为“0x4”。支持的“REGAD”为0x0-0x5（0h-5h）、0x1D（1dh）和0x1F（1fh）。

表4-31: MIIM 寄存器

地址	名称	说明	模式	默认值
<b>寄存器0h: 基本控制</b>				
15	软复位	1 = PHY软复位。 0 = 正常工作。	R/W (SC)	0
14	环回	1 = 执行MAC环回，环回路径如下： 假定环回位于端口1 MAC处，端口2为监视端口。 端口1 MAC环回（端口1寄存器0 bit[14] = 1） 开始：RXP2/RXM2（端口2）。也可从端口3、4和5开始 环回：端口1 MAC的MAC/PHY接口 结束：TXP2/TXM2（端口2）。也可分别在端口3、4和5结束  将寄存器0中地址0x3、0x4和0x5的bit[14]置1会分别对端口3、4和5执行MAC环回。 0 = 正常工作。	R/W	0
13	强制100	1 = 100 Mbps。 0 = 10 Mbps。	R/W	1
12	AN使能	1 = 使能自动协商。 0 = 禁止自动协商。	R/W	1
11	掉电	1 = 掉电。 0 = 正常工作。	R/W	0
10	PHY隔离	1 = PHY与Tx+/Tx-电气PHY隔离。 0 = 正常工作。	R/W	0
9	重新启动AN	1 = 重新启动自动协商。 0 = 正常工作。	R/W	0
8	强制全双工	1 = 全双工。 0 = 半双工。	R/W	1
7	保留	—	RO	0
6	保留	—	RO	0
5	Hp_mdix	1 = HP自动MDI/MDIX模式 0 = Microchip自动MDI/MDIX模式	R/W	1
4	强制MDI	1 = 禁止自动MDI/MDIX时进入MDI模式。 0 = 禁止自动MDI/MDIX时进入MDIX模式。	R/W	0
3	禁止自动MDI/MDI-X	1 = 禁止自动MDI/MDIX。 0 = 使能自动MDI/MDIX。	R/W	0
2	禁止远端故障	1 = 禁止远端故障检测。 0 = 正常工作。	R/W	0

**表4-31: MIIM寄存器 (续)**

地址	名称	说明	模式	默认值
1	禁止发送	1 = 禁止发送。 0 = 正常工作。	R/W	0
0	禁止LED	1 = 禁止LED。 0 = 正常工作。	R/W	0
<b>寄存器 1h: 基本状态</b>				
15	T4功能	0 = 不具备 100 BASE-T4 功能。	RO	0
14	100全双工功能	1 = 具备 100BASE-TX全双工功能。 0 = 不具备 100BASE-TX全双工功能。	RO	1
13	100半双工功能	1 = 具备 100BASE-TX半双工功能。 0 = 不具备 100BASE-TX半双工功能。	RO	1
12	10全双工功能	1 = 具备 10BASE-T全双工功能。 0 = 不具备 10BASE-T全双工功能。	RO	1
11	10半双工功能	1 = 具备 10BASE-T半双工功能。 0 = 具备 10BASE-T半双工功能。	RO	1
10 - 7	保留	—	RO	0
6	保留	—	RO	0
5	AN完成	1 = 自动协商已完成。 0 = 自动协商尚未完成。	RO	0
4	远端故障	1 = 检测到远端故障。 0 = 未检测到远端故障。	RO	0
3	AN功能	1 = 具备自动协商功能。 0 = 不具备自动协商功能。	RO	1
2	链路状态	1 = 链路连通。 0 = 链路中断。	RO	0
1	保留	—	RO	0
0	扩展功能	0 = 不具备扩展寄存器功能。	RO	0
<b>寄存器 2h: PHYID 高位</b>				
15 - 0	Phyid 高位	PHYID 高位。	RO	0x0022
<b>寄存器 3h: PHYID 低位</b>				
15 - 0	Phyid 低位	PHYID 低位。	RO	0x1550
<b>寄存器 4h: 通告能力</b>				
15	保留	—	RO	0
14	保留	—	RO	0
13	保留	—	RO	0
12 - 11	保留	—	RO	01
10	暂停	1 = 通告暂停功能。 0 = 不通告暂停功能。	R/W	1
9	保留	—	R/W	0
8	通告 100 全双工	1 = 通告 100 全双工功能。 0 = 不通告 100 全双工功能。	R/W	1
7	通告 100 半双工	1 = 通告 100 半双工功能。 0 = 不通告 100 半双工功能。	R/W	1
6	通告 10 全双工	1 = 通告 10 全双工功能。 0 = 不通告 10 全双工功能。	R/W	1
5	通告 10 半双工	1 = 通告 10 半双工功能。 0 = 不通告 10 半双工功能。	R/W	1

# KSZ8795CLX

表4-31: MIIM寄存器(续)

地址	名称	说明	模式	默认值
4 - 0	选择器字段	[00001] = IEEE 802.3	RO	00001
<b>寄存器 5h: 链路伙伴功能</b>				
15	保留	—	RO	0
14	保留	—	RO	0
13	保留	—	RO	0
12 - 11	保留	—	RO	0
10	暂停	1 = 链路伙伴具备流控制功能。 0 = 链路伙伴不具备流控制功能。	RO	0
9	保留	—	RO	0
8	通告 100 全双工	1 = 链路伙伴具备 100BT 全双工功能。 0 = 链路伙伴不具备 100BT 全双工功能。	RO	0
7	通告 100 半双工	1 = 链路伙伴具备 100BT 半双工功能。 0 = 链路伙伴不具备 100BT 半双工功能。	RO	0
6	通告 10 全双工	1 = 链路伙伴具备 10BT 全双工功能。 0 = 链路伙伴不具备 10BT 全双工功能。	RO	0
5	通告 10 半双工	1 = 链路伙伴具备 10BT 半双工功能。 0 = 链路伙伴不具备 10BT 半双工功能。	RO	0
4 - 0	保留	—	RO	00001
<b>寄存器 1dh: LinkMD 控制/状态</b>				
15	CDT_Enable	1 = 使能电缆诊断。完成 CDT 测试后, 该位会自清零。 0 = 指示电缆诊断测试 (如果已使能) 已完成, 可读取状态信息。	R/W (SC)	0
14 - 13	CDT_Result	00 = 正常情况 01 = 检测到电缆中存在开路情况 10 = 检测到电缆中存在短路情况 11 = 电缆诊断测试失败	RO	00
12	CDT 短于 10m	1 = 短于 10 米	RO	0
11 - 9	保留	—	RO	0
8 - 0	CDT_Fault_Count	到故障的距离, 约为 $0.4m \times CDT\_Fault\_Count[8:0]$	RO	000000000
<b>寄存器 1fh: PHY 特殊控制/状态</b>				
15 - 11	保留	—	RO	0000000000
10 - 8	端口工作模式指示	指示端口工作模式的当前状态: 000 = 保留 001 = 仍处于自动协商状态 010 = 10BASE-T 半双工 011 = 100BASE-TX 半双工 100 = 保留 101 = 10BASE-T 全双工 110 = 100BASE-TX 全双工 111 = PHY/MII 隔离	RO	001
7 - 6	保留	—	RO	00
5	Polrvs	1 = 极性是相反的 0 = 极性不是相反的	RO	0
4	MDI-X 状态	1 = MDI 0 = MDI-X	RO	0

**表4-31: MIIM寄存器 (续)**

地址	名称	说明	模式	默认值
3	Force_Ink	1 = 强制链路传递 0 = 正常工作	R/W	0
2	Pwrsave	1 = 使能节能 0 = 禁止节能	R/W	0
1	远程环回	1 = 执行远程环回，环回路径如下： 端口1（寄存器1fh中PHY ID地址0x1的bit[1] = 1） 开始：RXP1/RXM1（端口1） 环回：端口1 PHY的PMD/PMA 结束：TXP1/TXM1（端口1） 将寄存器1fh中PHY ID地址0x2、0x3、0x4和0x5的bit[1]置1，会对端口2、3、4和5执行远程环回。 0 = 正常工作。	R/W	0
0	保留	—	RO	0

# KSZ8795CLX

## 5.0 工作特性

### 5.1 绝对最大值\*

供电电压 ( $V_{DD12A}$ 和 $V_{DD12D}$ )	-0.5V至+1.8V
( $V_{DDAT}$ 和 $V_{DDIO}$ )	-0.5V至+4.0V
输入电压	-0.5V至+4.0V
输出电压	-0.5V至+4.0V
引脚温度 (焊接, 10s)	+260°C
储存温度 ( $T_S$ )	-55°C至+150°C
最高结温	+125°C
ESD 额定值	5 kV

