

**GigaDevice Semiconductor Inc.**

**GDSCN832xx**

**用户手册**

1.0 版本

(2024 年 11 月)

# 目录

目录.....	2
图索引 .....	9
表索引 .....	11
<b>1. 系统和总线架构 .....</b>	<b>12</b>
<b>1.1. 总线架构.....</b>	<b>12</b>
<b>1.2. 存储器映射 .....</b>	<b>12</b>
<b>1.3. AHB 直接 / 间接访问.....</b>	<b>15</b>
1.3.1. 直接 AHB 传输访问 ESC 控制器寄存器.....	15
1.3.2. 间接传输访问 ESC 内核寄存器.....	15
1.3.3. 间接传输访问 ESC 内核 PRAM .....	16
<b>1.4. 忙碌状态下的寄存器保护 .....</b>	<b>17</b>
<b>1.5. OPB 传输超时功能 .....</b>	<b>17</b>
<b>1.6. EFUSE 功能 .....</b>	<b>17</b>
<b>1.7. EFUSE 寄存器定义.....</b>	<b>18</b>
1.7.1. 芯片 ID 寄存器 (EF_CHIP_ID) .....	18
1.7.2. EFUSE UID 读取寄存器 (EFUSE_UID_READ) .....	18
<b>1.8. ESC 内核控制器 (ESC_CCTL) .....</b>	<b>19</b>
<b>1.9. ESC 内核控制器寄存器定义.....</b>	<b>19</b>
1.9.1. ESC_CCTL_DATA 数据寄存器(ESC_CCTL_DATA).....	19
1.9.2. ESC CCTL 命令寄存器 (ESC_CCTL_CMD) .....	20
1.9.3. ESC PRAM FIFO 数据读取寄存器 (ESC_PRAM_FIFO_DR) .....	20
1.9.4. ESC PRAM 地址和长度读取寄存器 (ESC_PRAM_ALR) .....	21
1.9.5. ESC PRAM 命令读取寄存器 (ESC_PRAM_CR) .....	21
1.9.6. ESC PRAM FIFO 数据写寄存器 (ESC_PRAM_FIFO_DW) .....	22
1.9.7. ESC PRAM 地址和长度写寄存器 (ESC_PRAM_ALW) .....	22
1.9.8. ESC PRAM 命令写寄存器 (ESC_PRAM_CW) .....	23
1.9.9. ESC OPB 控制和状态寄存器 (ESC_OPB_CS) .....	24
<b>1.10. 系统配置控制器 (SYSCFG) .....</b>	<b>25</b>
<b>1.11. 系统配置寄存器定义.....</b>	<b>25</b>
1.11.1. 系统配置寄存器 0 (SYSCFG_CFG0) .....	25
1.11.2. 系统配置芯片 ID 寄存器 (SYSCFG_CHIPID) .....	26
1.11.3. 系统配置芯片版本寄存器 (SYSCFG_CHIPVER) .....	27
1.11.4. 系统配置保留寄存器 (SYSCFG_RESERVED) .....	27
<b>2. 电源管理单元 (PMU) .....</b>	<b>28</b>
<b>2.1. 简介.....</b>	<b>28</b>

<b>2.2.</b>	<b>主要特征</b> .....	<b>28</b>
<b>2.3.</b>	<b>功能说明</b> .....	<b>28</b>
2.3.1.	设备就绪.....	28
2.3.2.	EFUSE 供电.....	28
2.3.3.	PHY 唤醒事件检测.....	29
2.3.4.	PME 唤醒通知.....	29
2.3.5.	模块级节能模式.....	29
2.3.6.	设备级节能模式.....	29
2.3.7.	进入设备级节能模式.....	30
2.3.8.	退出设备级别的节能模式.....	31
<b>2.4.</b>	<b>寄存器定义</b> .....	<b>32</b>
2.4.1.	控制寄存器 0 (PMU_CTL0).....	32
2.4.2.	控制寄存器 1 (PMU_CTL1).....	34
2.4.3.	数据接口参考值寄存器 (PMU_PDIREFVAL).....	34
<b>3.</b>	<b>复位和时钟单元 (RCU)</b> .....	<b>35</b>
<b>3.1.</b>	<b>复位控制单元</b> .....	<b>35</b>
3.1.1.	简介.....	35
3.1.2.	主要特征.....	35
3.1.3.	功能说明.....	35
<b>3.2.</b>	<b>时钟控制单元 (CCTL)</b> .....	<b>36</b>
3.2.1.	简介.....	36
3.2.2.	主要特征.....	36
3.2.3.	功能说明.....	36
<b>3.3.</b>	<b>RCU 寄存器</b> .....	<b>37</b>
3.3.1.	AHB 使能寄存器 (RCU_AHBMEN).....	37
3.3.2.	APB 使能寄存器 (RCU_APBEN).....	38
3.3.3.	内核使能寄存器 (RCU_COREEN).....	39
3.3.4.	时钟配置寄存器 (RCU_CLKCFG).....	40
3.3.5.	复位配置寄存器 (RCU_RSTCFG).....	41
3.3.6.	PLL 配置密钥寄存器 (RCU_PLL_CFG_KEY).....	42
3.3.7.	复位标志寄存器 (RCU_PRSTF).....	42
<b>4.</b>	<b>中断控制器 (INTC)</b> .....	<b>44</b>
<b>4.1.</b>	<b>概述</b> .....	<b>44</b>
<b>4.2.</b>	<b>特征</b> .....	<b>44</b>
<b>4.3.</b>	<b>功能说明</b> .....	<b>44</b>
4.3.1.	软件中断.....	45
4.3.2.	设备准备就绪中断.....	45
4.3.3.	以太网 PHY 中断.....	45
4.3.4.	定时器中断.....	45
4.3.5.	PME 中断.....	46

4.3.6.	AHB2OPB 桥中断 .....	46
4.3.7.	EtherCAT 中断 .....	46
4.3.8.	时钟输出测试模式 .....	46
<b>4.4.</b>	<b>INTC 寄存器 .....</b>	<b>46</b>
4.4.1.	控制寄存器 (INTC_CTL) .....	46
4.4.2.	标志寄存器 (INTC_FLAG) .....	48
4.4.3.	使能寄存器 (INTC_EN) .....	49
<b>5.</b>	<b>通用输入/输出 (GPIO) .....</b>	<b>51</b>
5.1.	概述 .....	51
5.2.	特征 .....	51
5.3.	功能概述 .....	51
5.3.1.	GPIO 引脚配置 .....	51
5.3.2.	外部中断/事件线路 .....	51
5.3.3.	备用功能 .....	51
5.3.4.	模拟配置 .....	54
5.3.5.	备用功能(AF)配置 .....	54
5.4.	寄存器定义 .....	55
5.4.1.	端口输出模式寄存器 0 (GPIO0_OMODE0) .....	55
5.4.2.	端口输出模式寄存器 1 (GPIO0_OMODE1) .....	56
5.4.3.	端口输出模式寄存器 2 (GPIO1_OMOD0) .....	57
5.4.4.	端口输出模式寄存器 3 (GPIO1_OMOD1) .....	58
5.4.5.	端口上拉/下拉寄存器 0 (GPIO0_PUD0) .....	59
5.4.6.	端口上拉/下拉寄存器 1 (GPIO0_PUD1) .....	60
5.4.7.	端口上拉/下拉寄存器 2 (GPIO1_PUD0) .....	61
5.4.8.	端口上拉/下拉寄存器 3 (GPIO1_PUD1) .....	62
5.4.9.	EXMC 控制寄存器 (EXMC_CTL) .....	63
<b>6.</b>	<b>定时器 (TIMER) .....</b>	<b>65</b>
6.1.	基本定时器 .....	65
6.1.1.	简介 .....	65
6.1.2.	主要特征 .....	65
6.1.3.	结构框图 .....	65
6.1.4.	功能说明 .....	65
6.1.5.	寄存器定义 .....	66
6.2.	独立运行计数器 .....	67
6.2.1.	简介 .....	67
6.2.2.	主要特征 .....	67
6.2.3.	结构框图 .....	67
6.2.4.	功能说明 .....	68
6.2.5.	寄存器定义 .....	68
<b>7.</b>	<b>总线接口 (PDI Wrapper) .....</b>	<b>70</b>

<b>7.1. SPI / QSPI / OSPI 从机</b> .....	<b>70</b>
7.1.1. 简介.....	70
7.1.2. 主要特征.....	70
7.1.3. 模块框图.....	71
7.1.4. SPI 信号描述.....	71
7.1.5. SPI/QSPI/OSPI 从机控制器.....	72
<b>7.2. 外部存储器控制器 (EXMC)</b> .....	<b>90</b>
7.2.1. 概述.....	90
7.2.2. 主要特征.....	90
7.2.3. 功能概述.....	90
<b>8. 以太网 PHYS</b> .....	<b>95</b>
<b>8.1. 简介</b> .....	<b>95</b>
<b>8.2. 主要特征</b> .....	<b>95</b>
<b>8.3. 功能概述</b> .....	<b>96</b>
8.3.1. 操作模式.....	96
8.3.2. MII 接口.....	96
8.3.3. SMI 接口.....	96
8.3.4. 自动 MDI/MDIX 和极性配置.....	96
8.3.5. 环回模式.....	96
8.3.6. Wake-On-LAN.....	97
8.3.7. LED 模式.....	97
8.3.8. LPI 信令.....	98
<b>8.4. PHY 寄存器</b> .....	<b>98</b>
8.4.1. Page0 寄存器.....	98
8.4.2. Page1 寄存器.....	115
8.4.3. Page2 寄存器.....	115
8.4.4. Page3 寄存器.....	117
8.4.5. Page6 寄存器.....	118
8.4.6. Page9 寄存器.....	119
8.4.7. MDIO 寄存器.....	125
<b>9. EtherCAT</b> .....	<b>142</b>
<b>9.1. 简介</b> .....	<b>142</b>
<b>9.2. 主要特征</b> .....	<b>142</b>
9.2.1. 结构框图.....	142
9.2.2. EtherCAT 从站控制器功能块.....	143
<b>9.3. 功能描述</b> .....	<b>144</b>
9.3.1. 过程数据接口 (PDI) .....	144
9.3.2. FMMU .....	146
9.3.3. 同步管理器(SyncManager) .....	147
9.3.4. 分布式时钟 (DC) .....	147

9.3.5.	EtherCAT 状态机(ESM) .....	148
9.3.6.	EEPROM .....	149
9.3.7.	复位 .....	149
9.3.8.	中断 .....	149
9.3.9.	LED .....	151
<b>9.4.</b>	<b>ESC 寄存器定义 .....</b>	<b>151</b>
9.4.1.	ESC 类型寄存器 (ESC_TYPE) .....	151
9.4.2.	ESC 版本寄存器 (ESC_REVISION) .....	151
9.4.3.	ESC 编译寄存器 (ESC_BUILD) .....	152
9.4.4.	ESC FMMU 数目寄存器 (ESC_FMMUS) .....	152
9.4.5.	ESC 同步管理器的数目寄存器 (ESC_SYNCMANAGERS) .....	153
9.4.6.	ESC RAM 大小寄存器 (ESC_RAMSIZE) .....	153
9.4.7.	ESC 端口描述寄存器 (ESC_PORT_DESCRIPTION) .....	153
9.4.8.	ESC 功能支持寄存器 (ESC_FEATURES_SUPPORTED) .....	154
9.4.9.	ESC 配置的站点地址寄存器 (ESC_STATION_ADDRESS) .....	155
9.4.10.	ESC 配置的站点别名寄存器 (ESC_STATION_ALIAS) .....	156
9.4.11.	写使能寄存器 (WRITE_ENABLE) .....	156
9.4.12.	ESC 写保护寄存器 (ESC_WRITE_PROTECTION) .....	157
9.4.13.	ESC 写寄存器使能寄存器 (ESC_WRITE_ENABLE) .....	157
9.4.14.	ESC 写寄存器保护寄存器 (ESC_WRITE_PROTECTION) .....	157
9.4.15.	ESC 复位 ECAT 寄存器 (ESC_RESET_ECAT) .....	158
9.4.16.	ESC 复位 PDI 寄存器 (ESC_RESET_PDI) .....	159
9.4.17.	ESC DL 控制寄存器 (ESC_DL_CONTROL) .....	159
9.4.18.	ESC 物理读写偏移寄存器 (ESC_PHYSICAL_OFFSET) .....	161
9.4.19.	ESC DL 状态寄存器 (ESC_DL_STATUS) .....	161
9.4.20.	ESC AL 控制寄存器 (ESC_AL_CONTROL) .....	163
9.4.21.	ESC AL 状态寄存器 (ESC_AL_STATUS) .....	164
9.4.22.	ESC AL 状态代码寄存器 (ESC_AL_STATUS_CODE) .....	165
9.4.23.	ESC 运行 LED 覆盖寄存器 (ESC_RUN_LED) .....	165
9.4.24.	ESC 错误 LED 覆盖寄存器 (ESC_ERR_LED) .....	166
9.4.25.	ESC PDI 控制寄存器 (ESC_PDI_CONTROL) .....	166
9.4.26.	ESC 配置寄存器 (ESC_CONFIG) .....	167
9.4.27.	ESC PDI 信息寄存器 (ESC_PDI_INFM) .....	168
9.4.28.	ESC PDI 配置寄存器 (ESC_PDI_CONFIG) .....	168
9.4.29.	ESC 同步/锁存器配置寄存器 (ESC_SL_CONFIG) .....	169
9.4.30.	ESC PDI 扩展配置寄存器 (ESC_DEXT_CFG) .....	170
9.4.31.	ESC 事件屏蔽寄存器 (ESC_EVENT_MASK) .....	171
9.4.32.	ESC PDI AL 事件寄存器 (ESC_PDI_AL_EVENT) .....	172
9.4.33.	ESC 事件请求寄存器 (ESC_EVENT_RQST) .....	172
9.4.34.	ESC AL 事件请求寄存器 (ESC_AL_EVENT_RQST) .....	174
9.4.35.	PortX 接收错误计数器寄存器 (RX_PORTX_ERROR) (X = 0,1,2,3) .....	176
9.4.36.	Port X 转发接收错误计数器寄存器 (FRX_PORTX_ERROR) (X = 0,1,2,3) .....	176
9.4.37.	ESC 处理单元错误计数器寄存器 (ESC_PU_ERROR) .....	177
9.4.38.	ESC PDI 错误计数器寄存器 (ESC_PDI_ERROR) .....	177

9.4.39.	ESC PortX 丢失链路计数器寄存器 (ESC_PORTX_LOST_LINK) (X = 0,1,2,3)	178
9.4.40.	ESC 看门狗分频器寄存器 (ESC_WTG_DIVIDER)	178
9.4.41.	ESC 看门狗定时器 PDI 寄存器 (ESC_WTG_TIME)	178
9.4.42.	ESC 看门狗定时器过程数据寄存器 (ESC_WTG_TPD)	179
9.4.43.	ESC 看门狗状态过程数据寄存器 (ESC_WTG_STATUS)	179
9.4.44.	ESC 看门狗计数器过程数据寄存器 (ESC_WTG_CTR)	180
9.4.45.	ESC 看门狗计数器 PDI 寄存器 (ESC_WTG_CTR_PDI)	180
9.4.46.	ESC EEPROM 配置寄存器 (ESC_EEPROM_CONFIG)	180
9.4.47.	ESC EEPROM PDI 访问寄存器 (ESC_EEPROM_ACCESS)	181
9.4.48.	ESC EEPROM 控制 / 状态寄存器 (ESC_EEPROM_CONTROL)	181
9.4.49.	ESC EEPROM 地址寄存器 (ESC_EEPROM_ADDR)	183
9.4.50.	ESC EEPROM 数据寄存器 (ESC_EEPROM_DATA)	183
9.4.51.	ESC MII 管理控制/状态寄存器 (ESC_MII_CTL)	184
9.4.52.	ESC PHY 地址寄存器 (ESC_PHY_ADDR)	185
9.4.53.	ESC PHY 寄存器地址寄存器 (ESC_PHY_RADDR)	186
9.4.54.	ESC PHY 数据寄存器 (ESC_PHY_DATA)	186
9.4.55.	MII 管理 ECAT 访问状态寄存器 (MII_ECAT_STATE)	186
9.4.56.	MII 管理 PDI 访问状态寄存器 (MII_PDI_STATE)	187
9.4.57.	PHY Port X 状态寄存器 (PHY_PORTX_STA) (X = 0,1,2,3)	187
9.4.58.	FMMUX 逻辑起始地址寄存器 (FMMUX_LOGIC_ADDR) (X = 0..F)	188
9.4.59.	FMMUX 长度寄存器 (FMMUX_LENGTH) (X = 0..F)	189
9.4.60.	FMMUX 逻辑起始位寄存器 (FMMUX_STRA_BIT) (X = 0..F)	189
9.4.61.	FMMUX 逻辑停止位寄存器 (FMMUX_STOP_BIT) (X = 0..F)	189
9.4.62.	FMMUX 物理起始地址寄存器 (FMMUX_ADRR) (X = 0..F)	190
9.4.63.	FMMUX 物理起始位寄存器 (FMMUX_PSBIT) (X = 0..F)	190
9.4.64.	FMMUX 类型寄存器 (FMMUX_TYPE) (X = 0..F)	190
9.4.65.	FMMUX 激活寄存器 (FMMUX_ACTIVE) (X = 0..F)	191
9.4.66.	SyncManager X 物理起始地址寄存器 (SMX_ADRR) (X = 0..F)	191
9.4.67.	SyncManager X 长度寄存器 (SMX_LENGTH) (X = 0..F)	192
9.4.68.	SyncManager X 控制寄存器 (SMX_CTL) (X = 0..F)	192
9.4.69.	SyncManager X 状态寄存器 (SMX_STA) (X = 0..F)	193
9.4.70.	SyncManager X 激活寄存器 (SMX_ACTIVE) (X = 0..F)	194
9.4.71.	PDI 控制寄存器 (SMX_PDICTL) (X = 0..F)	194
9.4.72.	ESC 接收时间端口 0 寄存器 (ESC_RECVE_TIMEP0)	195
9.4.73.	ESC 接收时间端口 1 寄存器 (ESC_RECVE_TIMEP1)	195
9.4.74.	ESC 接收时间端口 2 寄存器 (ESC_RECVE_TIMEP2)	196
9.4.75.	ESC 系统时间寄存器 (ESC_SYS_TIME)	196
9.4.76.	ESC 接收时间 ECAT 处理单元寄存器 (ESC_RCVTIME)	197
9.4.77.	ESC 系统时间偏移寄存器 (ESC_OFFSET_TIME)	197
9.4.78.	ESC 系统时间延迟寄存器 (ESC_DELAY_TIME)	198
9.4.79.	ESC 系统时间差值寄存器 (ESC_DIFF_TIME)	198
9.4.80.	ESC 速度计数器起始值寄存器 (ESC_COUNT_START)	199
9.4.81.	ESC 速度计数器差值寄存器 (ESC_COUNT_DIFF)	199
9.4.82.	ESC 系统时间差值滤波器深度寄存器 (ESC_TIME_DIFF)	199

9.4.83.	ESC 速度计数器滤波器深度寄存器 (ESC_SPEED_COUNT) .....	200
9.4.84.	ESC 循环单元控制寄存器 (ESC_UNIT_CTL) .....	200
9.4.85.	ESC 激活寄存器 (ESC_REGISTER_ACTIVE) .....	201
9.4.86.	ESC 同步信号寄存器的脉冲长度 (ESC_PLEN_SM) .....	202
9.4.87.	ESC 激活状态寄存器 (ESC_ACTIVE_STATUS) .....	202
9.4.88.	ESC SYNC0 状态寄存器 (ESC_SYNC0_STATUS) .....	203
9.4.89.	ESC SYNC1 状态寄存器 (ESC_SYNC1_STATUS) .....	203
9.4.90.	ESC 起始时间循环操作寄存器 (ESC_START_TIME) .....	204
9.4.91.	ESC 下一个 SYNC1 脉冲寄存器 (ESC_NEXT_SYNC1) .....	204
9.4.92.	ESC SYNC0 周期时间寄存器 (ESC_SYNC0_CYCLE) .....	205
9.4.93.	ESC SYNC1 周期时间寄存器 (ESC_SYNC1_CYCLE) .....	205
9.4.94.	ESC LATHC0 控制寄存器 (ESC_LATCH0_CTL) .....	206
9.4.95.	ESC LATCH1 控制寄存器 (ESC_LATCH1_CTL) .....	206
9.4.96.	ESC LATCH0 状态寄存器 (ESC_LATCH0_STATUS) .....	207
9.4.97.	ESC LATCH1 状态寄存器 (ESC_LATCH1_STATUS) .....	207
9.4.98.	ESC LATCH0 时间上升沿寄存器 (ESC_LATCH0_POSITIVE) .....	208
9.4.99.	ESC LATCH0 时间下降沿寄存器 (ESC_LATCH0_NEGATIVE) .....	208
9.4.100.	ESC LATCH1 时间上升沿寄存器 (ESC_LATCH1_POSITIVE) .....	209
9.4.101.	ESC LATCH1 时间下降沿寄存器 (ESC_LATCH1_NEGATIVE) .....	209
9.4.102.	ESC 缓冲区变化事件时间寄存器 (ESC_EVENT_TIME) .....	210
9.4.103.	ESC PDI 缓冲区起始事件时间寄存器 (ESC_PDI_SEVENT_TIME) .....	210
9.4.104.	ESC PDI 缓冲区变化事件时间寄存器 (ESC_PDI_CEVENT_TIME) .....	211
9.4.105.	ESC 产品 ID 寄存器 (ESC_PRODUCT_ID) .....	211
9.4.106.	ESC 供应商 ID 寄存器 (ESC_VENDOR_ID) .....	212
9.4.107.	ESC 数字 I/O 输出数据寄存器 (ESC_DIG_DATA) .....	212
9.4.108.	ESC 通用输出寄存器 (ESC_GP_OUTPUT) .....	212
9.4.109.	ESC 通用输入寄存器 (ESC_GP_INPUTS) .....	213
9.4.110.	ESC 用户 RAM 寄存器 (ESC_USER_RAM) .....	213
9.4.111.	ESC PDI 数字 I/O 输入数据寄存器 (ESC_PDI_DATA) .....	213
9.4.112.	ESC 过程数据 RAM (ESC_PDRAM) .....	214
<b>10.</b>	<b>修订历史 .....</b>	<b>215</b>

# 图索引

图 1-1. ESC 的总线架构.....	12
图 1-2. EFUSE 控制框图.....	18
图 2-1. PME 中断待处理.....	29
图 3-1. 时钟树.....	36
图 3-2. HXTAL 时钟源.....	37
图 4-1. 中断框图.....	45
图 5-1. 端口连接 PHYS.....	53
图 5-2. 模拟配置的基本结构.....	54
图 5-3. 备用功能配置的基本结构.....	54
图 6-1. 基本定时器结构框图.....	65
图 6-2. 向下计数时序图.....	66
图 6-3. 独立运行计数器结构框图.....	68
图 7-1. PDI Wrapper 框图.....	70
图 7-2. SPI 的 block 图.....	71
图 7-3. 使能 QSPI.....	75
图 7-4. 使能 OSPI.....	76
图 7-5. SPI 模式复位 SPI.....	76
图 7-6. QSPI 模式复位 QSPI.....	77
图 7-7. OSPI 模式复位 OSPI.....	77
图 7-8. SPI 读取.....	78
图 7-9. QSPI 读取.....	78
图 7-10. OSPI 读取.....	79
图 7-11. SPI 双输出读取.....	80
图 7-12. SPI 四线输出读取.....	81
图 7-13. SPI 双 I/O 读取.....	82
图 7-14. SPI 四线 I/O 读取.....	83
图 7-15. SPI 写入.....	84
图 7-16. QSPI 写入.....	84
图 7-17. OSPI 写入.....	85
图 7-18. SPI 双数据写入.....	86
图 7-19. SPI 四线数据写入.....	87
图 7-20. SPI 双地址/数据写入.....	88
图 7-21. SPI 四路地址/数据写入.....	89
图 7-22. EXMC 模块框图.....	90
图 7-23. 异步写传输.....	92
图 7-24. 异步读传输.....	92
图 7-25. 包含 nwait 的背靠背传输.....	92
图 7-26. 同步模式下写传输.....	93
图 7-27. 同步模式下读传输.....	93
图 8-1. PHY 功能模块图.....	95
图 8-2. LED 连接图.....	97

---

图 8-3. 100Base-TX LPI .....	98
图 9-1. EtherCAT 系统框图.....	143
图 9-2. EtherCAT 状态机 .....	148
图 9-3. EEPROM 布局.....	149
图 9-4. PDI 中断屏蔽和中断信号 .....	150
图 9-5. EtherCAT 中断屏蔽.....	151

# 表索引

表 1-1. GDSCN 存储映射表 .....	12
表 1-2. 有效数据的对齐 .....	16
表 2-1. 节能模式总结 .....	30
表 5-1. GPIO 配置表 .....	52
表 7-1. 4 线模式 .....	71
表 7-2. 6 线模式 .....	71
表 7-3. 8 线模式下的 OSPI .....	72
表 7-4. SPI 指令 .....	74
表 7-5. QSPI 指令 .....	74
表 7-6. OSPI 指令 .....	75
表 7-7. EXMC 引脚和描述 .....	91
表 9-1. EtherCAT 的 PDI .....	144
表 9-2. 通过写入确认 PDI 寄存器功能的受影响寄存器 .....	145
表 10-1. 修订历史 .....	215

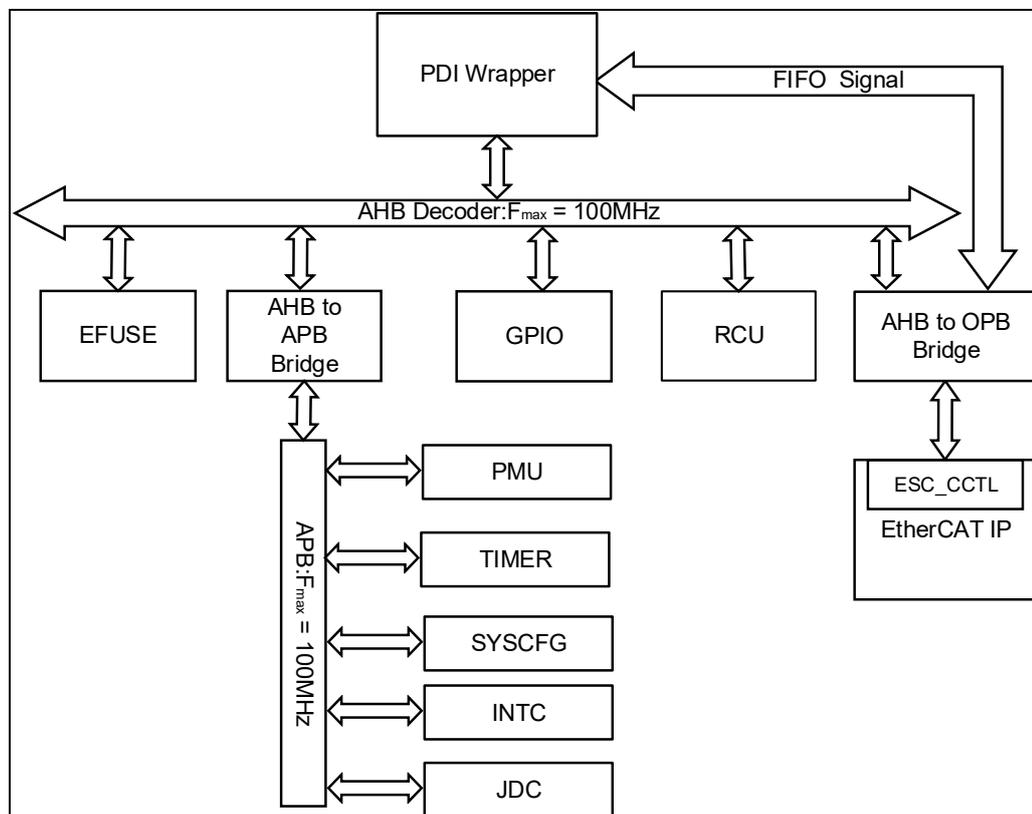
## 1. 系统和总线架构

### 1.1. 总线架构

ESC 的总线架构如下所示。基于 AMBA 5 AHB-LITE 的 AHB 矩阵是一个多层 AHB，它允许系统中一个主设备与多个从设备之间的并行访问路径。AHB 解码器上的一个主设备，包括 PDI 包装器的 AHB 总线。AHB 解码器由五个从设备组成，包括 AHB 到 OPB 桥，EFUSE，AHB 到 APB 桥，GPIO 和 RCU。

AHB 与包括一个 AHB 至 APB 桥接器在内的 AHB 外设相连，该桥接器在 AHB 解码器和一条 APB 总线之间提供完全同步的连接。这条 APB 总线与所有 APB 外设相连，包括电源管理单元（PMU）、定时器（TIMER）、系统配置（SYSCFG）和中断控制器（INTC）。

图 1-1. ESC 的总线架构



### 1.2. 存储器映射

本节将介绍 GDSCN 的存储器映射分布如下 [表 1-1. GDSCN 存储映射表](#) 所示

表 1-1. GDSCN 存储映射表

预先定义的地址空间	地址范围	外设
EtherCAT	0x0000	Type

预先定义的地址空间	地址范围	外设
	0x0001	Revision
	0x0002 - 0x0003	Build
	0x0004	FMMUs Supported
	0x0005	SyncManagers Supported
	0x0006	RAM Size
	0x0007	Port Descriptor
	0x0008 - 0x0009	ESC Features Supported
	0x0010 - 0x0011	Configured Station Address
	0x0012 - 0x0013	Configured Station Alias
	0x0013 - 0x001F	保留
	0x0020	Write Register Enable
	0x0021	Write Register Protection
	0x0022 - 0x002F	保留
	0x0030	ESC Write Enable
	0x0031	ESC Write Protection
	0x0032 - 0x003F	Reserved
	0x0040	ESC Reset ECAT
	0x0041	ESC Reset PDI
	0x0042 - 0x0FF	保留
	0x0100 - 0x0103	ESC DL Control
	0x0104 - 0x0107	保留
	0x0108 - 0x0109	Physical Read/Write Offset
	0x0110 - 0x0111	ESC DL Status
	0x0112 - 0x011F	保留
	0x0120 - 0x0121	AL Control
	0x0122 - 0x012F	保留
	0x0130 - 0x0131	AL Status
	0x0132 - 0x0133	保留
	0x0134 - 0x0135	AL Status Code
	0x0136 - 0x0137	Reserved
	0x0138	RUN LED Override
	0x0139	保留
	0x0140	PDI Control
	0x0141	ESC Configuration
	0x0142 - 0x0143	ASIC Configuration
	0x0144 - 0x0145	RESERVED Register
	0x0146 - 0x014F	保留
	0x0150	PDI Configuration
	0x0151	Sync/Latch PDI Configuration
	0x0152 - 0x0153	Extended PDI Configuration
	0x0154 - 0x01FF	保留