\*超出绝对最大值可能会损坏器件。如果器件的工作条件超过上述“绝对最大值”，可能对器件造成永久性损坏。我们建议不要使器件工作在最大值甚至超过最大值的条件下。器件长时间工作在最大值条件下，其可靠性可能受到影响。

### 5.2 工作额定值\*\*

供电电压 ( $V_{DD12A}$ 和 $V_{DD12D}$ )	+1.140V至+1.260V
( $V_{DDAT}$ @ 3.3V)	+3.135V至+3.465V
( $V_{DDAT}$ @ 2.5V)	+2.375V至+2.625V
( $V_{DDIO}$ @ 3.3V)	+3.135V至+3.465V
( $V_{DDIO}$ @ 2.5V)	+2.375V至+2.625V
( $V_{DDIO}$ @ 1.8V)	+1.710V至+1.890V
环境温度 ( $T_A$ )	
商业级	0°C至+70°C
工业级	-40°C至+85°C
封装热阻 ( $\Theta_{JA}$ , 注5-1)	+55.05°C/W
封装热阻 ( $\Theta_{JC}$ , 注5-1)	+25.06°C/W

\*\* 器件无法保证在超出工作额定值范围的条件下正常工作。未使用的输入必须始终连接到适当的逻辑电平 (GND或 $V_{DD}$ )。

**注5-1** 封装中不包含散热器。结至环境热阻 ( $\Theta_{JA}$ ) 和结至外壳热阻 ( $\Theta_{JC}$ ) 是在空气流速为0 m/s的条件下得到的。

**注:** 请勿在器件未上电的情况下驱动输入信号。



## 6.0 电气特性

$V_{IN} = 1.2V/3.3V$  (典型值);  $T_A = +25^\circ C$ 。规范值仅针对已封装的产品。由于10BASE-T和100BASE-TX均使用片内终结技术和内部偏置,因此没有额外的变压器功耗。测试条件为端口5 RGMII模式下(默认)。测量结果在工作额定值下得出。

**表6-1: 电气特性**

参数	符号	最小值	典型值	最大值	单位	注
<b>100BASE-TX操作——所有端口的利用率为100%</b>						
100BASE-TX (发送器) 3.3V 模拟	$I_{DX}$	—	142	—	mA	$V_{DDAT}$
100BASE-TX 1.2V	$I_{D12}$	—	35	—		$V_{DD12A} + V_{DD12D}$
100BASE-TX (数字IO) 3.3V 数字	$I_{DDIO}$	—	15	—		$V_{DDIO}$
<b>10BASE-T操作——所有端口的利用率为100%</b>						
10BASE-T (发送器) 3.3V 模拟	$I_{DX}$	—	135	—	mA	$V_{DDAT}$
10BASE-T 1.2V	$I_{D12}$	—	30	—		$V_{DD12A} + V_{DD12D}$
10BASE-T (数字IO) 3.3V 数字	$I_{DDIO}$	—	14	—		$V_{DDIO}$
<b>自动协商模式</b>						
3.3V 模拟	$I_{DX}$	—	66	—	mA	$V_{DDAT}$
1.2V 模拟/数字	$I_{D12}$	—	35	—		$V_{DD12A} + V_{DD12D}$
3.3V 数字 I/O	$I_{DDIO}$	—	14	—		$V_{DDIO}$
<b>电源管理模式</b>						
软掉电模式 3.3V	$I_{SPDM1}$	—	0.07	—	mA	$V_{DDAT} + V_{DDIO}$
软掉电模式 1.2V	$I_{SPDM2}$	—	0.2	—		$V_{DD12A} + V_{DD12D}$
电能检测模式 (EDPD) 3.3V	$I_{EDM1}$	—	21	—		$V_{DDAT} + V_{DDIO}$
电能检测模式 (EDPD) 1.2V	$I_{EDM2}$	—	26.5	—		$V_{DD12A} + V_{DD12D}$
空闲时的100BT EEE 模式 3.3V	$I_{EEE1}$	—	22.5	—		$V_{DDAT} + V_{DDIO}$
空闲时的100BT EEE 模式 1.2V	$I_{EEE2}$	—	27	—		$V_{DD12A} + V_{DD12D}$
<b>CMOS输入</b>						
输入高电压	$V_{IH}$	2.0	—	—	V	$V_{DDIO} = 3.3V$
		1.8	—	—		$V_{DDIO} = 2.5V$
		1.3	—	—		$V_{DDIO} = 1.8V$
输入低电压	$V_{IL}$	—	—	0.8	V	$V_{DDIO} = 3.3V$
		—	—	0.7		$V_{DDIO} = 2.5V$
		—	—	0.5		$V_{DDIO} = 1.8V$
输入电流 (不包括上拉/下拉)	$I_{IN}$	—	—	10	$\mu A$	$V_{IN} = GND \sim V_{DDIO}$
<b>CMOS输出</b>						
输出高电压	$V_{OH}$	2.4	—	—	V	$V_{DDIO} = 3.3V$
		2.0	—	—		$V_{DDIO} = 2.5V$
		1.5	—	—		$V_{DDIO} = 1.8V$

# KSZ8795CLX

表6-1: 电气特性 (续)

参数	符号	最小值	典型值	最大值	单位	注
输出低电压	$V_{OL}$	—	—	0.4	V	$V_{DDIO} = 3.3V$
		—	—	0.4		$V_{DDIO} = 2.5V$
		—	—	0.3		$V_{DDIO} = 1.8V$
输出三态泄漏电流	$I_{OZ}$	—	—	10	$\mu A$	$V_{IN} = GND \sim V_{DDIO}$
<b>100BASE-TX发送 (在1:1变压器后以差分方式测量)</b>						
峰值差分输出电压	$V_O$	0.95	—	1.05	V	差分输出上有100 $\Omega$ 终结电阻
输出电压不平衡度	$V_{IMB}$	—	—	2	%	差分输出上有100 $\Omega$ 终结电阻
上升/下降时间	$t_r/t_f$	3	—	5	ns	—
上升/下降时间不平衡度		0	—	0.5		—
占空比失真	—	—	—	$\pm 0.5$	ns	—
过冲	—	—	—	5	%	—
输出抖动	—	0	0.75	1.4	ns	峰峰值
<b>10BASE-T接收</b>						
静噪阈值	$V_{SQ}$	300	400	585	mV	5 MHz方波
<b>10BASE-T发送 (在1:1变压器后以差分方式测量) <math>V_{DDAT} = 3.3V</math></b>						
峰值差分输出电压	$V_P$	2.2	2.5	2.8	V	差分输出上有100 $\Omega$ 终结电阻
输出抖动	—	—	1.4	3.5	ns	峰峰值
上升/下降时间	—	—	28	30	ns	—
<b>I/O引脚内部上拉和下拉电阻</b>						
I/O引脚有效上拉电阻	$R_{1.8PU}$	75	95	135	k $\Omega$	$V_{DDIO} = 1.8V$
I/O引脚有效下拉电阻	$R_{1.8PD}$	53	68	120		$V_{DDIO} = 1.8V$
I/O引脚有效上拉电阻	$R_{2.5PU}$	46	60	93		$V_{DDIO} = 2.5V$
I/O引脚有效下拉电阻	$R_{2.5PD}$	46	59	103		$V_{DDIO} = 2.5V$
I/O引脚有效上拉电阻	$R_{3.3PU}$	35	45	65		$V_{DDIO} = 3.3V$
I/O引脚有效下拉电阻	$R_{3.3PD}$	37	46	74		$V_{DDIO} = 3.3V$