预先定义的地址空间	地址范围	外设
	0x0200 - 0x0201	ECAT Event Mask
	0x0202 - 0x0203	保留
	0x0204 - 0x0207	PDI AL Event Mask
	0x0208 - 0x0209	保留
	0x0210 - 0x0211	ECAT Event Request
	0x0212 - 0x021F	保留
	0x0220 - 0x0223	AL Event Request
	0x0223 - 0x022F	保留
	0x0300 - 0x0307	RX Error Counter
	0x0308 - 0x030B	Forwarded RX Error Counter
	0x030C	ECAT Processing Unit Error Counter
	0x030D	PDI Error Counter
	0x030E	PDI Error Code
	0x030F	保留
	0x0310 - 0x0313	Lost Link Counter
	0x0314 - 0x03FF	保留
	0x0400 - 0x0401	Watchdog Divider
	0x0410 - 0x0411	Watchdog Time PDI
	0x0420 - 0x0421	Watchdog Time Process Data
	0x0440 - 0x0441	Watchdog Status Process Data
	0x0442	Watchdog Counter Process Data
	0x0443	Watchdog Counter PDI
	0x0444 - 0x04FF	保留
	0x0500	EEPROM Configuration
	0x0501	EEPROM PDI Access State
	0x0502 - 0x0503	EEPROM Control/Status
	0x0504 - 0x0507	EEPROM Address
	0x0508 - 0x050B	EEPROM Data
	0x050C- 0x050F	保留
	0x0510 - 0x0511	MII Management Control/Status
	0x0512	PHY Address
	0x0513	PHY Register Address
	0x0514 - 0x0515	PHY DATA
	0x0516	MII Management ECAT Access State
	0x0517	MII Management PDI Access State
	0x0518 - 0x051B	PHY Port Status
	0x051C - 0x05FF	保留
	0x0600 - 0x06FF	FMMU
	0x0700 - 0x07FF	Reserved
	0x0800 - 0x087F	SyncManager
	0x0880 - 0x08FF	保留

预先定义的地址空间	地址范围	外设
	0x0900 - 0x09FF	Distributed Clocks (DC)
	0x0A00 - 0x0AFF	保留
	0x0E00 - 0x0E07	Product ID
	0x0E08 - 0x0E0F	Vendor ID
	0x0E10 - 0x0EFF	保留
	0x0F00 - 0x0F03	Digital I/O Output Data
	0x0F04 - 0x0F0F	保留
	0x0F10 - 0x0F17	General Purpose Outputs
	0x0F18 - 0x0F1F	General Purpose Inputs
	0x0F20 - 0x0F7F	保留
	0x0F80 - 0x0FFF	User RAM
	0x1000 - 0x2FFF	Process Data RAM
外设	0x3300 - 0x33FF	AHB2OPB Bridge
	0x3400 - 0x34FF	RCU
	0x3500 - 0x35FF	GPIO
	0x3600 - 0x36FF	EFUSE
	0x3700 - 0x37FF	PMU
	0x3800 - 0x38FF	TIMER
	0x3900 - 0x39FF	SYSCFG
	0x3A00 - 0x3AFF	INTC

### 1.3. AHB 直接 / 间接访问

GDSCN 可以通过 AHB 到 OPB 桥以三种方式直接或间接访问 ESC 寄存器和内核 PRAM：可直接访问的 ESC 寄存器、可间接访问的 ESC 内核寄存器和可间接访问的 ESC 内核 PRAM。该桥向上游侧提供 AHB 从接口，向下游侧提供 OPB 接口，其中上游 AHB 侧的频率比下游 ESC 内核快，并且时钟同步相位，具有 N:1（最大 N=16）的频率比。

#### 1.3.1. 直接 AHB 传输访问 ESC 控制器寄存器

直接访问 ESC 寄存器用于将数据 / 命令传输到间接访问 ESC 内核寄存器。当 AHB 总线访问地址范围为 0x0000-0x0FFF 时，将根据 ESC\_CCTL\_CMD 寄存器中的 CCTL\_RW 位启动对 ESC 内核的单个寄存器读/写操作。开始读取周期时，将 ESC\_CCTL\_CMD 寄存器中的 CCTL\_BUSY 位置 1，读取周期结束时，CCTL\_BUSY 被清 0，可以在 AHB 总线上读取有效数据。写入周期开始时，将 CCTL\_BUSY 位置 1，向总线上写入有效数据，写入周期结束时，CCTL\_BUSY 位被清 0，寄存器中的有效数据被写入。

#### 1.3.2. 间接传输访问 ESC 内核寄存器

GDSCN 可以通过 ESC\_CCTL\_DATA 和 ESC\_CCTL\_CMD 寄存器以间接方式访问 ESC 内核寄存器。当读取 ESC 内核寄存器时，需要执行以下步骤：首先将 ESC\_CCTL\_CMD 寄存器中 CCTL\_STOP 位置 1 用来将 CCTL\_BUSY 位清 0；将 ESC\_CCTL\_CMD 寄存器中的 CCTL\_RW

位置 1, 向 CCTL\_ADDR 位域中写入需要访问的寄存器地址, 并向 CCTL\_SIZE 位域中写入所需读取的字节数, 再将 CCTL\_BUSY 位置为 1。当 CCTL\_BUSY 位被清除时, 即可以从 ESC\_CCTL\_DATA 寄存器中读取数据。

在向 ESC 内核寄存器写入数据时, 需要执行以下步骤: 首先将 ESC\_CCTL\_CMD 寄存器中 CCTL\_STOP 位置 1 用来将 CCTL\_BUSY 位清 0; 将 ESC\_CCTL\_CMD 寄存器中的 CCTL\_RW 位清 0, 向 CCTL\_ADDR 位域中写入需要访问的寄存器地址, 并向 CCTL\_SIZE 位域中写入所需写入的字节数, 再将 CCTL\_BUSY 位置为 1。可一次将配置数据写入 ESC\_CCTL\_CMD 寄存器。写入周期的完成由 CCTL\_BUSY 位被清除为零来指示。

在上述读写操作中, 有效数据始终与 ESC\_CCTL\_DATA 寄存器的低位对齐。有效数据可以参考下表。

**表 1-2. 有效数据的对齐**

CCTL_SIZE	ESC CCTL_ADDR[1:0]	ESC CCTL_DATA valid bytes
1	00/01/10/11	[7:0]/ [15:8]/ [23:16]/ [31:24]
2	00/10	[15:0]/[31:16]
4	00	[31:0]

### 1.3.3. 间接传输访问 ESC 内核 PRAM

当通过 AHB 对内核 PRAM 发起读操作时, 写入 PRAM 起始地址和读长度到 ESC\_PRAM\_ALR 寄存器后, 将 PRAM\_BUSY\_READ 设置为 1, 模块开始发起多个 OPB 读操作, 从内核 PRAM 读取数据并写入 TX FIFO。所有 OPB 读操作完成后, 清除 PRAM\_BUSY\_READ。当数据从 ESC 内核传输到 TX FIFO 时, PRAM 读长度 PRAM\_LEN\_READ 和 PRAM 读地址 PRAM\_ADDR\_READ 会更新以显示进程。根据起始地址确定第一次读数据的有效字节。根据起始地址和操作长度确定最后一次读数据的有效字节。如果需要, 可以通过将 ESC\_PRAM\_CR[PRAM\_STOP\_READ]位设置为 1 来停止读命令。如果 OPB 读周期已经开始, 停止命令在当前读操作完成后生效。停止命令生效后, TX FIFO 中的数据会被清除。

通过 AHB 向 PRAM 写入时, 先写入 RAM 起始地址和写入长度到 ESC\_PRAM\_ALW 寄存器, 然后向 ESC\_PRAM\_CW 寄存器的 PRAM\_BUSY\_WRITE 位写入 1, 模块启动多个 OPB 写操作。从 RX FIFO 读取数据并写入内核 PRAM。所有 OPB 写操作完成后, 清除 PRAM\_BUSY\_WRITE。写操作支持等待机制, PRAM\_BUSY\_WRITE 位被设置为 1, 但是当 RX FIFO 为空时, OPB 模块不会立即启动 OPB 传输操作, 直到 RX FIFO 中有数据才会启动此操作。在每次 OPB 确认响应后, 它将检测 RX FIFO 的状态。如果 RX FIFO 为空, 它将进入等待状态并进入下一个传输操作, 直到有数据。当数据从 RX FIFO 传输到内核时, PRAM 写入长度 PRAM\_LEN\_WRITE 和 PRAM 写入地址 PRAM\_ADDR\_WRITE 将更新以显示进程。根据起始地址确定第一次数据写入的有效字节。根据起始地址和操作长度, 确定最后一次数据写入的有效字节。

如果需要, 可以通过将 PRAM\_STOP\_WRITE 位设置为 1 来停止写命令。如果 OPB 写周期开始, 停止命令在读取操作完成后生效。停止命令生效后, RX FIFO 中的数据将被清除。

## 1.4. 忙碌状态下的寄存器保护

如果 BUSY 相关的寄存器位，例如 CCTL\_BUSY、PRAM\_BUSY\_READ 和 PRAM\_BUSY\_WRITE，被设置为 1，您可以设置 BRP 位以防止寄存器被重写。当 CCTL\_BUSY 设置为 1 时，ESC\_CCTL\_DATA/ESC\_CCTL\_CMD 将受到 AHB 写操作的保护，防止被重写。当 PRAM\_BUSY\_READ 设置为 1 时，ESC\_PRAM\_ALR 将受到 AHB 写操作的保护，防止被重写。当 PRAM\_BUSY\_WRITE 设置为 1 时，ESC\_PRAM\_ALW 将受到 AHB 写操作的保护，防止被覆盖。

当 BRP 位设置为 1 时，当与 BUSY 相关的寄存器位设置为 1 时，用户对相应寄存器的 AHB 写操作将丢失，然后 ESC\_OPB\_CS 寄存器中的 WDLF 标志将设置为 1，并且当 WDIE 位设置为 1 时，将触发中断。

当 BRP 位设置为 0，当与 BUSY 相关的寄存器位设置为 1 时，用户对相应寄存器的 AHB 写操作将导致当前 OPB 传输错误，然后 ESC\_OPB\_CS 寄存器中的 WEF 标志将被设置为 1，并且当 WEIE 位设置为 1 时将触发中断。建议尽快处理以避免更多错误。

## 1.5. OPB 传输超时功能

OPB 传输超时功能可以通过将 ESC\_OPB\_CS 寄存器中的 TOEN 位设置为 1 来启用。超时间隔可以通过 ESC\_OPB\_CS 寄存器中的 TO\_CNT 位进行配置。当计数器超过 TO\_CNT 编程值时，ESC\_OPB\_CS 寄存器中的 TOF 标志将被设置为 1。如果 ESC\_OPB\_CS 寄存器中的 TOIE 位设置为 1，则会生成超时中断。超时中断响应处理：

如果使用 ESC CCTL 直接读写模式，当超时中断响应发生后，TOF 标志会被设置。直接终止传输以避免总线被占用。

如果以太网控制器(ESC)的通信控制列表(CCTL)在间接读写模式下使用，超时中断响应发生后，会设置 TOF 标志。可以通过写入 1 到 ESC\_CCTL\_STOP 位来停止这个操作。

当 PRAM 以间接读写模式使用时，超时中断响应发生后，会设置 TOF 标志。您可以通过写入 1 到 PRAM\_STOP\_WRITE/PRAM\_STOP\_READ 位来停止此操作。

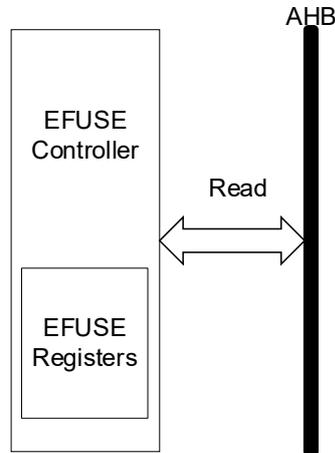
## 1.6. EFUSE 功能

EFUSE 控制器拥有 EFUSE 存储单元，用于存储系统参数。作为非易失性存储单元，一旦 EFUSE 存储单元的位被编程为 1，就不能恢复为 0。根据软件操作，EFUSE 控制器可以对系统参数中的所有位进行编程。

EFUSE 的主要目的如下：

- 一次性可编程非易失性电熔存储单元，为 32×8 位；
- EFUSE 中的所有位都不能从 1 变回 0；
- 只能通过相应的寄存器访问。

图 1-2. EFUSE 控制框图



## 1.7. EFUSE 寄存器定义

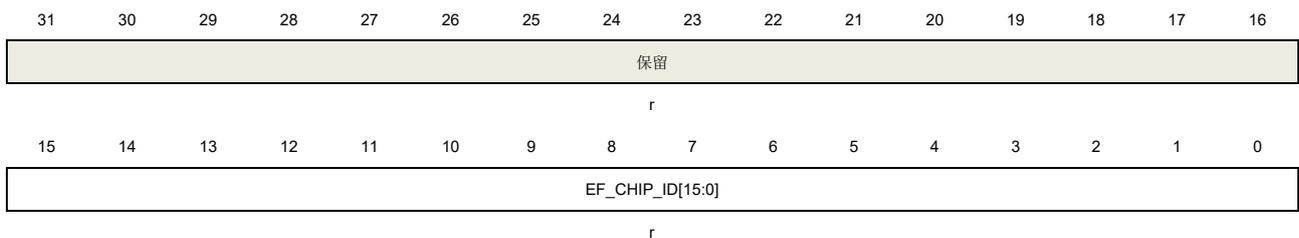
EFUSE 基地址: 0x3600

### 1.7.1. 芯片 ID 寄存器 (EF\_CHIP\_ID)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



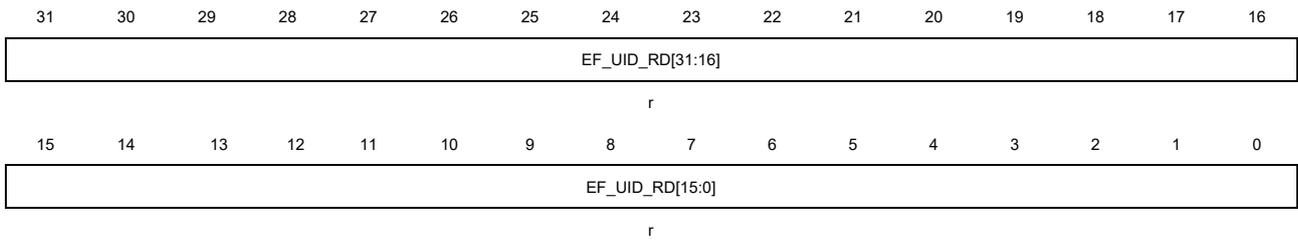
位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	EF_CHIP_ID[15:0]	读取 CHIP ID

### 1.7.2. EFUSE UID 读取寄存器 (EFUSE\_UID\_READ)

地址偏移: 0x1C+X\*4(X=0,1,2,3)

复位值: 0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:0	EF_UID_RD[31:0]	读取 ESC UID

## 1.8. ESC 内核控制器 (ESC\_CCTL)

ESC 内核控制器 (ESC\_CCTL) 的主要目的如下：

- 配置以太网控制器 (ESC) 内核寄存器的间接传输访问；
- 配置 ESC 内核 PRAM 的间接传输访问。

## 1.9. ESC 内核控制器寄存器定义

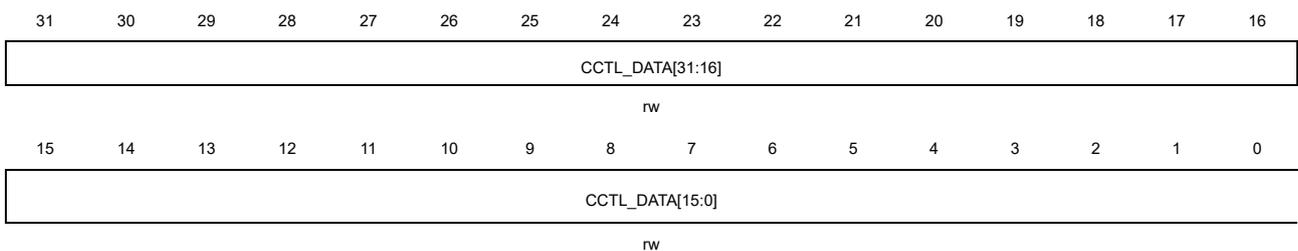
ESC 内核控制寄存器基地址：0x3300

### 1.9.1. ESC CCTL 数据寄存器(ESC\_CCTL\_DATA)

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



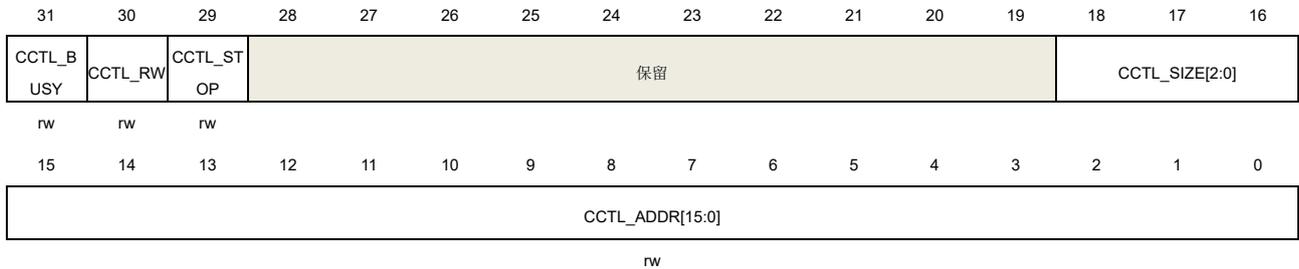
位/位域	名称	描述
31:0	CCTL_DATA[31:0]	ESC CCTL 数据 此字段指示从 ESC 核心读取或写入的值。读取或写入取决于 ESC_CCTL_CMD[CCTL_RW]位。如果 CCTL_RW 位为 1，则此值为从 ESC 核心读取的数据；如果 CCTL_RW 位为 0，则此值为写入 ESC 核心的数据。 此字段的最低位始终指示已写入或读取的有效数据。