## 7.0 时序图

图7-1: GMII信号时序图

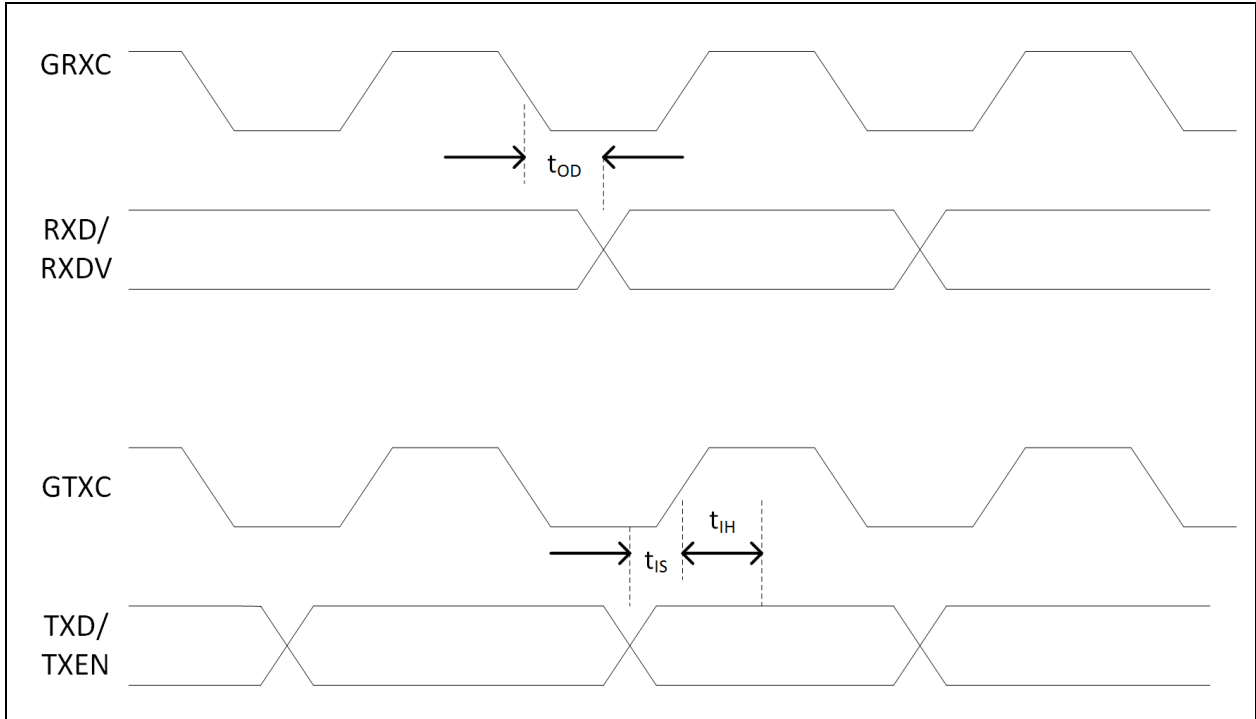


表7-1: GMII时序参数

符号	参数	最小值	典型值	最大值	单位
—	时钟周期	—	8	—	ns
$t_{IS}$	建立时间	1.2	—	—	
$t_{IH}$	保持时间	1.2	—	—	
$t_{OD}$	相对于时钟下降沿的输出延时	—	—	1	

# KSZ8795CLX

图7-2: RGMII V2.0规范

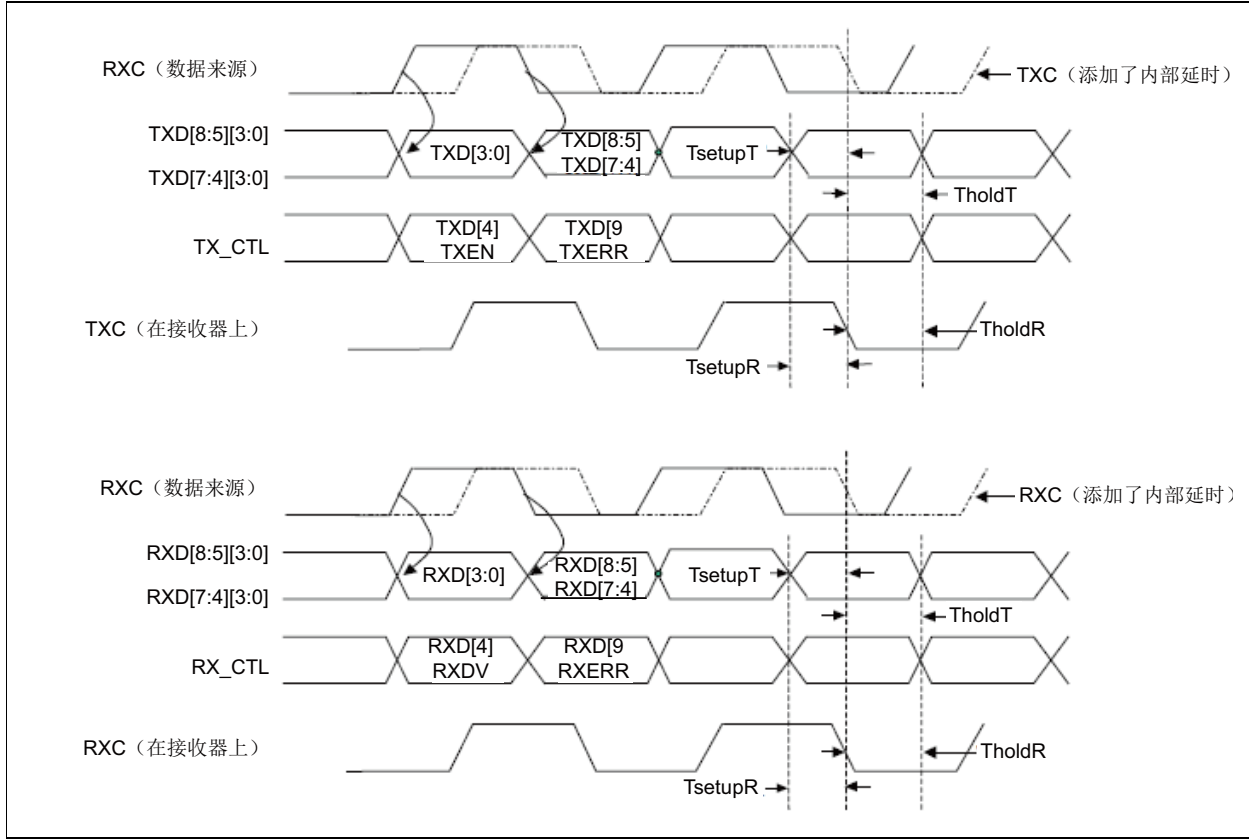


表7-2: RGMII时序参数

符号	参数	最小值	典型值	最大值	单位
TskewT	数据到时钟输出偏移 (发送器端) (注7-1)	-500	0	500	ps
TskewR	数据到时钟输入偏移 (接收器端) (注7-1)	1	—	2.6	ns
TsetupT	数据到时钟输出建立 (发送器端——集成延时)	1.0	2.0	—	
TholdT	时钟到数据输出保持 (发送器端——集成延时)	1.0	2.0	—	
TsetupR	数据到时钟输入建立 (接收器端——集成延时)	0.8	2.0	—	
TholdR	时钟到数据输入保持 (接收器端——集成延时)	0.8	2.0	—	
Tcyc	时钟周期持续时间 (注7-2)	7.2	8.0	8.8	
Duty_G	千兆位占空比	45	50	55	%
Duty_T	10/100T 占空比	40	50	60	
t <sub>r</sub> /t <sub>f</sub>	上升/下降时间 (20%-80%)	—	—	0.75	ns

注7-1 RGMII v2.0添加了内部延时 (RGMII-ID) 选项来匹配数据到时钟输出/输入偏移, 以实现RGMII发送和接收。有关详细信息, 请参见寄存器86的bit[4:3]。

注7-2 对于10 Mbps和100 Mbps, Tcyc将换算为400 ns ± 40 ns和40 ns ± 4 ns。

图7-3: MAC模式MII时序——通过MII接收数据

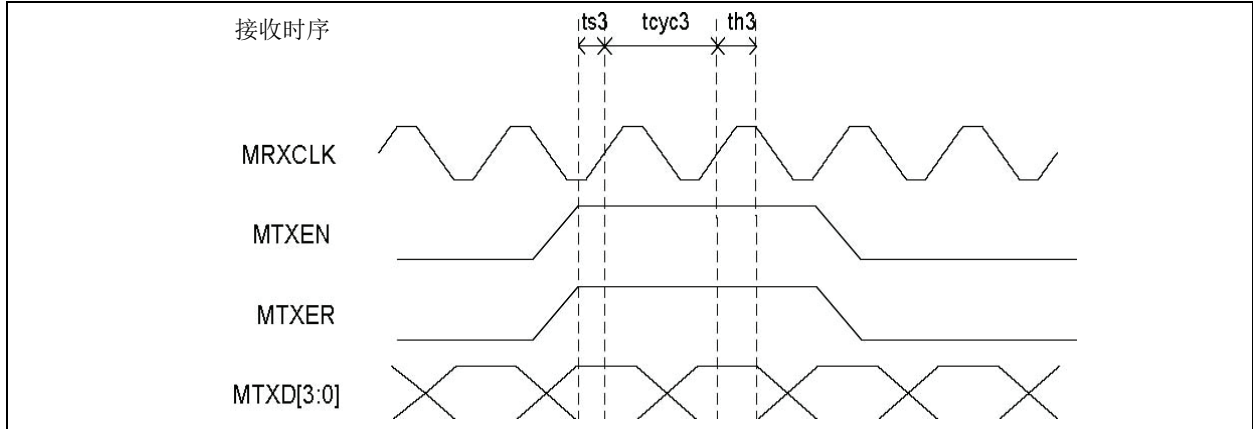


图7-4: MAC模式MII时序——通过MII发送数据

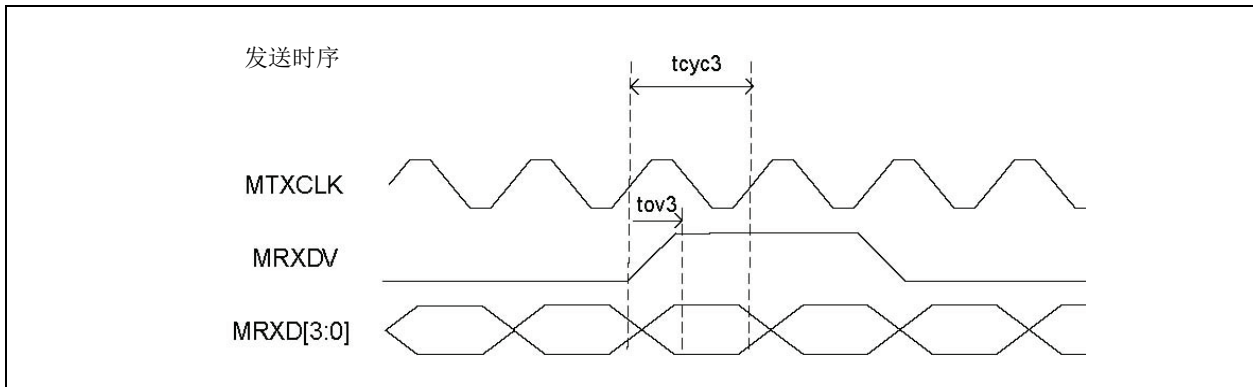


表7-3: MAC模式MII时序参数

符号	参数	10BASE-T/100BASE-TX			
		最小值	典型值	最大值	单位
$t_{cyc3}$	时钟周期	—	400/40	—	ns
$t_{s3}$	建立时间	2	—	—	
$t_{h3}$	保持时间	2	—	—	
$t_{ov3}$	输出有效	3	8	10	

# KSZ8795CLX

图7-5: PHY模式MII时序——通过MII接收数据

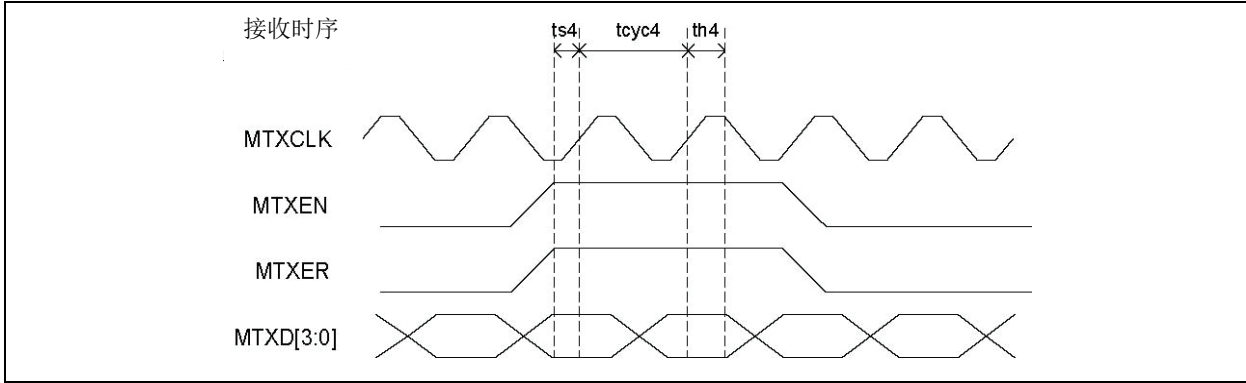


图7-6: PHY模式MII时序——通过MII发送数据

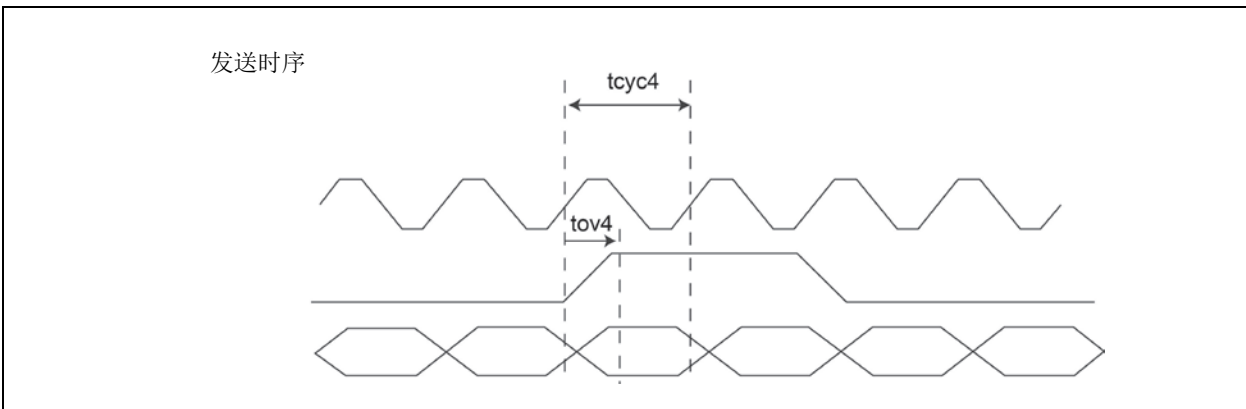


表7-4: PHY模式MII时序参数

符号	参数	10BASET/100BASET			
		最小值	典型值	最大值	单位
$t_{cyc4}$	时钟周期	—	400/40	—	ns
$t_{s4}$	建立时间	10	—	—	
$t_{h4}$	保持时间	0	—	—	
$t_{ov4}$	输出有效	16	20	25	