### 1.9.2. ESC CCTL 命令寄存器 (ESC\_CCTL\_CMD)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31	CCTL_BUSY	CCTL 忙。如果将该位写为零，则无效。 0: 没有进行读写 (根据 CCTL_RW) 操作。 1: 正在进行读写操作。 <b>注意:</b> 当读写操作完成时，此位将被清除。然后，HOST 可以从 ESC_CCTL_DATA 寄存器读取有效数据或写入有效数据。当此位为 0 时，需要修改 ESC_CCTL_CMD 和 ESC_CCTL_DATA 寄存器。
30	CCTL_RW	读写操作 0: 写操作 1: 读操作
29	CCTL_STOP	停止读操作或写操作。如果将该位写为零，则无效 0: 无效果 1: 停止 ESC 核心寄存器的读或写操作 <b>注意:</b> 当 CCTL_BUSY 被清除时，CCTL_STOP 将被清除
28:19	保留	必须保持在复位值。
18:16	CCTL_SIZE[2:0]	此字段指定 ESC CCTL 大小 (字节)。1、2 和 4 是有效的，其他值是无效的。更多细节请参阅 <a href="#">表 1-2. 有效数据的对齐</a>
15:0	CCTL_ADDR[15:0]	此字段指定将被访问的 ESC 核心寄存器的地址。

### 1.9.3. ESC PRAM FIFO 数据读取寄存器 (ESC\_PRAM\_FIFO\_DR)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



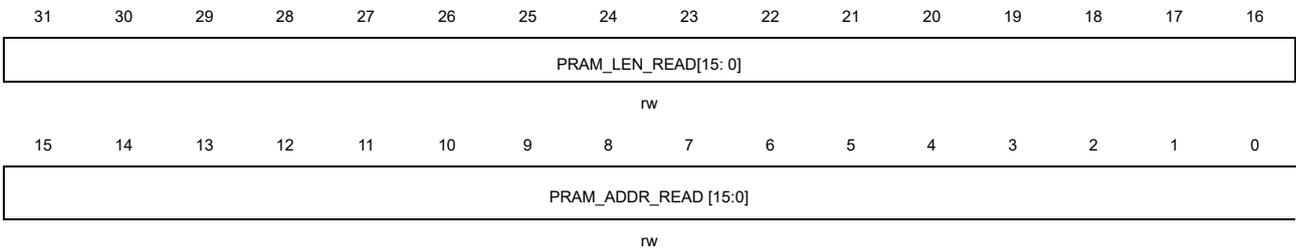


Bits	Fields	Descriptions
31:0	PRAM_FIFO_DATA_READ[31:0]	从 ESC PRAM 读取的数据。 数据的有效值是根据起始地址和传输长度来确定的。

### 1.9.4. ESC PRAM 地址和长度读取寄存器 (ESC\_PRAM\_ALR)

地址偏移: 0x14  
 复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:16	PRAM_LEN_READ[15: 0]	从 ESC PRAM 读取的数据长度, 以字节为单位。当数据被读入 FIFO 时, 此字段会递减。 <b>注意:</b> 当 PRAM_BUSY_READ 为 1 时, 此字段不能被修改。
15:0	PRAM_ADDR_READ [15:0]	ESC PRAM 数据读取地址。当数据被读入 FIFO 时, 此字段会递增。 <b>注意:</b> 当 PRAM_BUSY_READ 为 1 时, 此字段不能被修改。

### 1.9.5. ESC PRAM 命令读取寄存器 (ESC\_PRAM\_CR)

地址偏移: 0x18  
 复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
------	----	----

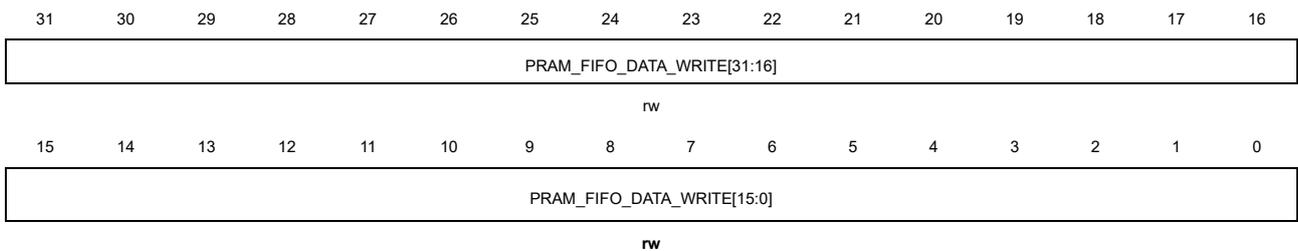
31	PRAM_BUSY_READ	PRAM 是否正在被读取。如果该位被写为零，则无效。 0: 没有 PRAM 读取操作 1: PRAM 正在被读取 <b>注意:</b> 当读取操作完成时，此位会被清除。
30	PRAM_STOP_READ	停止 PRAM 读取操作。如果该位被写为零，则无效。 0: 无效果 1: 停止 PRAM 读取操作 <b>注意:</b> 在此位被设置为 1 之后，PRAM_BUSY_READ 将被清除，并且 RX FIFO 将被复位。然后此位将自动清除。
29:13	保留	必须保持复位值。
12:8	PRAM_DATA_CNT_READ[4:0]	PRAM 数据读取有效计数 当数据从 PRAM 读入 RX FIFO 时，此计数增加；当整个 DWORD 大小的数据从 RX FIFO 读出时，此计数减少。
7:1	保留	必须保持复位值。
0	PRAM_VALID_DATA_READ	PRAM 有效数据读取 0: 没有有效数据可读。 1: 有有效数据可读。

### 1.9.6. ESC PRAM FIFO 数据写寄存器 (ESC\_PRAM\_FIFO\_DW)

地址偏移: 0x20

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:0	PRAM_FIFO_DATA_WRITE[31:0]	写入 ESC PRAM 的数据。 数据的有效值是根据起始地址和传输长度来确定的。

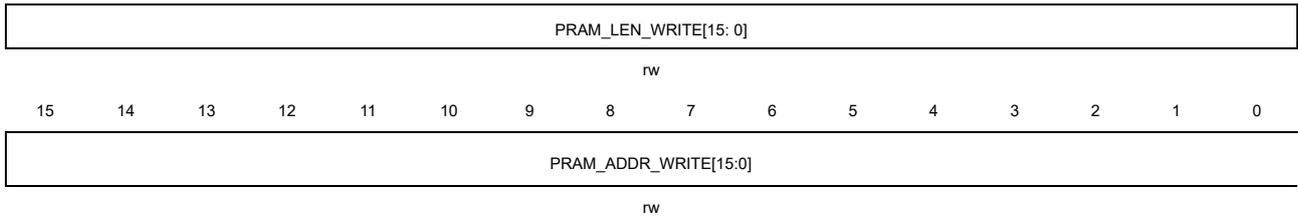
### 1.9.7. ESC PRAM 地址和长度写寄存器 (ESC\_PRAM\_ALW)

地址偏移: 0x24

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。





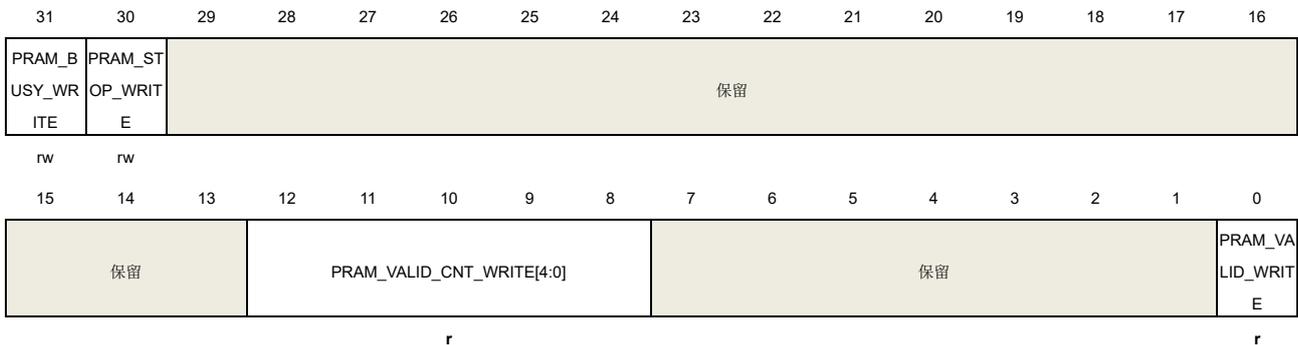
位/位域	名称	描述
31:16	PRAM_LEN_WRITE[15: 0]	写入 ESC PRAM 的数据长度，以字节为单位。当数据被写入 FIFO 时，此字段会递减。 <b>注意：</b> 当 PRAM_BUSY_WRITE 为 1 时，此字段不能被修改。
15:0	PRAM_ADDR_WRITE[15:0]	ESC PRAM 数据写入地址。当数据从 FIFO 中读取时，此字段会递增。 <b>注意：</b> 当 PRAM_BUSY_WRITE 为 1 时，此字段不能被修改。

### 1.9.8. ESC PRAM 命令写寄存器 (ESC\_PRAM\_CW)

地址偏移：0x28

复位值：0x0000 1001

该寄存器只能按字（32位）访问。



位/位域	名称	描述
31	PRAM_BUSY_WRITE	PRAM 是否正在被写入。如果该位被写为零，则无效。 0: 没有 PRAM 写入操作 1: PRAM 正在被写入 <b>注意：</b> 当写入操作完成时，此位会被清除。
30	PRAM_STOP_WRITE	停止 PRAM 写操作。如果该位写为 0，则没有效果 0: 无效果 1: 停止 PRAM 写操作 <b>注意：</b> 当该位被设置为 1 后，PRAM_BUSY_WRITE 会被清除，并且 TX FIFO 会被复位。然后该位会自动清除。
29:13	保留	必须保持复位值。
12:8	PRAM_VALID_CNT_WRITE[4:0]	PRAM 数据写入有效计数 当数据从 TX FIFO 读取到 PRAM 时，此计数增加；当整个 DWORD 大小的数据写

入 PRAM 时，此计数减少。

7:1	保留	必须保持复位值。
0	PRAM_VALID_DATA _WRITE	PRAM 有效数据写入 0: 没有有效数据要写入 1: 有有效数据要写入

### 1.9.9. ESC OPB 控制和状态寄存器 (ESC\_OPB\_CS)

地址偏移: 0x30

复位值: 0x0000 0479

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RAAF	RAAIE	WDLF	WDLIE	TOF	TOIE	WEF	WEIE	ESC CCTLVIF	ESC CCTLVIE	保留					
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					BRP	TO_CNT[8:0]								TOEN	
					r/w	r/w								r/w	

位/位域	名称	描述
31	RAAF	保留地址访问标志。当访问的地址位于保留的地址段内时，此标志会被触发。通过向其写入 1，可以清除此标志。 0: 访问的地址是一个有效的地址段 1: 访问的地址是一个保留的地址段
30	RAAIE	保留地址访问中断使能 0: 中断被禁止 1: 每当 RAAF 位被设置时，就会发生中断
29	WDLF	写数据丢失标志。当 BRP 启用时，如果忙标志高，则 AHB 写操作会导致数据丢失，此标志将被触发。此标志可以通过写入 1 来清除。 0: 没有数据丢失 1: 此次传输的数据已丢失
28	WDLIE	写数据丢失中断使能 0: 中断被禁止 1: 每当 RAAF 位被设置时，就会发生中断
27	TOF	超时标志，当单次传输时间超过设定值时，此标志将被触发。此标志可以通过写入 1 来清除。 0: 没有发生 ESC 核心超时传输 1: 发生了 ESC 核心超时传输
26	TOIE	超时中断使能 0: 中断被禁止

		1: 每当 RAAF 位被设置时, 就会发生中断
25	WEF	写错误标志。当 BRP 禁用时, 如果忙标志高, AHB 写操作将导致当前 OPB 传输错误, 此标志将被触发。此标志可以通过写入 1 来清除
24	WEIE	写错误中断使能。 0: 中断被禁止 1: 每当 WEF 位被设置时, 就会发生中断
23	ESC CCTLIVF	CCTL_SIZE 和 CCTL_ADDR 非法值标志。写入寄存器的 CCTL_SIZE 和 CCTL_ADDR 值不符合要求, 导致此次传输无法进行, 此标志将被触发。此标志可以通过写入 1 来清除。
22	ESC CCTLIVIE	CCTL_SIZE 和 CCTL_ADDR 非法值中断使能。 0: 中断被禁止 1: 每当 IESC CCTLVIF 位被设置时, 就会发生中断
21:11	保留	必须保持复位值。
10	BRP	当忙标志置位时, 防止寄存器变化 1: 保护生效 0: 保护不生效
9:1	TO_CNT[8:0]	超时计数器 此字段表示在未收到 ACK 响应的情况下传输超时的位数; 程序员数据不能小于手册推荐 (60) 的最小值。
0	TOEN	超时使能 此字段表示该功能是否生效 1: 启用超时特性 0: 禁用超时特性。

## 1.10. 系统配置控制器 (SYSCFG)

系统配置控制器 (SYSCFG) 的主要目的如下:

- 配置微控制器HCLK频率比例;
- 配置SPI扩展模式;
- 提供芯片ID和版本。

## 1.11. 系统配置寄存器定义

系统配置基地址: 0x3900

### 1.11.1. 系统配置寄存器 0 (SYSCFG\_CFG0)

地址偏移: 0x00

复位值：0x001F 0000

该寄存器只能按字（32位）访问。



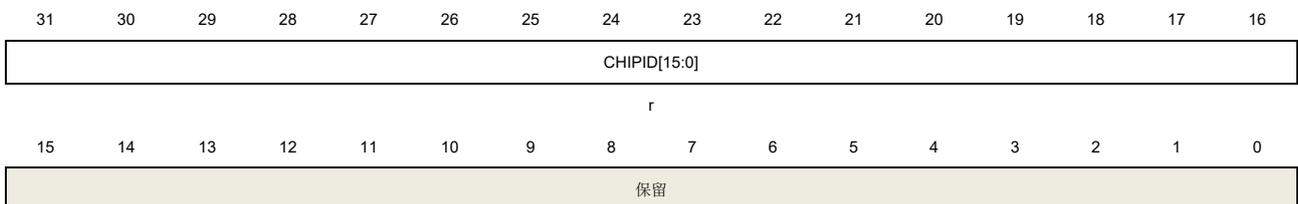
位/位域	名称	描述
31:21	保留	必须保持复位值。
20:16	MCUFREQ[4:0]	MCU HCLK 频率比率 00000: MCU_HCLK_FREQ >=100MHz 00001: 100MHz/2 <= MCU_HCLK_FREQ < 100MHz 00010: 100MHz/3 <= MCU_HCLK_FREQ < 100MHz/2 .... 11111: 100MHz/32 <= MCU_HCLK_FREQ < 100MHz/31
15:3	保留	必须保持复位值。
2	SPIEXTMOD	PDI 类型与 SPI 结合 0: GPIO 1: MII (OSPI: 不输出 clk_25M; 其他: 输出 clk_25M)
1:0	保留	必须保持复位值。