图7-7: RMII时序——通过RMII接收数据

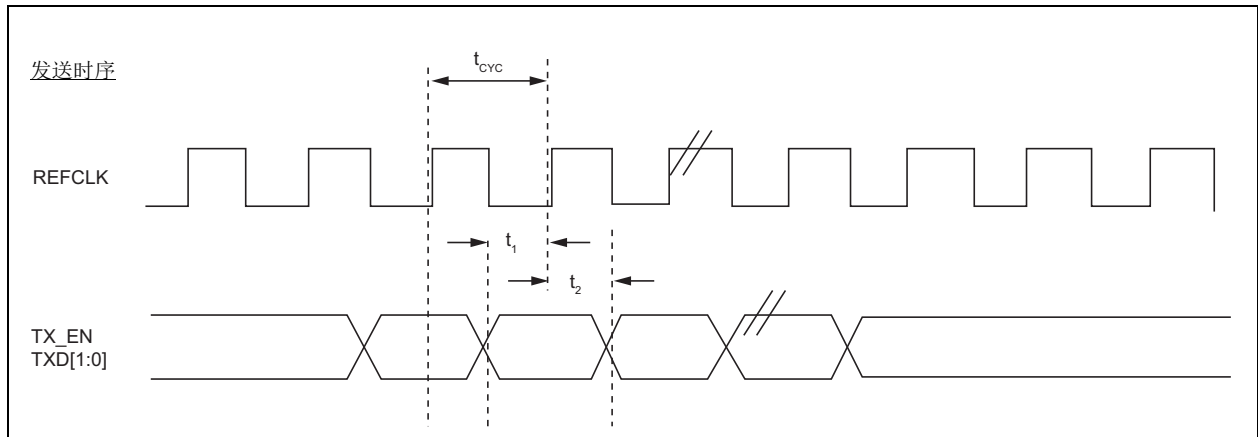


图7-8: RMII时序——通过RMII发送数据

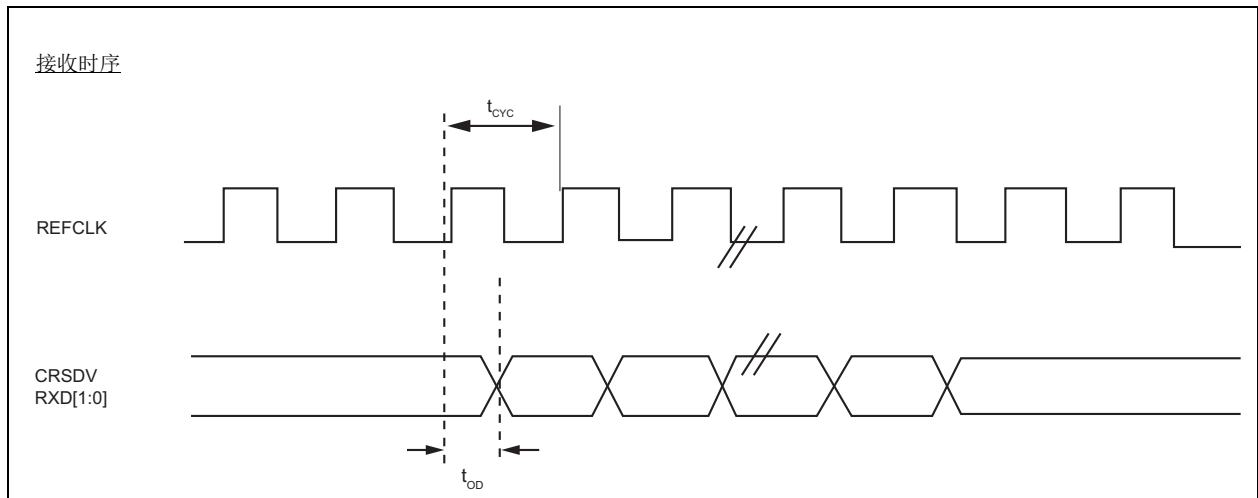


表7-5: RMII时序参数

符号	参数	最小值	典型值	最大值	单位
$t_{cyc}$	时钟周期	—	20	—	ns
$t_1$	建立时间	4	—	—	
$t_2$	保持时间	2	—	—	
$t_{od}$	输出延时	3	—	10	

# KSZ8795CLX

图7-9: SPI输入时序

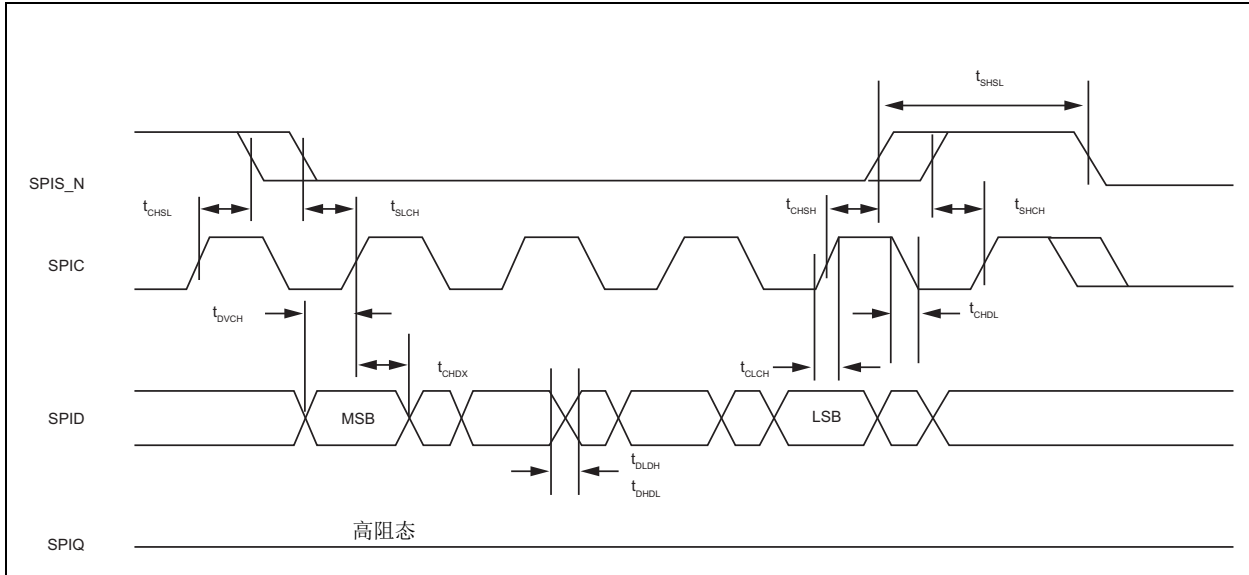


表7-6: SPI输入时序参数

符号	参数	最小值	典型值	最大值	单位
$f_C$	时钟频率	—	—	50	MHz
$t_{CHSL}$	SPIS_N 无效保持时间	2	—	—	ns
$t_{SLCH}$	SPIS_N 有效建立时间	4	—	—	
$t_{CHSH}$	SPIS_N 有效保持时间	2	—	—	
$t_{SHCH}$	SPIS_N 无效建立时间	4	—	—	
$t_{SHSL}$	SPIS_N 取消选择时间	10	—	—	
$t_{DVCH}$	数据输入建立时间	4	—	—	
$t_{CHDX}$	数据输入保持时间	2	—	—	$\mu s$
$t_{CLCH}$	时钟上升时间	—	—	1	
$t_{CHCL}$	时钟下降时间	—	—	1	
$t_{DLDH}$	数据输入上升时间	—	—	1	
$t_{DHDL}$	数据输入下降时间	—	—	1	



图7-10: SPI输出时序

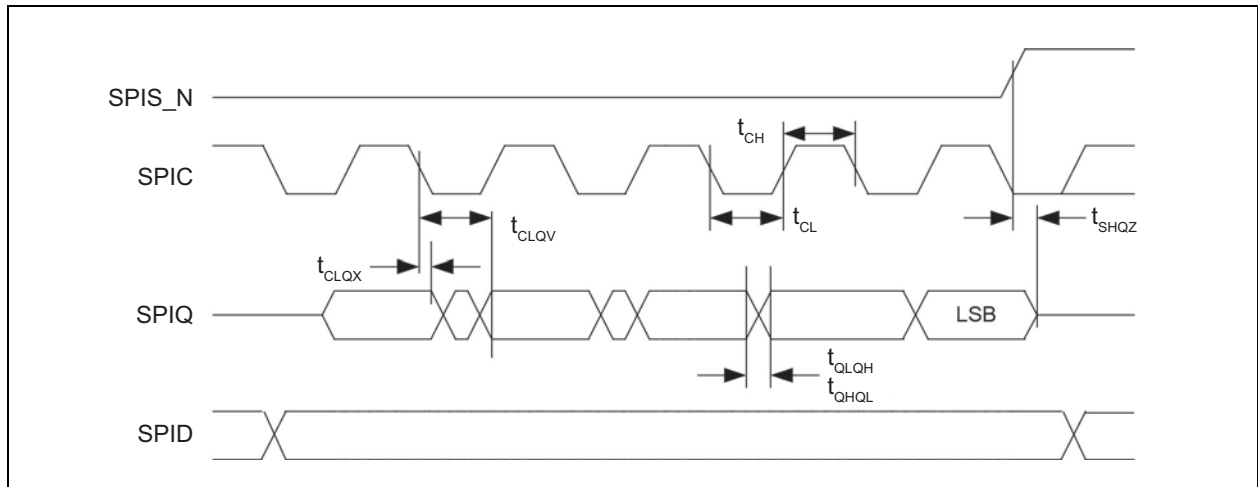


表7-7: SPI输出时序参数

符号	参数	最小值	典型值	最大值	单位
$f_C$	时钟频率	—	—	50	MHz
$t_{CLQX}$	SPIQ 保持时间	0	—	0	ns
$t_{CLQV}$	时钟到SPIQ低电平有效的的时间	—	—	60	
$t_{CH}$	时钟高电平时间	9	—	—	
$t_{CL}$	时钟低电平时间	9	—	—	
$t_{QLQH}$	SPIQ 上升时间	—	—	50	
$t_{QHQL}$	SPIQ 下降时间	—	—	50	
$t_{SHQZ}$	SPIQ 禁止时间	—	—	15	

# KSZ8795CLX

图7-11: 自动协商时序

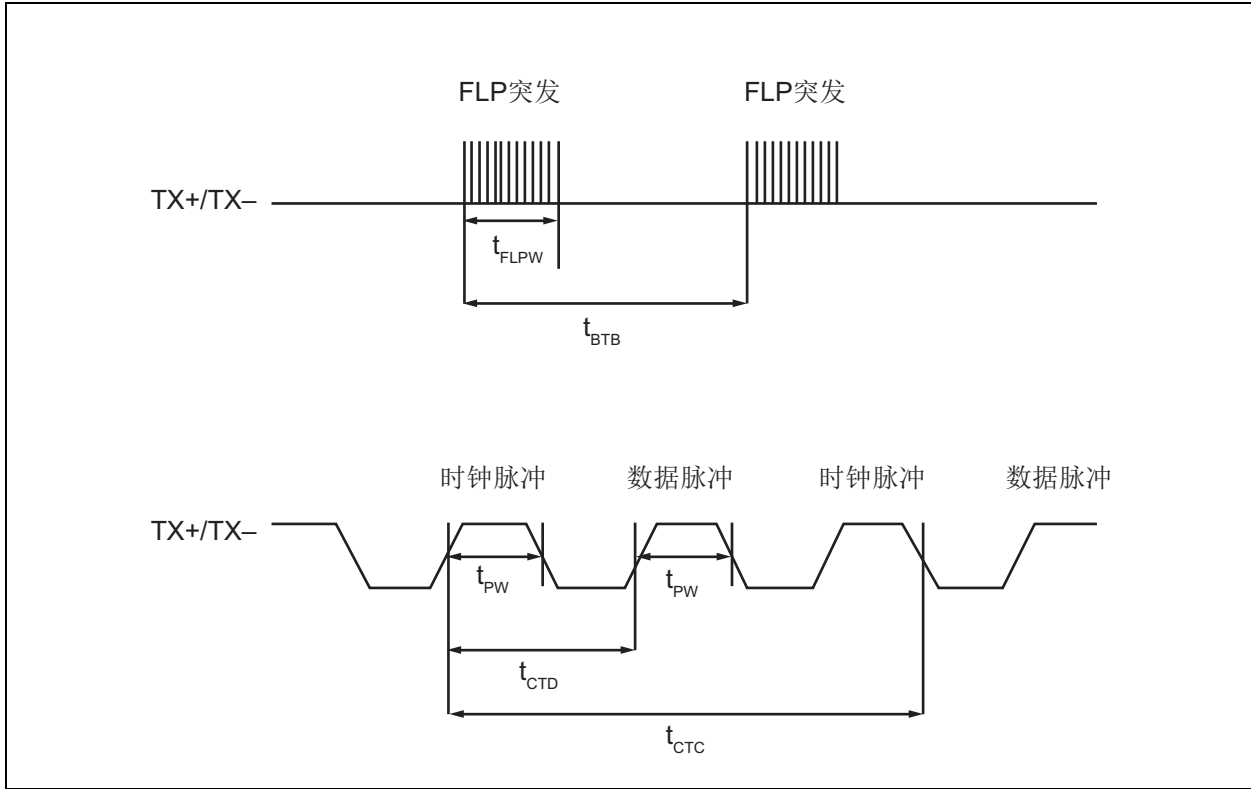


表7-8: 自动协商时序参数

符号	参数	最小值	典型值	最大值	单位
$t_{BTB}$	FLP突发至FLP突发的时间	8	16	24	ms
$t_{FLPW}$	FLP突发宽度	—	2	—	
$t_{PW}$	时钟/数据脉冲宽度	—	100	—	ns
$t_{CTD}$	时钟脉冲到数据脉冲的时间	55.5	64	69.5	$\mu$ s
$t_{CTC}$	时钟脉冲到时钟脉冲的时间	111	128	139	
—	每次突发的时钟/数据脉冲数	17	—	33	—

图7-12: MDC/MDIO时序

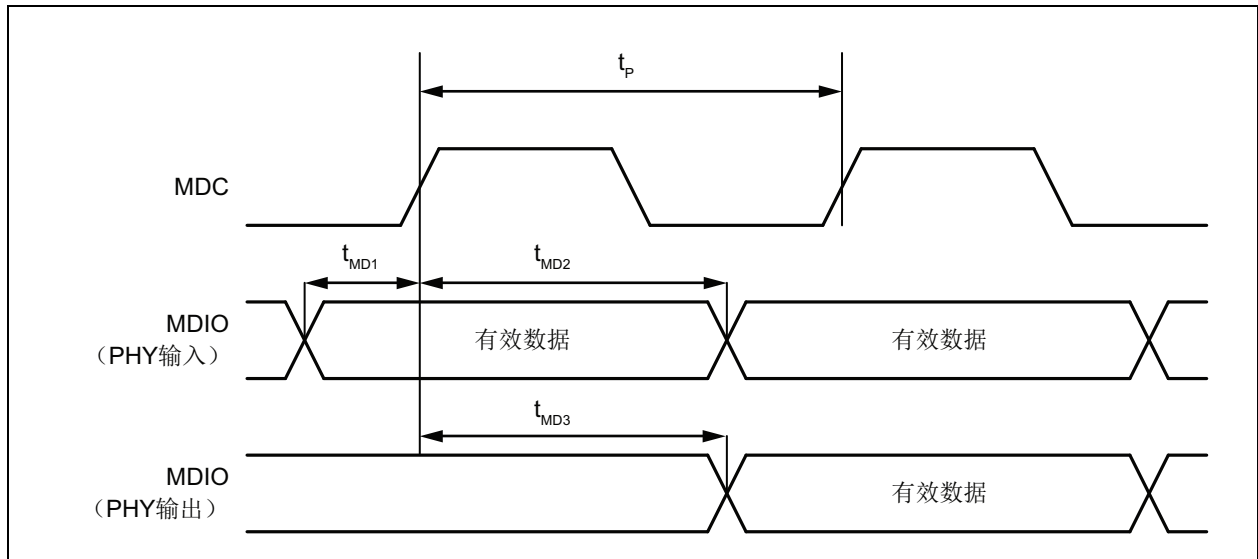


表7-9: MDC/MDIO典型时序参数

符号	参数	最小值	典型值	最大值	单位
$f_C$	时钟频率	—	2.5	25	MHz
$t_p$	MDC周期	—	400	—	ns
$t_{MD1}$	MDIO (PHY输入) 建立到MDC上升沿的时间	10	—	—	
$t_{MD2}$	从MDC上升沿开始的MDIO (PHY输入) 保持时间	4	—	—	
$t_{MD3}$	从MDC上升沿开始的MDIO (PHY输出) 延时	5	222	—	

# KSZ8795CLX

图7-13: 掉电/上电和复位时序

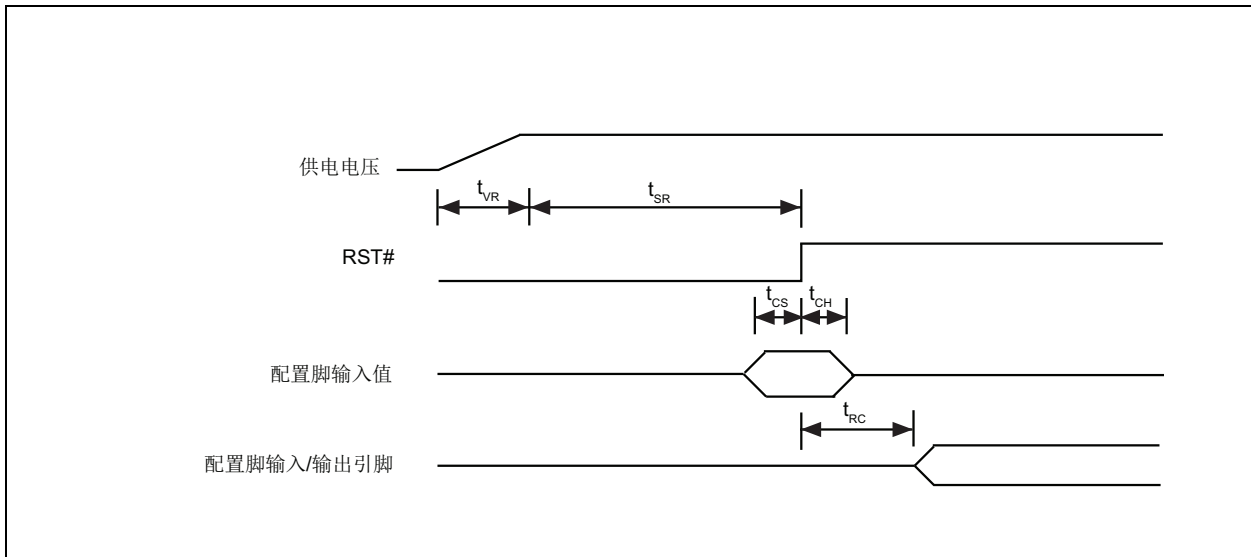


表7-10: 复位时序参数

符号	参数	最小值	典型值	最大值	单位
$t_{SR}$	稳定供电电压到复位高电平的时间	10	—	—	ms
$t_{CS}$	配置建立时间	5	—	—	ns
$t_{CH}$	配置保持时间	5	—	—	
$t_{RC}$	复位到配置脚输入引脚输出的时间	6	—	—	
$t_{VR}$	3.3V上升时间	200	—	—	$\mu s$