### 1.11.2. 系统配置芯片 ID 寄存器（SYSCFG\_CHIPID）

地址偏移：0x90

复位值：0x0000 0000

该寄存器只能按字（32位）访问。



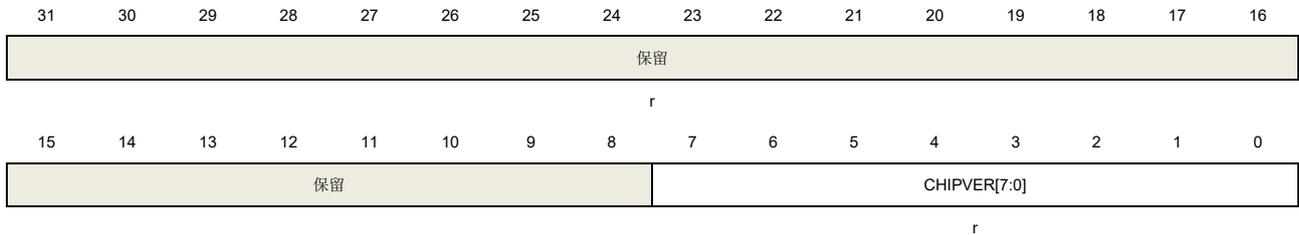
位/位域	名称	描述
31:16	CHIPID[15:0]	芯片 ID
15:0	保留	必须保持复位值。

### 1.11.3. 系统配置芯片版本寄存器 (SYSCFG\_CHIPVER)

地址偏移: 0x94

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:8	保留	必须保持复位值。
15:0	CHIPVER[7:0]	芯片版本

### 1.11.4. 系统配置保留寄存器 (SYSCFG\_RESERVED)

地址偏移: 0xF0

复位值: 0x0000 0000

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	RESERVED	保留寄存器

## 2. 电源管理单元 (PMU)

### 2.1. 简介

功耗设计是 GDSCN 系列产品比较注重的问题之一。电源管理单元 (PMU) 提供了四种设备级别的节能模式和三种模块级别的节能模式，设备级别的节能模式包括 MOD0、MOD1、MOD2 和 MOD3，模块级别的节能模式包括 EtherCAT 时钟管理、PHY 电源管理和 LED 引脚电源管理。这些模式降低了功耗，并允许应用程序在设备运行时间、速度和功耗之间的冲突需求之间实现最佳权衡。PMU 还支持唤醒事件检测和电源管理事件 (PME) 通知。

### 2.2. 主要特征

- EtherCAT 时钟管理；
- PHY 功耗管理，包括 PHY A 和 B 的能量检测(ED)掉电管理和通用掉电管理；
- LED 引脚掉电管理；
- 四种设备级别的节能模式，包括 MOD0、MOD1、MOD2 和 MOD3；
- PHY 唤醒事件检测，包括 PHY ED 上电唤醒和 PHY LAN 魔术包唤醒；
- 中断唤醒通知。

### 2.3. 功能说明

#### 2.3.1. 设备就绪

PMU\_CTL0 寄存器中的 RDY 位可以指示设备是否就绪。主机可以读取此位以获取设备的就绪状态。

- 在上电后，如果 EtherCAT 设备复位或数字复位，并且 RDY 位置位，这表明设备已成功读取 EEPROM 中的内容，并根据读取的内容进行了配置；
- 将 RCU\_RSTCFG 寄存器中的 ESCRST 位置为 1，会复位 EtherCAT 内核，从而使 EtherCAT 重新读取 EEPROM 并根据读取的配置内容重新配置设备。在这个过程中，RDY 位会短暂地变为低电平；
- 当设备进入节能模式 MOD1、MOD2 或 MOD3 时，RDY 位将变为低电平。一旦设备从节能模式唤醒回到 MOD0，并且 PLL 稳定后，RDY 位将重新设置为高电平。

**注意：**设备支持电压检测。当供电电压达到预定值时，RDY 位将置位。

#### 2.3.2. EFUSE 供电

VDDIO 是一个专为 EFUSE 写入设计的电源引脚，当使用内部 LDO 时，需要一个合适的电源电压（详见数据手册的详细描述），如果设置了 EFUSE\_LDO\_BYPASS 位，则需要 2.5V 的 LDO 电压。

### 2.3.3. PHY 唤醒事件检测

该设备支持两种类型的 PHY 唤醒事件检测：

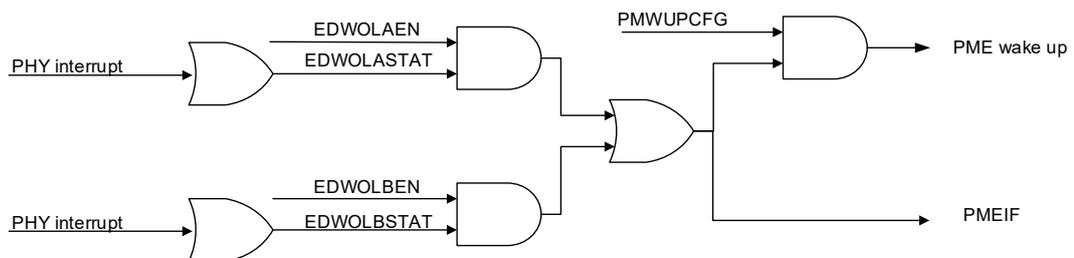
- PHY ED 上电唤醒事件；
- PHY LAN 魔术包唤醒。

### 2.3.4. PME 唤醒通知

PME 模块负责处理 PMU\_CTL 寄存器中的 EDWOLASTAT 和 EDWOLBSTAT 位的锁存功能。可以参考[图 2-1. PME 中断待处理](#)，了解 PME 中断控制的逻辑。如果置位了 EDWOLAEN 或 EDWOLBEN，当端口 A 或 B 的 PHY 发生能量检测 / WoL 事件时，中断状态寄存器中的 PMEIF 位将被置位。

当置位 PMWUPCFG 时，PME 事件可以在某些设备级节能模式下自动唤醒系统。

图 2-1. PME 中断待处理



### 2.3.5. 模块级节能模式

设备支持三种类型的模块级节能模式：

- PMU\_CTL0 寄存器中的 ECATCLKDIS 位可以用来禁用 EtherCAT 内核时钟；
- PHY 电源管理：
  - PHYA 和 PHYB 掉电管理，支持自动能量检测掉电管理；
  - 通用掉电管理。
- LED 输出管理：
  - PMU\_CTL0 寄存器中的 LEDOUTDIS 位可以用来禁用 LED 输出；
  - PMU\_CTL0 寄存器中 LEDMODCFG 位可以用来配置 LED 的工作模式（仅当 LEDOUTDIS 置位时生效）；
  - PMU\_CTL0 寄存器中 LEDINACT 位可用于配置 LED 在推挽模式下工作时的非活动状态（仅当 LEDOUTDIS 置位时才有效）。

### 2.3.6. 设备级节能模式

设备级复位后，设备以全功能运行，所有时钟都处于激活状态。用户可以通过关闭未使用功能的时钟来实现更低的功耗。此外，还提供了四种设备级节能模式，以实现更低的功耗，它们是

MOD0、MOD1、MOD2 和 MOD3。

### MOD0

设备级复位后，设备以 MOD0 模式运行，并处于全功能状态，所有时钟都处于激活状态。

### MOD1

在 MOD1 模式下，设备将禁用所有来自 PLL 时钟的衍生时钟。如果通过 PHY 或外部供电，网络时钟仍然保持启用状态。晶体振荡器和 PLL 仍然保持启用。这种模式可以通过手动或自动方式退出。

这种模式适用于 PHY 的通用掉电管理、PHY 的 WoL（局域网唤醒）模式以及 PHY 的能量监测掉电管理。

### MOD2

在 MOD2 模式下，设备将禁用所有来自 PLL 时钟的衍生时钟。如果通过 PHY 或外部供电，网络时钟仍然保持启用状态。允许禁用 PLL（如果两个 PHY 都处于 ED 或通用掉电管理，PLL 将被禁用）。XTAL 和 PLL 仍然保持启用状态。这种模式可以通过手动或自动方式退出。

MOD2 适用于 PHY 的通用掉电管理、PHY 的 WoL（局域网唤醒）模式以及 PHY 的能量监测掉电管理。

### MOD3

在 MOD3 模式下，设备将禁用所有由 PLL 时钟衍生的时钟。PLL 被禁用。外部网络时钟被关闭。晶体振荡器被禁用。这种模式只能手动退出。

MOD3 适用于 PHY 的通用掉电管理。

在设置此电源状态之前，主设备应置位 POWERDOWN 位。将 PHY 置于通用掉电模式。

表 2-1. 节能模式总结

Mode	PLL	System clocks	Network clocks	XTAL
MOD0	ON	ON	usable	ON
MOD1	ON	OFF	usable	ON
MOD2	OFF	OFF	usable	ON
MOD3	OFF	OFF	OFF	OFF

#### 2.3.7. 进入设备级节能模式

要从 MOD0 转换到 MOD1、MOD2 或 MOD3，可以遵循以下步骤进行：

1. 配置 PMWUPCFG 位；
2. 配置 PHY 唤醒检测，关于 PHY 唤醒检测可以参考 [PHY 唤醒事件检测](#)；
3. 配置 PHY 唤醒通知，关于 PHY 唤醒通知可以参考 [PME 唤醒通知](#)；
4. 确保设备已成功进入节能模式（确保无需发送数据包，接收器已禁用等）；
5. 设置 PMSLPEN 位。

**注意:**

- 进入省电模式后，寄存器 PMU\_CTL0 中的 RDY 位将被设置为低电平；
- 进入省电模式后，主机接口将无效。

### 2.3.8. 退出设备级别的节能模式

设备级别的节能模式可以手动或自动退出。

如果设置了 PMWUPCFG 位，则启用了 PME 唤醒功能，可能会自动唤醒 PME。关于 PME 唤醒，可以参考 [PME 唤醒通知](#)。

主机可以通过如下操作手动唤醒设备：

- 对设备执行 EXMC 写操作。尽管在设备被唤醒并执行读操作之前，所有的写操作都被忽略，但主机仍应指示写入 PMU\_PDIREFVAL 寄存器。在设备被唤醒之前，不应尝试写入任何其他地址；
- 对设备执行 SPI/SQI 周期（CS 低，SCK 高）。尽管在设备被唤醒之前，所有的读写操作都被忽略，但主机仍应通过读取 PMU\_PDIREFVAL 寄存器来指示唤醒设备。在设备被唤醒之前，不应尝试读写任何其他地址。

**注意:**

- 通过读取 PMU\_PDIREFVAL 寄存器可以确定主接口的工作状态。一旦读取到正确的值，主接口将进入就绪状态。然后，RDY 位将指示设备完全唤醒；
- 在自动或手动唤醒后，一旦设备返回到 MOD0 并且 PLL 重新稳定，设备就绪（RDY）位就会被设置，PMMODCFG 和 PMSLPEN 将被清除（置为 0）；
- 如果一切正常，设备的唤醒时间应该小于 2 毫秒。

## 2.4. 寄存器定义

PMU 基地址: 0x0000 3700

### 2.4.1. 控制寄存器 0 (PMU\_CTL0)

地址偏移: 0x00

复位值: 0x0000 C000

该寄存器可以按字 (32 位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMMODCFG[1:0]		保留		PMWUPC	LEDOUT	LEDMOD	LEDINAC	保留		ECATCL	保留			EDWOLB	EDWOLA
rw				FG	DIS	CFG	T			KDIS				STAT	STAT
rw				rw	rw	rw	rw			rw				r_w1	r_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDWOLB	EDWOLB	保留												RDY	
EN	EN														
rw	rw													r	

位/位域	名称	描述
31:30	PMMODCFG[1:0]	电源管理模式配置位 当 PMSLPEN 被设置时, 这些位可以用来配置电源管理模式。这些位在设备唤醒时会被清除。 00: MOD0 01: MOD1 10: MOD2 11: MOD3
29:28	保留	必须保持复位值。
27	PMWUPCFG	电源管理唤醒模式配置位 0: 由主机唤醒 1: 由 PME 或者主机唤醒
26	LEDOUTDIS	LEDs 输出失能位 当此位被置位时, LEDs 的输出将被失能。在漏极开路/源极开路模式下工作时, LEDs 不会被驱动; 在推挽模式下工作时, LEDs 仍将被驱动, 但 LEDs 的状态将变为非激活状态。 0: LEDs 输出被启用 1: LEDs 输出被禁用
25	LEDMODCFG	LEDs 工作模式配置位 (仅当 LEDOUTDIS 置位时生效) 0: LEDs 的工作模式为漏极开路/源极开路 1: LEDs 的工作模式为推挽
24	LEDINACT	推挽模式下 LEDs 非激活状态配置位 (仅当 LEDOUTDIS 置位时生效)

		0: 0 表示非激活状态 1: 1 表示非激活状态
23:22	保留	必须保持复位值。
21	ECATCLKDIS	EtherCAT 内核时钟失能位 如果 ECATCLKDIS 被置位，EtherCAT 内核时钟将被禁用。要设置这个位需要连续写入一两次。写入 0 将清除计数。 0: 启用 EtherCAT 内核时钟 1: 禁用 EtherCAT 内核时钟
20:18	保留	必须保持复位值。
17	EDWOLBSTAT	能量检测/ WoL 端口 B 状态位 要清除这个位，首先需要清除 PHY 上的事件。 0: 端口 B PHY 未发生能量检测/ WoL 事件 1: 端口 B PHY 发生能量检测/ WoL 事件
16	EDWOLASTAT	能量检测/ WoL 端口 A 状态位 要清除这个位，首先需要清除 PHY 上的事件。 0: 端口 A PHY 未发生能量检测/ WoL 事件 1: 端口 A PHY 发生能量检测/ WoL 事件
15	EDWOLBEN	能量检测/ WoL 端口 B 使能位 当能量检测/ WoL 事件发生在端口 B PHY 上并且该位被置位时，INTC_FLAG 寄存器中的 PMEIF 位将被设置。 0: 关闭能量检测/ WoL 端口 B 1: 使能量检测/ WoL 端口 B
14	EDWOLAEN	能量检测/ WoL 端口 A 使能位 当能量检测/ WoL 事件发生在端口 A PHY 上并且该位被置位时，INTC_FLAG 寄存器中的 PMEIF 位将被设置。 0: 关闭能量检测/ WoL 端口 A 1: 使能量检测/ WoL 端口 A
13:1	保留	必须保持复位值。
0	RDY	设备就绪位 这个位指示设备是否就绪。主机可以在上电、EtherCAT 设备复位、模块复位、数字复位或退出节能模式后读取这个位来获取设备的就绪状态。 0: 设备未就绪 1: 设备已就绪 <b>注意:</b> <ul style="list-style-type: none"> <li>■ 该位的上升沿将设置 INTC_FLAG 寄存器中的 READYIF 位，并可以触发一个中断；</li> <li>■ 当这个位被清除时，除了 PMU_CTL、PMU_PDIVAL 和 RCU_RSTCFG 寄存器之外，对任何内部资源的读取访问是被禁止的；</li> <li>■ 在该位被置位之前，对任何地址的写操作都是无效的。</li> </ul>