## 8.0 复位电路

为KSZ8795器件上电时，建议使用图8-1所示的分立式复位电路。对于复位电路信号来自其他器件（例如，CPU和FPGA等）的应用而言，建议使用图8-2所示的复位电路。

图8-1: 建议的复位电路

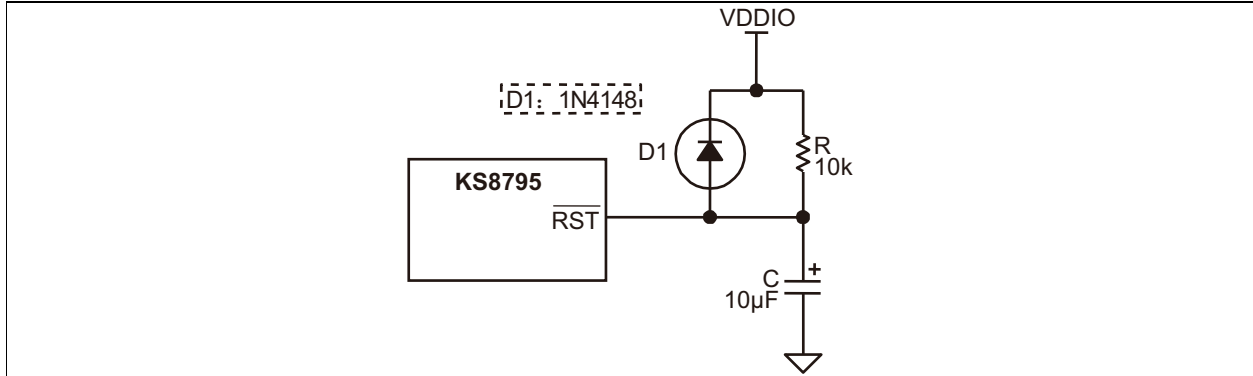


图8-2: 与CPU/FPGA复位接口的建议电路

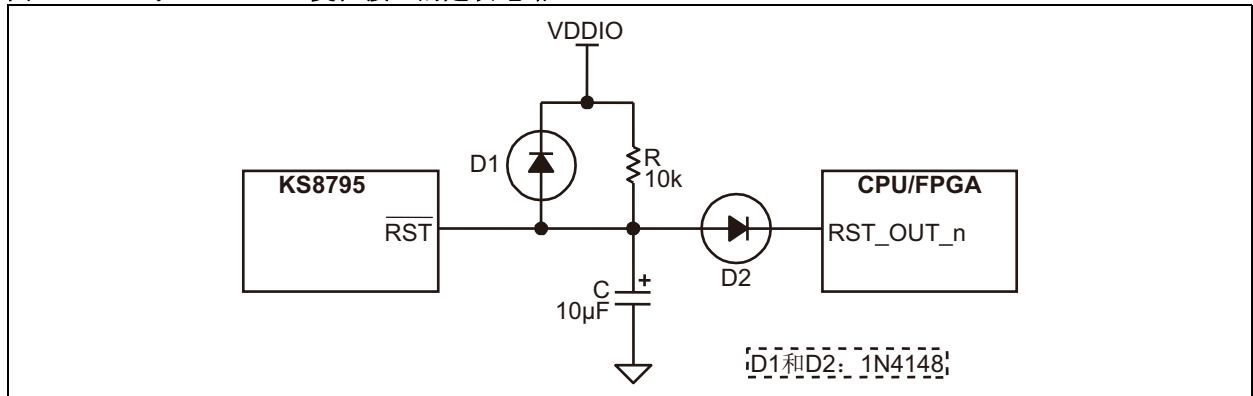


图8-2给出了由其他器件（例如，CPU或FPGA）驱动复位的应用中建议的复位电路。来自CPU/FPGA的复位输出RST\_OUT\_n在上电复位后提供热复位。如果在开关和CPU/FPGA之间使用不同的V<sub>DDIO</sub>电压，则需要D2。选择二极管D2的原则是可提供最大0.3V的V<sub>F</sub>（正向电压），例如VISHAY BAT54和MSS1P2L。此外，还可使用电平转换器。如果开关和CPU/FPGA使用相同的V<sub>DDIO</sub>电压，则不需要D2。

# KSZ8795CLX

## 9.0 隔离变压器的选择

线路接口处需要一个简单的1:1隔离变压器。当线路侧超出FCC要求时，建议使用带集成共模扼流器的隔离变压器。建议分离芯片侧RX/TX的中心抽头。100BASE-TX的IEEE 802.3u标准假定变压器损耗为0.5 dB。对于输电线路变压器，可通过减小ISET电阻值来增大线路驱动电流，从而补偿最高1.3 dB的插入损耗。表9-1给出了推荐的变压器特性。

表9-1: 25 MHz晶振/参考时钟选择标准

特性	值	测试条件
匝数比	1 CT :1 CT	—
开路电感（最小值）	350 $\mu$ H	100 mV, 100 kHz, 8 mA
插入损耗（最大值）	1.1 dB	0.1 MHz至100 MHz
HIPOT（最小值）	1500 V <sub>RMS</sub>	—

表9-2列出了为此器件提供兼容磁件的变压器供应商。

表9-2: 合格的磁件供应商

供应商和部件		自动MDIX	端口数	供应商和部件		自动MDIX	端口数
Pulse	H1164NL	支持	4	Pulse	H1102	支持	1
YCL	PH406082	支持	4	Bel Fuse	S558-5999-U7	支持	1
TDK	TLA-6T718A	支持	1	YCL	PT163020	支持	1
LanKom	LF-H41S	支持	1	Transpower	HB726	支持	1
Datatronic	NT79075	支持	1	Delta	LF8505	支持	1