### 2.4.2. 控制寄存器 1 (PMU\_CTL1)

地址偏移: 0x04

复位值: 0x0000 0000

该寄存器可以按字 (32 位) 访问。



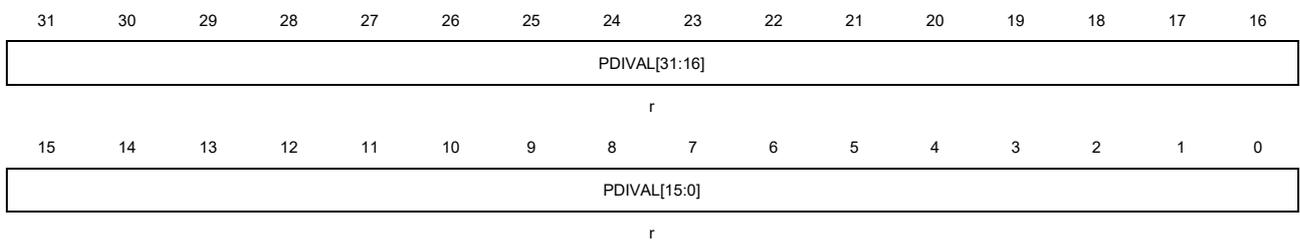
位/位域	名称	描述
31:1	保留	必须保持复位值。
0	PMSLPEN	电源管理睡眠模式使能位 当 PMSLPEN 置位时, 设备将进入由 PMMODCFG 位配置的电源管理模式。 0: 禁用电源管理睡眠模式 1: 启用电源管理睡眠模式 <b>注意:</b> <ul style="list-style-type: none"> <li>■ 当设备唤醒时, 该位将被清除;</li> <li>■ 当 PMMODCFG 为 0b00 时, 不应设置这个位, 尽管硬件不会阻止写操作。</li> </ul>

### 2.4.3. 数据接口参考值寄存器 (PMU\_PDIREFVAL)

地址偏移: 0x1C

复位值: 0x7654 3210

该寄存器可以按字 (32 位) 访问。



位/位域	名称	描述
31:0	PDIVAL[31:0]	当 PDI 就绪时, 读取这个寄存器将返回复位值, 否则, 将返回其他无效值 (非复位值)。用于 PDI 接口测试。

## 3. 复位和时钟单元（RCU）

### 3.1. 复位控制单元

#### 3.1.1. 简介

GDSCN复位控制包括两种控制方式：系统复位和模块复位。系统复位包括上电复位（POR）、外部引脚复位（RSTN）和EtherCAT系统复位，可以复位设备中的所有电路。模块复位包括数字复位、PHY复位和EtherCAT内核复位，可以复位相应的模块。

#### 3.1.2. 主要特征

- 系统复位，复位设备中的所有电路。
- 多模块复位，复位PHY之外的数字电路。
- 单模块复位，复位以太网内核和PHY。

#### 3.1.3. 功能说明

##### 系统复位

系统复位可以复位整个设备，包括上电复位（POR）、外部引脚复位（RSTN）和以太网系统级复位，具体描述如下：

**上电复位：**当设备刚上电或者电源断开后重上电时，发生上电复位。

**RSTN引脚复位：**将RSTN输入引脚驱动为低电平可启动外部引脚复位。

**EtherCAT系统复位：**EtherCAT系统复位是由三个连续的特殊帧/命令序列启动。

##### 模块复位

模块复位会影响一个或多个模块，并可以为各模块生成复位信号，描述如下：

**多模块复位：**通过设置配置寄存器（RCU\_RSTCFG）的DRST位来执行数字复位。数字复位会复位除PHY之外的所有设备子模块。

**单模块复位：**单模块复位仅复位指定的模块。单模块复位不会锁定配置引脚，包括端口A的PHY复位、端口B的PHY复位和EtherCAT控制器复位。单模块复位描述如下：

可通过置位复位配置寄存器（RCU\_RSTCFG）中的PHYARST位或置位PHY控制寄存器（PHY\_MII\_CTL）中的MR\_MAIN\_REST位实现端口A的PHY复位。端口A的PHY复位后，PHYARST位和软复位Bit会自动清除。该复位不会影响设备的其他模块。可以通过复位配置寄存器（RCU\_RSTCFG）中的PHYARST位或PHY控制寄存器（PHY\_MII\_CTL）中的MR\_MAIN\_REST位是否清除来检查端口A的PHY复位是否完成。

可通过置位复位配置寄存器（RCU\_RSTCFG）中的PHYBRST位或置位PHY控制寄存器

(PHY\_MII\_CTL) 中的MR\_MAIN\_REST位实现端口B的PHY复位。端口B的PHY复位后，PHYBRST位和软复位Bit会自动清除。该复位不会影响设备的其他模块。可以通过复位配置寄存器 (RCU\_RSTCFG) 中的PHYBRST位或PHY控制寄存器 (PHY\_MII\_CTL) 中的MR\_MAIN\_REST位是否清除来检查端口B的PHY复位是否完成。

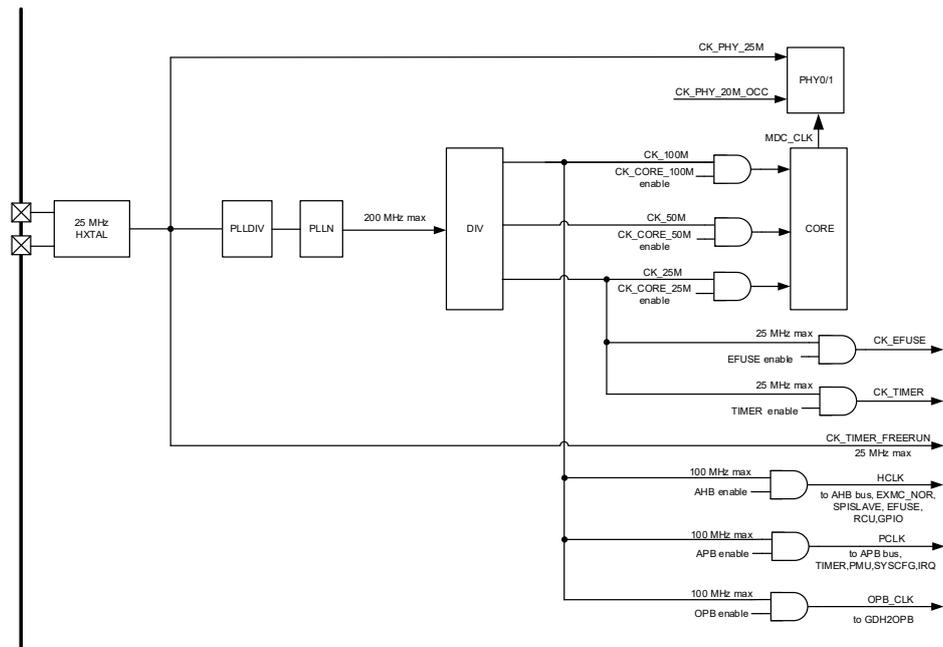
可以通过清除配置寄存器(RCU\_RSTCFG)中的ESCRST位实现EtherCAT控制器的单独复位，这将复位EtherCAT内核及其寄存器。

## 3.2. 时钟控制单元 (CCTL)

### 3.2.1. 简介

EtherCAT时钟控制单元主要由外部高速晶体振荡器 (HXTAL) 和锁相环 (PLL) 组成。时钟通常由25MHz无源晶体振荡器的OSCIN和OSCOU提供，或者由单端25MHz时钟源驱动器的OSCIN引脚提供。

图 3-1. 时钟树



### 3.2.2. 主要特征

- 25 MHz高速晶振振荡器 (HXTAL)。
- 锁相环 (PLL)

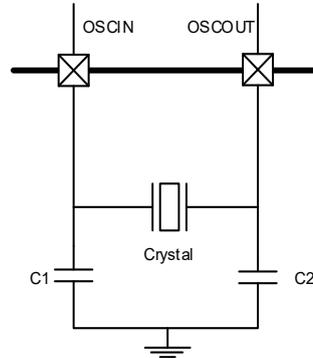
### 3.2.3. 功能说明

#### 高速晶体振荡器 (HXTAL)

系统需要一个固定频率为25 MHz的时钟源，供内部时钟振荡器和锁相环使用。通常是通

过将25 MHz晶体振荡器连接到芯片的OSCIN和OSCOUT引脚来提供。如果选择使用单端25 MHz时钟源，也可以通过驱动OSCIN输入引脚来提供这个时钟。如果选择了单端时钟源，为了设备正常工作，时钟输入必须持续运行。省电模式允许振荡器或外部时钟输入暂停。

图 3-2. HXTAL 时钟源



时钟配置寄存器（RCU\_CLKCFG）中的HXTALSTB位指示高速外部晶振是否稳定。在启动时，直到这一位被硬件置‘1’，时钟才被释放出来。这个特定的延迟时间被称为振荡器的启动时间。此时，HXTAL时钟可以被直接用作系统时钟源或者PLL输入时钟。

### 锁相环

PLL输入是一个25MHz的HXTAL时钟，经过PLLN进行8倍频后，得到CK\_PLL（200MHz）。CK\_PLL时钟通过PLLDIV进行2/4/8分频，得到100MHz、50MHz、25MHz时钟用于EtherCAT内核且相关的时钟可以在内核使能寄存器（RCU\_COREEN）中打开。相应的模块时钟可以在AHB使能寄存器（RCU\_AHBEN）、APB使能寄存器（RCU\_APBEN）和内核使能寄存器（RCU\_COREEN）中打开。

## 3.3. RCU 寄存器

RCU基地址：0x3400

### 3.3.1. AHB 使能寄存器（RCU\_AHBEN）

地址偏移：0x00

复位值：0x0000 000F

该寄存器可以按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												EFUSEE	EFUSEF	GPIOEN	OPBEN
												N	UNEN	rw	rw
												rw	rw	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3	EFUSEEN	EFUSE时钟使能 由软件置位或复位 0: 关闭 EFUSE 时钟 1: 开启 EFUSE 时钟
2	EFUSEFUNEN	EFUSE功能时钟使能 由软件置位或复位 0: 关闭EFUSE功能时钟 1: 开启 EFUSE 功能时钟
1	GPIOEN	GPIO 时钟使能 由软件置位或复位 0: 关闭 GPIO 时钟 1: 开启 GPIO 时钟
0	OPBEN	OPB 时钟使能 由软件置位或复位 0: 关闭 OPB 时钟 1: 开启 OPB 时钟

### 3.3.2. APB 使能寄存器 (RCU\_APBEN)

地址偏移: 0x04

复位值: 0x0000 0077

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31:7	保留	必须保持复位值。
6	TIMERFUNEN	TIMER 功能时钟使能 由软件置位或复位 0: 关闭 TIMER 功能时钟 1: 开启 TIMER 功能时钟
5	TIMEREN	TIMER 时钟使能 由软件置位或复位

		0: 关闭 TIMER 时钟 1: 开启 TIMER 时钟 (CK_TIMER)
4	PMUEN	PMU 时钟使能 由软件置位或复位 0: 关闭 PMU 时钟 1: 开启 PMU 时钟
3	保留	必须保持复位值。
2	SYSCFGEN	SYSCFG 时钟使能 由软件置位或复位 0: 关闭 SYSCFG 时钟 1: 开启 SYSCFG 时钟
1	IRQEN	IRQ 时钟使能 由软件置位或复位 0: 关闭 IRQ 时钟 1: 开启 IRQ 时钟
0	保留	必须保持复位值。

### 3.3.3. 内核使能寄存器 (RCU\_COREEN)

地址偏移: 0x08

复位值: 0x0000 0007

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2	CORE100MEN	EtherCAT 内核 100M 时钟使能 由软件置位或复位 0: 关闭 CK_CORE_100M 时钟 1: 开启 CK_CORE_100M 时钟
1	CORE50MEN	EtherCAT 内核 50M 时钟使能 由软件置位或复位 0: 关闭 CK_CORE_50M 时钟

		1: 开启 CK_CORE_50M 时钟
0	CORE25MEN	EtherCAT 内核 25M 时钟使能 由软件置位或复位 0: 关闭 CK_CORE_25M 时钟 1: 开启 CK_CORE_25M 时钟

### 3.3.4. 时钟配置寄存器 (RCU\_CLKCFG)

地址偏移: 0x0C

复位值: 0x0528 0400

**注意:** PLLBWCTL、PLLNDIV和PLLDIV只能在PLL\_CFG\_KEY = 1时读取和写入。该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31:28	保留	必须保持复位值。
27:24	PLLBWCTL[3:0]	PLL 带宽控制信号
23:22	保留	必须保持复位值。
21:16	PLLNDIV[5:0]	PLL时钟倍频因子 由软件置位或复位 000000: 保留 000001: 保留 ... 000110: 保留 000111: 保留 001000: 倍频因子为8 001001: 倍频因子为9 001010: 倍频因子为10 ... 111110: 倍频因子为62 111111: 倍频因子为63
15:11	保留	必须保持复位值。

10:8	PLLDIV[2:0]	PLL时钟频率分频因子 由软件置位或复位 000: 分频因子为1 001: 分频因子为2 ... 110: 分频因子为7 111: 分频因子为8
7:6	保留	必须保持复位值。
5	PLLSTB	PLL时钟稳定标志位 硬件置1来表示PLL输出时钟是否稳定待用 0: PLL未稳定 1: PLL已稳定
4	保留	必须保持复位值。
3	HXTALSTB	高速晶体振荡器（HXTAL）时钟稳定标志位 硬件置‘1’来指示HXTAL振荡器时钟是否稳定待用 0: HXTAL振荡器未稳定 1: HXTAL振荡器已稳定
2:0	保留	必须保持复位值。

### 3.3.5. 复位配置寄存器（RCU\_RSTCFG）

地址偏移：0x10

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:10	保留	必须保持复位值。
9	PHYBREL	端口B的PHY释放 0: 端口B的PHY保持复位状态 1: 端口B的PHY从复位状态释放 <b>注意:</b> 当phyrst_mode = 1时, 该位有效
8	PHYAREL	端口A的PHY释放 0: 端口A的PHY保持复位状态 1: 端口A的PHY从复位状态释放

**注意：**当`phyrst_mode = 1`时，该位有效

7	保留	必须保持复位值。
6	ESCRST	<b>EtherCAT复位</b> 将该位置1将复位EtherCAT内核。当EtherCAT内核从复位状态释放时，该位由硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。
5:3	保留	必须保持复位值。
2	PHYBRST	<b>端口B复位</b> 将此位置1将复位端口B PHY。当端口B PHY从复位状态释放时，该位由硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。
1	PHYARST	<b>端口 A PHY 复位</b> 将此位置 1 将复位端口 A PHY。当端口 A PHY 从复位状态释放时，该位由硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。
0	DRST	<b>数字复位</b> 将此位置1将复位整个芯片（除了锁相环，端口B PHY和端口A PHY）。当芯片从复位状态被释放时，该位被硬件自动清零。当该位被置位时，所有写入该位的操作都会被忽略。

### 3.3.6. PLL 配置密钥寄存器（RCU\_PLL\_CFG\_KEY）

地址偏移：0x14

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:1	保留	必须保持复位值。
0	PLL_CFG_KEY	当寄存器写入 0x78b465a1 时，该位为 1 并且可以读写 RCU_CLKCFG 寄存器中的 PLLBWCTL、PLLN 和 PLLDIV 位。

### 3.3.7. 复位标志寄存器（RCU\_PRSTF）

地址偏移：0x18

复位值：0x0000 0002

该寄存器可以按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留													PRSTF	PRSTC	
													r	rw	

位/位域	名称	描述
31:2	保留	必须保持复位值。
1	PRSTF	Pin 复位标志 该位在引脚复位时置位，通过置位 PRSTC 清零。 0: 未发生 Pin 复位 1: 发生 Pin 复位
0	PRSTC	Pin 复位标志清除 写 1 清除 PRSTF 标志

## 4. 中断控制器（INTC）

### 4.1. 概述

器件提供了可编程的多层中断结构，通过中断控制器（INTC）进行控制。中断事件是由各个子模块在内部生成，经过配置可由IRQ引脚输出单个外部主机中断。

### 4.2. 特征

- IRQ中断缓冲器模式、极性和置为无效间隔时间皆可配置；
- IRQ中断可配置为开漏输出，实现多个器件共用中断；
- 所有内部中断均可屏蔽且能够触发IRQ中断；
- 器件支持生成以下8种类型中断：
  - 软件中断；
  - 器件就绪中断；
  - 以太网PHY中断；
  - 定时器中断；
  - PME中断；
  - AHB2OPB桥接中断；
  - EtherCAT中断；
  - 时钟输出测试模式。

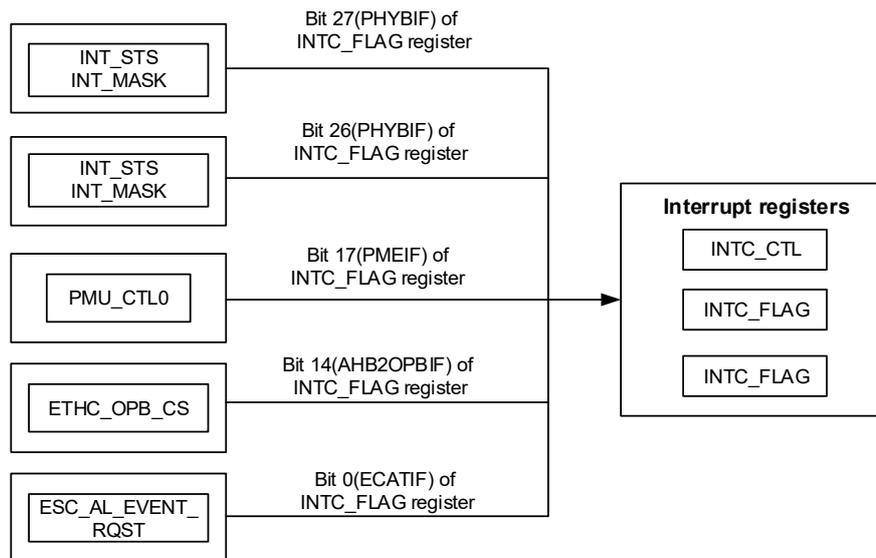
### 4.3. 功能说明

器件的中断按照中断源是否在子模块的寄存器中进行使能和清除操作，可划分为以下两种类型：

- 第一种类型包含软件、器件就绪和定时器中断，通过INTC\_FLAG寄存器和INTC\_EN寄存器对中断直接进行访问与配置（含监控、使能 / 禁止和清除）。
- 第二种类型包含以太网PHY、电源管理、AHB2OPB桥接和EtherCAT中断，INTC\_FLAG寄存器可以提供中断事件指示，但没有具体的中断源信息，需要软件轮询子模块的中断寄存器才能确定中断源。只有在处理了中断并将中断源清除后，才能清除INTC\_FLAG寄存器对应中断。

中断事件能够触发外部IRQ中断引脚输出，通过配置INTC\_CTL寄存器可以实现使能 / 禁止IRQ中断引脚输出以及配置IRQ中断缓冲器模式、极性和置为无效间隔时间。INTC\_CTL寄存器的DEAS位域用于配置中断请求无效间隔时间，其确保了最小的IRQ输出中断置为无效间隔周期，且置为无效间隔始终从IRQ引脚置为无效时开始。中断寄存器和中断源控制寄存器之间的关系，如[图4-1. 中断框图](#)所示。

图 4-1. 中断框图



#### 4.3.1. 软件中断

中断控制器提供了对通用的软件中断的控制。当 INTC\_EN 寄存器的 SWIE 位从 0 切换为 1 时，INTC\_FLAG 寄存器的 SWIF 位会被置位。此中断提供了一种较为简单的软件产生中断的方法，用于常规软件设计。

为了使软件中断事件能够触发外部 IRQ 中断引脚，必须置位 INTC\_CTL 寄存器的 IRQEN 位使能 IRQ 输出。

#### 4.3.2. 设备准备就绪中断

中断控制器提供了对器件就绪中断的控制。当 INTC\_EN 寄存器的 READYIE 位从 0 切换为 1 时，INTC\_FLAG 寄存器的 READYIF 位用于指示器件已准备就绪，并在上电或复位后可接受访问。

为了使器件就绪中断事件能够触发外部 IRQ 中断引脚，必须置位 INTC\_CTL 寄存器的 IRQEN 位使能 IRQ 输出。

#### 4.3.3. 以太网 PHY 中断

中断控制器提供了对以太网 PHY 中断的控制。当 INTC\_EN 寄存器的 PHYAIE 位从 0 切换为 1 时，INTC\_FLAG 寄存器的 PHYAIF 和 PHYBIF 位用于指示来自以太网 PHY 的中断事件。关于以太网 PHY 中断源的详细信息，请参考 [以太网PHYS](#)。

为了使以太网 PHY 中断事件能够触发外部 IRQ 中断引脚，必须置位 INTC\_CTL 寄存器的 IRQEN 位使能 IRQ 输出。

#### 4.3.4. 定时器中断

中断控制器提供了对定时器中断的控制。当定时器的计数寄存器值从 0 变为 0xFFFF 时，将产生此中断。当 INTC\_EN 寄存器的 TIMIE 位从 0 切换为 1 时，INTC\_FLAG 寄存器的 TIMIF 位用于指示

来自定时器的中断事件。

为了使定时器中断事件能够触发外部IRQ中断引脚，必须置位INTC\_CTL寄存器的IRQEN位使能IRQ输出。

#### 4.3.5. PME 中断

中断控制器提供了对功耗管理中断的控制。当INTC\_EN寄存器的PMEIE位从0切换为1时，INTC\_FLAG寄存器的PMEIF位用于指示来自PMU的中断事件。关于电源管理中断源的详细信息，请参考[电源管理单元 \(PMU\)](#)。

为了使功耗管理中断事件能够触发外部IRQ中断引脚，必须置位INTC\_CTL寄存器的IRQEN位使能IRQ输出。

#### 4.3.6. AHB2OPB 桥中断

中断控制器提供了对AHB2OPB桥接中断的控制。当INTC\_EN寄存器的AHB2OPBIE位从0切换为1时，INTC\_FLAG寄存器的AHB2OPBIF位用于指示来自BUS的AHB2OPB桥接中断事件。关于AHB2OPB桥接中断源的详细信息，请参考[系统和总线架构](#)。

为了使AHB2OPB桥接中断事件能够触发外部IRQ中断引脚，必须置位INTC\_CTL寄存器的IRQEN位使能IRQ输出。

#### 4.3.7. EtherCAT 中断

中断控制器提供了对EtherCAT中断的控制。当INTC\_EN寄存器的ECATIE位从0切换为1时，INTC\_FLAG寄存器的ECATIF位用于指示EtherCAT中断事件。关于EtherCAT中断源的详细信息，请参考[EtherCAT](#)。

为了使EtherCAT中断事件能够触发外部IRQ中断引脚，必须置位INTC\_CTL寄存器的IRQEN位使能IRQ输出。

#### 4.3.8. 时钟输出测试模式

为了进行系统调试，观察时钟情况，可通过设置INTC\_CTL寄存器的IRQCKOUT位为1，实现IRQ引脚输出晶振时钟。此时，IRQ引脚须配置为推挽输出模式（IRQMODE=1），以求达到最佳效果。

### 4.4. INTC 寄存器

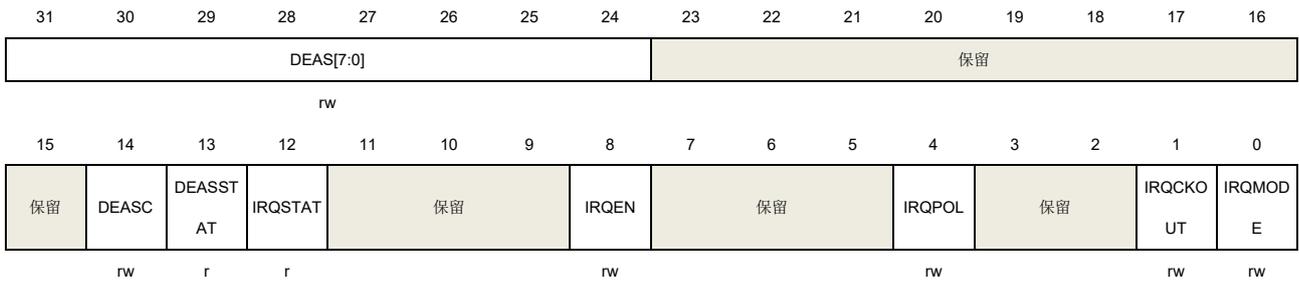
INTC 基地址：0x3A00

#### 4.4.1. 控制寄存器（INTC\_CTL）

地址偏移：0x00

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。



位/位域	名称	描述
31:24	DEAS[7:0]	中断无效间隔时间 该位域用于配置中断请求无效间隔时间，单位为 10us。 该位域配置为 0 将导致器件禁用 DEAS 时间间隔、复位间隔计数器并发送出任何挂起的中断。该位域配置为非 0 值，则后续产生的中断都将遵循该设置值。
23:15	保留	必须保持复位值。
14	DEASC	中断无效间隔时间清除 该位域配置为 1，将会清零无效间隔计数器，同时会开启新的无效间隔计数（无论当前是否处于激活的无效间隔时间）。该位置位后，将由硬件自动清零。当器件产生软件复位时，该位不会被复位。 0: 无影响 1: 清零无效间隔计数器
13	DEASSTAT	中断无效间隔状态 该位用于指示中断无效间隔的状态。 0: 不处于无效间隔时间（中断会发送至 IRQ 引脚） 1: 处于无效间隔时间（中断不会发送至 IRQ 引脚）
12	IRQSTAT	内部 IRQ 线状态 该位用于指示内部 IRQ 线的状态，其不受 IRQEN 位配置影响。 0: 无使能的中断处于激活状态 1: 存在使能的中断处于激活状态
11:9	保留	必须保持复位值。
8	IRQEN	IRQ 引脚输出使能 该位控制 IRQ 引脚的中断输出功能。 0: 禁止 IRQ 引脚输出 1: 使能 IRQ 引脚输出
7:5	保留	必须保持复位值。
4	IRQPOL	IRQ 输出极性 当器件产生软件复位时，该位不会被复位。当设备发生系统复位(如上电复位、引脚复位和 EtherCAT 系统复位)时，此位复位。 0: IRQ 有效输出为低电平 1: IRQ 有效输出为高电平

**注意：**当 IRQ 配置为开漏输出（IRQMODE=0）时，该位被忽略且 IRQ 有效输出始终为低电平。

3:2	保留	必须保持复位值。
1	IRQCKOUT	IRQ 时钟输出 0: 无时钟输出 1: IRQ 引脚输出晶振时钟（用于系统调试，观察时钟） <b>注意：</b> 当该位配置为 1 时，IRQ 引脚须配置为推挽输出模式（IRQMODE=1）。
0	IRQMODE	IRQ 引脚输出模式 当器件产生软件复位时，该位不会被复位。当设备发生系统复位(如上电复位、引脚复位和 EtherCAT 系统复位)时，此位复位。 0: 输出开漏模式 1: 输出推挽模式

## 4.4.2. 标志寄存器（INTC\_FLAG）

地址偏移：0x04

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIF	READYIF	保留		PHYBIF	PHYAIF	保留					TIMIF	保留	PMEIF	保留	
rc_w1	rc_w1			r	r						rc_w1			rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		AHB2OP		保留										ECATIF	
		BIF													

位/位域	名称	描述
31	SWIF	软件中断标志 当 INTC_EN 寄存器的 SWIE 位为 1 时，将产生此中断。软件写 1 清 0。
30	READYIF	器件就绪中断标志 此中断用于指示器件已准备就绪，并在上电或复位后接受访问。软件写 1 清 0。
29:28	保留	必须保持复位值。
27	PHYBIF	PHY B 中断标志 该位用于指示来自 PHY B 的中断事件。
26	PHYAIF	PHY A 中断标志 该位用于指示来自 PHY A 的中断事件。
25:20	保留	必须保持复位值。
19	TIMIF	定时器中断标志

当定时器的计数寄存器从 0 变为 0xFFFF 时，将产生此中断。软件写 1 清 0。

18	保留	必须保持复位值。
17	PMEIF	<p>PME 中断标志</p> <p>当电源管理单元检测到 PMU_CTL 寄存器中配置的功耗管理事件时，将产生此中断。软件写 1 清 0。</p> <p><b>注意：</b>配置中断请求无效间隔不适用于 PME 中断。</p>
16:15	保留	必须保持复位值。
14	AHB2OPBIF	<p>AHB2OPB 桥接中断标志</p> <p>该位用于指示来自 BUS 的 AHB2OPB 桥接中断事件。</p>
13:1	保留	必须保持复位值。
0	ECATIF	<p>EtherCAT 中断标志</p> <p>该位用于指示 EtherCAT 中断事件。</p>

### 4.4.3. 使能寄存器 (INTC\_EN)

地址偏移：0x08

复位值：0x0000 0000

该寄存器只能按字（32 位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIE	READYIE	保留	PHYBIE	PHYAIE	保留							TIMIE	保留	PMEIE	保留
rw	rw		rw	rw								rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	AHB2OPBIE	保留											ECATIE		
	rw												rw		

位/位域	名称	描述
31	SWIE	<p>软件中断使能</p> <p>0: 禁止软件中断</p> <p>1: 使能软件中断</p>
30	READYIE	<p>器件就绪中断使能</p> <p>0: 禁止器件就绪中断</p> <p>1: 使能器件就绪中断</p>
29:28	保留	必须保持复位值。
27	PHYBIE	<p>PHY B 中断使能</p> <p>0: 禁止以太网 PHY B 中断</p> <p>1: 使能以太网PHY B中断</p>
26	PHYAIE	PHY A 中断使能

		0: 禁止以太网 PHY A 中断 1: 使能以太网PHY A中断
25:20	保留	必须保持复位值。
19	TIMIE	定时器中断使能 0: 禁止定时器中断 1: 使能定时器中断
18	保留	必须保持复位值。
17	PMEIE	PME 中断使能 0: 禁止功耗管理中断 1: 使能功耗管理中断
16:15	保留	必须保持复位值。
14	AHB2OPBIE	AHB2OPB 桥接中断使能 0: 禁止 AHB2OPB 桥接中断 1: 使能AHB2OPB桥接中断
13:1	保留	必须保持复位值。
0	ECATIE	EtherCAT 中断使能 0: 禁止 EtherCAT 中断 1: 使能EtherCAT中断

## 5. 通用输入/输出 (GPIO)

### 5.1. 概述

最多支持 35 个通用输入 / 输出引脚 (GPIO) 的模式配置。每个 GPIO 端口将根据芯片的当前工作模式确定该端口的当前功能，包括输入 / 输出模式。每个 GPIO 引脚可以配置为上拉 / 下拉或浮空。当引脚处于输出模式时，引脚可以配置为推挽/漏极开路/源开路输出。

### 5.2. 特征

- 每个引脚具有弱上拉 / 下拉功能；
- 推挽 / 漏极开漏 / 源极开漏输出使能控制；
- 根据芯片模式配置选定引脚的功能；

### 5.3. 功能概述

#### 5.3.1. GPIO 引脚配置

在复位期间，所有 GPIO 端口都被配置成输入浮空模式，这种输入模式禁用上拉 (PU) / 下拉 (PD) 电阻。但是复位后，等待 EEPROM 加载，加载完成后。根据 ESC PDI\_TYPE 决定芯片复位后引脚的初始状态，当 PDI\_TYPE 等于 0x04 时，选择 Digital IO 模式，当 PDI\_TYPE 等于 0x80 时，选择 SPI 模式或 EXMC 模式。

GPIO 管脚依据工作状态控制为输入或输出状态。

所有的 GPIO 管脚都有一个内部的弱上拉和弱下拉可以选择。当 GPIO 引脚配置为输出引脚，用户可以配置输出驱动模式：推挽或漏极开漏以及源极开漏模式。上下拉模式以及输出模式配置支持通过 EXMC / SPI 通信写入。