## 10.0 参考晶振的选择

表10-1列出了此器件的典型参考晶振特性。

表10-1: 典型参考晶振特性

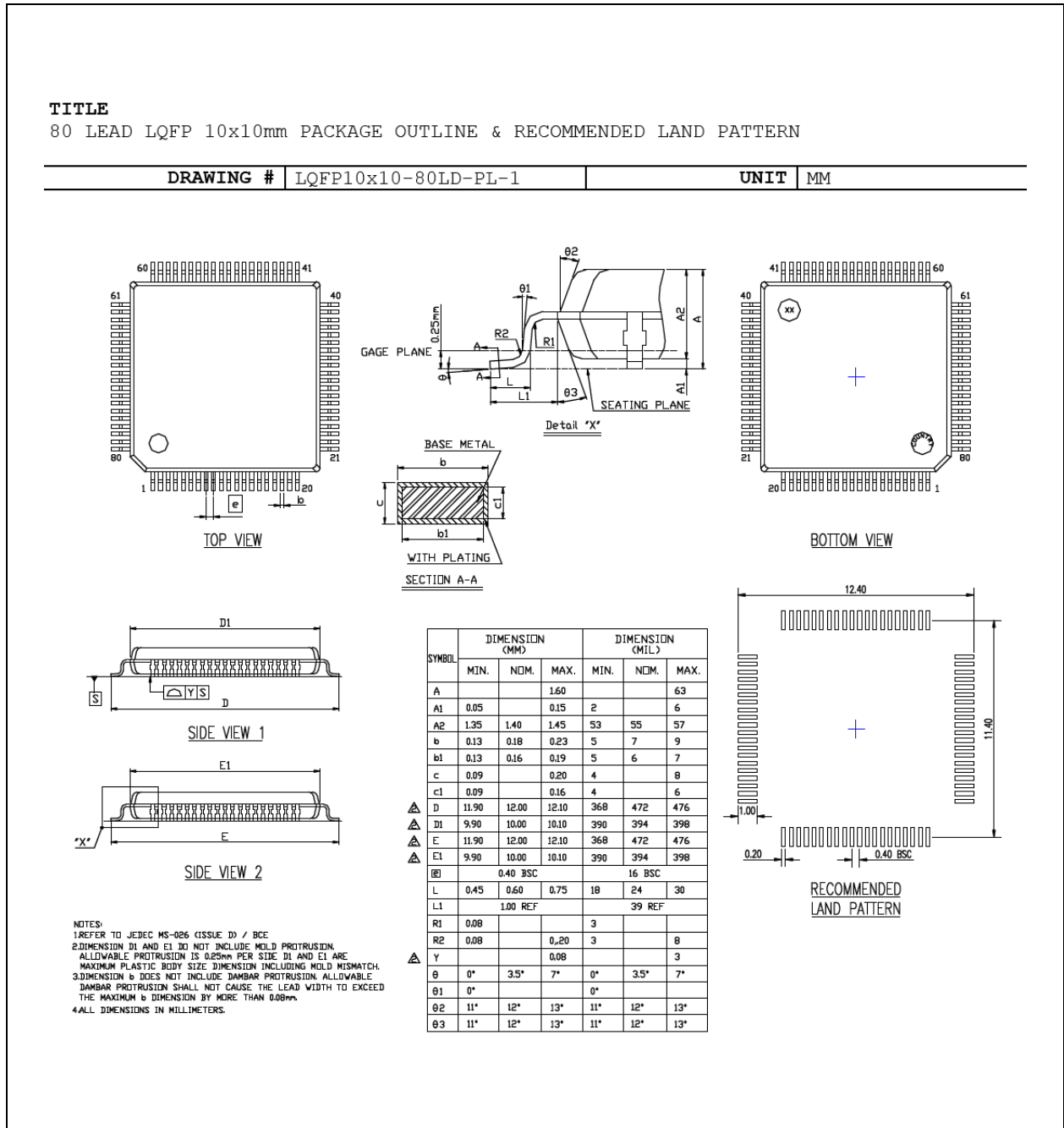
特性	值
频率	25.00000 MHz
频率容差（最大值）	$\leq \pm 50$ ppm
负载电容（最大值）（注10-1）	27 pF
串联电阻（最大ESR）	40 $\Omega$

注10-1 每个特定晶振规范的典型值有所不同。

## 11.0 封装外形

注： 最新的封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

图 11-1: 80 引脚 10 MM X 10 MM LQFP



# KSZ8795CLX

---

## 附录 A: 数据手册版本历史

表 A-1: 版本历史

版本	节/图/条目	修正
DS00002112A (2016年3月28日)	—	将 Micrel 数据手册 KSZ8795CLX 转换为 Microchip DS00002112A。通篇进行了少量文字更改。
	寄存器	更新了各个端口寄存器的说明。
	GMII 和 RGMII 图	更新了图片和相关表参数。



## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务** —— 产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 [www.microchip.com](http://www.microchip.com)。在“支持” (Support) 下，点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

# KSZ8795CLX

## 产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

器件编号	X	X	X	X	X
器件	接口	封装	特殊属性	温度	接合线
<b>器件:</b>	KSZ8795——具有千兆位GMII/RGMII和MII/RMII接口的集成5端口10/100管理型以太网开关				
<b>接口:</b>	C = 可配置				
<b>封装:</b>	L = 80引脚LQFP				
<b>特殊属性:</b>	X = 无				
<b>温度:</b>	C = 0°C至+70°C (商业级) I = -40°C至+85°C (工业级)				
<b>接合线:</b>	C = 铜制				

**示例:**

- a) KSZ8795CLXCC  
可配置接口  
80引脚LQFP  
商业级温度  
铜线接合
- b) KSZ8795CLXIC  
可配置接口  
80引脚LQFP  
工业级温度  
铜线接合

---

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

---

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适用性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KeeLoq<sup>®</sup> 跳码器件、串行 EEPROM、单片机外设、非易失性存储器 and 模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、KeeLoq 徽标、Kleer、LANCheck、LINK MD、MediaLB、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、PICSTART、PIC32 徽标、RightTouch、SpyNIC、SST、SST 徽标、SuperFlash 及 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、ETHERSYNCH、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 QUIET-WIRE 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、RightTouch 徽标、REAL ICE、Ripple Blocker、Serial Quad I/O、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2016, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-0752-2

## 全球销售及及服务网点

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:  
<http://www.microchip.com/support>

网址: [www.microchip.com](http://www.microchip.com)

**亚特兰大 Atlanta**  
Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

**奥斯汀 Austin, TX**  
Tel: 1-512-257-3370

**波士顿 Boston**  
Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

**芝加哥 Chicago**  
Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

**克里夫兰 Cleveland**  
Independence, OH  
Tel: 1-216-447-0464  
Fax: 1-216-447-0643

**达拉斯 Dallas**  
Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

**底特律 Detroit**  
Novi, MI  
Tel: 1-248-848-4000

**休斯敦 Houston, TX**  
Tel: 1-281-894-5983

**印第安纳波利斯 Indianapolis**  
Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453

**洛杉矶 Los Angeles**  
Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

**纽约 New York, NY**  
Tel: 1-631-435-6000

**圣何塞 San Jose, CA**  
Tel: 1-408-735-9110

**加拿大多伦多 Toronto**  
Tel: 905-695-1980  
Fax: 905-695-2078

### 亚太地区

亚太总部 **Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2943-5100

Fax: 852-2401-3431

**中国 - 北京**  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

**中国 - 成都**  
Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

**中国 - 重庆**  
Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

**中国 - 东莞**  
Tel: 86-769-8702-9880

**中国 - 广州**  
Tel: 86-20-8755-8029

**中国 - 杭州**  
Tel: 86-571-8792-8115  
Fax: 86-571-8792-8116

**中国 - 南京**  
Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

**中国 - 青岛**  
Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

**中国 - 上海**  
Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

**中国 - 沈阳**  
Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

**中国 - 深圳**  
Tel: 86-755-8864-2200  
Fax: 86-755-8203-1760

**中国 - 武汉**  
Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

**中国 - 西安**  
Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

**中国 - 厦门**  
Tel: 86-592-238-8138  
Fax: 86-592-238-8130

**中国 - 香港特别行政区**  
Tel: 852-2943-5100  
Fax: 852-2401-3431

### 亚太地区

**中国 - 珠海**  
Tel: 86-756-321-0040  
Fax: 86-756-321-0049

**台湾地区 - 高雄**  
Tel: 886-7-213-7828

**台湾地区 - 台北**  
Tel: 886-2-2508-8600  
Fax: 886-2-2508-0102

**台湾地区 - 新竹**  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

**澳大利亚 Australia - Sydney**  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

**印度 India - Bangalore**  
Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

**印度 India - New Delhi**  
Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

**印度 India - Pune**  
Tel: 91-20-3019-1500

**日本 Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**日本 Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

**韩国 Korea - Daegu**  
Tel: 82-53-744-4301  
Fax: 82-53-744-4302

**韩国 Korea - Seoul**  
Tel: 82-2-554-7200  
Fax: 82-2-558-5932 或  
82-2-558-5934

**马来西亚 Malaysia - Kuala Lumpur**  
Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

**马来西亚 Malaysia - Penang**  
Tel: 60-4-227-8870  
Fax: 60-4-227-4068

**菲律宾 Philippines - Manila**  
Tel: 63-2-634-9065  
Fax: 63-2-634-9069

**新加坡 Singapore**  
Tel: 65-6334-8870  
Fax: 65-6334-8850

**泰国 Thailand - Bangkok**  
Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### 欧洲

**奥地利 Austria - Wels**  
Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

**丹麦 Denmark-Copenhagen**  
Tel: 45-4450-2828  
Fax: 45-4485-2829

**法国 France - Paris**  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

**德国 Germany - Dusseldorf**  
Tel: 49-2129-3766400

**德国 Germany - Karlsruhe**  
Tel: 49-721-625370

**德国 Germany - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

**意大利 Italy - Milan**  
Tel: 39-0331-742611  
Fax: 39-0331-466781

**意大利 Italy - Venice**  
Tel: 39-049-7625286

**荷兰 Netherlands - Drunen**  
Tel: 31-416-690399  
Fax: 31-416-690340

**波兰 Poland - Warsaw**  
Tel: 48-22-3325737

**西班牙 Spain - Madrid**  
Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

**瑞典 Sweden - Stockholm**  
Tel: 46-8-5090-4654

**英国 UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820