#### 5.3.2. 外部中断/事件线路

只支持一个外部中断输出接口。中断输出配置由[中断控制器 \(INTC\)](#)的内部寄存器决定，输出模式也由内部 Bit 位决定。

#### 5.3.3. 备用功能

当芯片处于不同模式时，每个引脚具有不同的功能。

Digital IO 模式：当 PDI\_TYPE = 0x04 时，AFIO 被调整为 Digital IO 模式。

当 PDI\_TYPE = 0x80 且 MCU\_PDI\_TYPE 的引脚为 1 时，AFIO 调整为 EXMC 模式。

SPI 模式：当 PDI\_TYPE 等于 0x80 且 MCU\_PDI\_TYPE 的引脚等于 0 时，AFIO 调整为 SPI 模式。

表 5-1. GPIO 配置表

模式名称	寄存器 / 信号	描述
EXMC	pdi_type / mcu_pdi_type / sip_mode	pdi_type == 0x80; mcu_pdi_type == 1'b1; sip_mode == 1'b0
DIO	pdi_type / sip_mode	pdi_type == 0x04; sip_mode == 1'b0
spi8w_gpio	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode	pdi_type == 0x80; mcu_pdi_type == 1'b0; sip_mode == 1'b0; line_mode == 2'b11; LINKACTLED1 引脚必须外部下拉
spi4w_mii_down	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode / chip_mode	pdi_type == 0x80; mcu_pdi_type == 1'b0; sip_mode == 1'b0; line_mode == 2'b10; chip_mode == 2'b10 LINKACTLED1 引脚必须在外部上拉
spi4w_mii_up	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode / chip_mode	pdi_type == 0x80; mcu_pdi_type == 1'b0; sip_mode == 1'b0; line_mode == 2'b10; chip_mode == 2'b11; LINKACTLED1 引脚必须在外部上拉

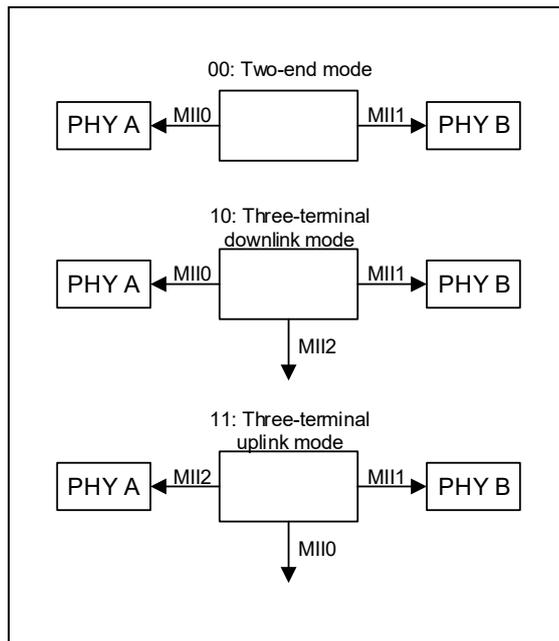
此外，当处于 SPI（2/4/8 线）+MII 模式且 chip\_mode[1:0]不等于 0x11 时，EtherCAT 端口 0 连接到内部 PHY A。

当 chip\_mode[1:0]等于 0x11 时，EtherCAT 端口 0 连接到 MII 引脚，端口 2 连接到内部 PHY A。

当 chip\_mode[1:0]等于 0x10 时，EtherCAT 端口 2 连接到 MII 引脚。

当 chip\_mode[1:0]等于 0x00 时，在这种情况下，SPI + GPIO 模式的输出不受影响，而 SPI + MII 模式中 MII 信号不输出。

图 5-1. 端口连接 PHYS



## 备注:

- 部分引脚在上电复位期间或当 RST# 设置为无效时被锁定，并在被锁定后自动切换。
- MII\_LINKPOL 信号在复位后被锁存，以确定 MII\_LINK 引脚的极性。如果 MII\_LINK 等于 0，表示电平低，表明已建立 100 Mbps 全双工链路。MII\_LINK 等于 1，表示电平高，表明已建立 100 Mbps 全双工链路。
- SYNC1\_LATCH1/SYNC0\_LATCH0 pad 的 omode / io\_en 由 ESC 内部寄存器决定。
- 如下锁存器信号必须在以下模式中配置上下拉配置，并且不能配置为 X 状态。
  - 如果 spi\_ext\_mode 等于 1 且 inphy\_bypass 等于 0，则 chip\_mode [1:0] 必须设置为下拉状态。
  - IO16 引脚必须配置为上拉或下拉状态。
  - EESIZE 引脚必须配置为上拉或下拉状态。
  - IO17 引脚必须配置为下拉和下拉状态。
  - 当 inphy\_bypass 等于 1 时，MII\_LINKPOL 引脚必须配置为上拉或下拉状态。

PDI\_TYPE: 参考以 EtherCAT 寄存器 [ESC PDI 控制寄存器 \(ESC PDI CONTROL\)](#)。

line\_mode: SPI 输出到 GPIO 是由 SPI 输入指令决定的。

spi\_ext\_mode: 参考 [系统配置寄存器 0 \(SYSCFG\\_CFG0\)](#) Bit 2。

chip\_mode [1:0]: 复位后，LINKACTLED1/LINKACTLED0 的引脚被锁定。

inphy\_bypass: 由工厂设置的寄存器位被配置为 1，不能修改。

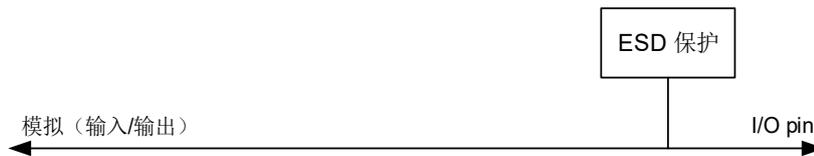
### 5.3.4. 模拟配置

当 GPIO 引脚用于模拟配置时：

- 弱上拉和下拉电阻禁用；
- 输出缓冲器禁用；
- 施密特触发器输入禁用；
- 读取端口输入状态寄存器“0”。

[图 5-2. 模拟配置的基本结构](#)是 I/O 端口的模拟模式配置。

图 5-2. 模拟配置的基本结构



### 5.3.5. 备用功能(AF)配置

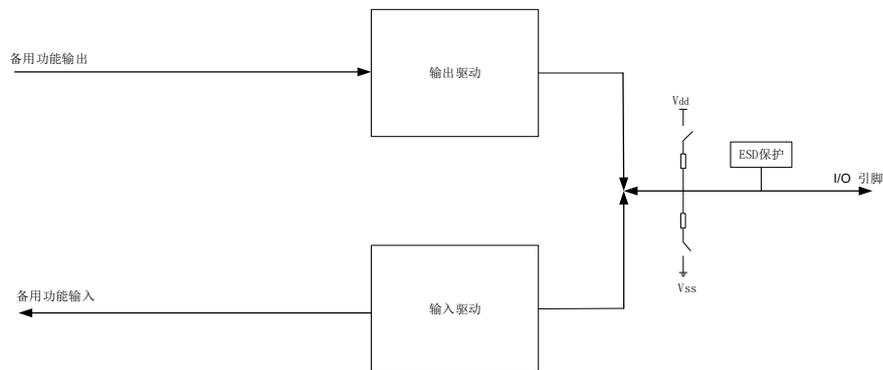
为了适应不同的器件封装，GPIO 端口支持软件配置将一些备用功能应用到其他引脚上。

当引脚配置为备用功能时：

- 输出缓冲器启用开漏或者推挽功能；
- 输出缓冲器由外设驱动；
- 施密特触发输入使能；
- 可选择的弱上拉/下拉电阻。

[图 5-3. 备用功能配置的基本结构](#)是 I/O 端口备用功能配置图。

图 5-3. 备用功能配置的基本结构



**注意：**

OSPI 模式下，由于 SPI PIN 脚占用数量较多，在 OSPI+GPIO 模式下不能使用 pdi\_gpio15；OSPI+MII 模式下不能使用 MII\_CLK25。

## 5.4. 寄存器定义

GPIO 基地址：0x3500

### 5.4.1. 端口输出模式寄存器 0 (GPIO0\_OMODE0)

地址偏移：0x00

复位值：0x0000 0000

该寄存器可以按字（32位）访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OM015[1:0]		OM014[1:0]		OM013[1:0]		OM012[1:0]		OM011[1:0]		OM010[1:0]		保留		OM008[1:0]	
rw		rw		rw		rw		rw		rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM007[1:0]		OM006[1:0]		OM005[1:0]		OM004[1:0]		保留		OM002[1:0]		OM001[1:0]		OM000[1:0]	
rw		rw		rw		rw				rw		rw		rw	

位/位域	名称	描述
31:30	OM015[1:0]	Pin IO11输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
29:28	OM014[1:0]	Pin OE_EXT输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
27:26	OM013[1:0]	Pin IO4输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
25:24	OM012[1:0]	Pin IO5输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
23:22	OM011[1:0]	Pin IO6输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
21:20	OM010[1:0]	Pin LATCH_IN输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
19:18	保留	必须保持复位值。
17:16	OM008[1:0]	Pin WD_STATE输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

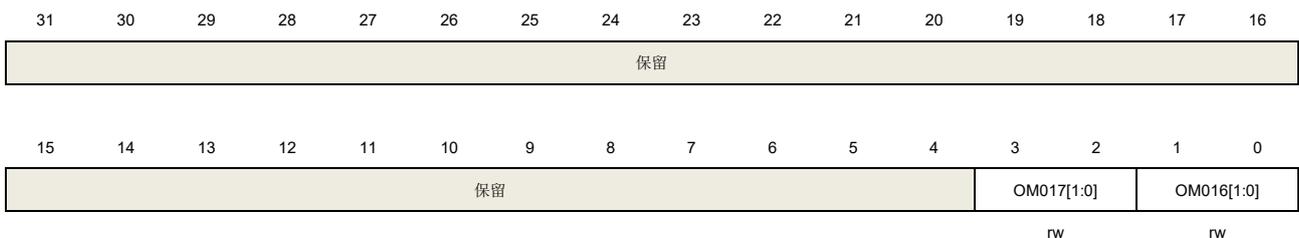
15:14	OM007[1:0]	Pin IO7输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
13:12	OM006[1:0]	Pin IO8输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
11:10	OM005[1:0]	Pin EOF输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
9:8	OM004[1:0]	Pin SOF输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
7:6	保留	必须保持复位值
5:4	OM002[1:0]	Pin IO18输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
3:2	OM001[1:0]	Pin IO17输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM000[1:0]	Pin IO16输出模式位 该位由软件置位和清除。 00: 输出推挽模式（复位值） 01: 输出开漏模式 10: 输出开源模式 11: 保留

#### 5.4.2. 端口输出模式寄存器 1（GPIO0\_OMODE1）

地址偏移：0x04

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
------	----	----

31:4	保留	必须保持复位值。
3:2	OM017[1:0]	Pin IO13输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM016[1:0]	Pin IO12输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

### 5.4.3. 端口输出模式寄存器 2 (GPIO1\_OMOD0)

地址偏移: 0x08

复位值: 0x0000 0000

该寄存器可以按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OM115	OM114	OM113	保留	OM111	OM110	保留	OM108								
rw	rw	rw		rw	rw		rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM107	OM106	OM105	保留	OM103	OM102	OM101	OM100								
rw	rw	rw		rw	rw	rw	rw								

位/位域	名称	描述
31:30	OM115[1:0]	Pin LINKACTLED0输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
29:28	OM114[1:0]	Pin LINKACTLED1输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
27:26	OM113[1:0]	Pin EESIZE输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
25:24	保留	必须保持复位值。
23:22	OM111[1:0]	Pin EESCL输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
21:20	OM110[1:0]	Pin EESDA输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
19:18	保留	必须保持复位值。
17:16	OM108[1:0]	Pin IO2输出模式位

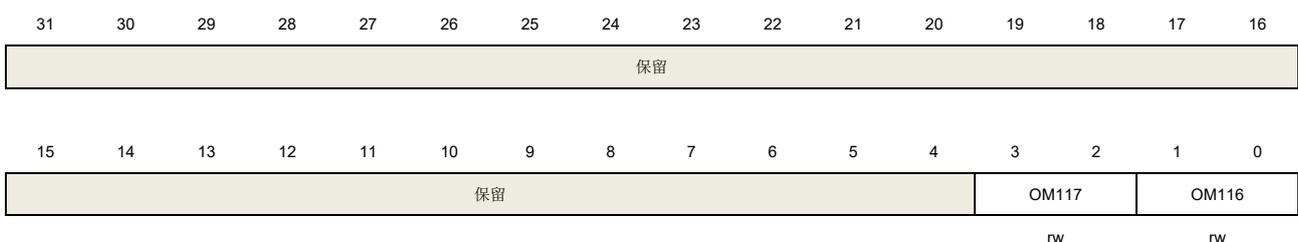
		该位由软件置位和清除。 参考 OM000[1:0]的描述
15:14	OM107[1:0]	Pin IO1输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
13:12	OM106[1:0]	Pin IO0输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
11:10	OM105[1:0]	Pin WD_TRIG输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
9:8	保留	必须保持复位值。
7:6	OM103[1:0]	Pin IO9输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
5:4	OM102[1:0]	Pin IO15输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
3:2	OM101[1:0]	Pin IO14输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM100[1:0]	Pin IO10输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

#### 5.4.4. 端口输出模式寄存器 3 (GPIO1\_OMOD1)

地址偏移：0x0C

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:4	保留	必须保持复位值。

3:2	OM117[1:0]	Pin OUTVALID输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述
1:0	OM116[1:0]	Pin IO3输出模式位 该位由软件置位和清除。 参考 OM000[1:0]的描述

#### 5.4.5. 端口上拉/下拉寄存器 0 (GPIO0\_PUD0)

地址偏移: 0x10

复位值: 0x0000 0000

该寄存器可以按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD015[1:0]		PUD014[1:0]		PUD013[1:0]		PUD012[1:0]		PUD011[1:0]		PUD010[1:0]		PUD009[1:0]		PUD008[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD007[1:0]		PUD006[1:0]		PUD005[1:0]		PUD004[1:0]		保留		PUD002[1:0]		PUD001[1:0]		PUD000[1:0]	
rw		rw		rw		rw				rw		rw		rw	

位/位域	名称	描述
31:30	PUD015[1:0]	Pin IO11上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
29:28	PUD014[1:0]	Pin OE_EXT上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
27:26	PUD013[1:0]	Pin IO4上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
25:24	PUD012[1:0]	Pin IO5上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
23:22	PUD011[1:0]	Pin IO6上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
21:20	PUD010[1:0]	Pin LATCH_IN上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
19:18	PUD009[1:0]	Pin SYNC1_LATCH1上拉或下拉位 该位由软件置位和清除。

		参照 PUD000[1:0]的描述
17:16	PUD008[1:0]	Pin WD_STATE上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
15:14	PUD007[1:0]	Pin IO7上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
13:12	PUD006[1:0]	Pin IO8上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
11:10	PUD005[1:0]	Pin EOF上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
9:8	PUD004[1:0]	Pin SOF上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
7:6	保留	必须保持复位值。
5:4	PUD002[1:0]	Pin IO18上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
3:2	PUD001[1:0]	Pin IO17上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD000[1:0]	Pin IO16上拉或下拉位 该位由软件置位和清除。 00: 悬空模式，无上拉和下拉（复位值） 01: 端口上拉模式 10: 端口下拉模式 11: 模拟模式

#### 5.4.6. 端口上拉/下拉寄存器 1 (GPIO0\_PUD1)

地址偏移: 0x14

复位值: 0x0000 0000

该寄存器可以按字（32位）访问。



保留	PUD017[1:0]	PUD016[1:0]
	rw	rw

位/位域	名称	描述
31:4	保留	必须保持复位值。
3:2	PUD017[1:0]	Pin IO13上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD016[1:0]	Pin IO12上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述

#### 5.4.7. 端口上拉/下拉寄存器 2 (GPIO1\_PUD0)

地址偏移: 0x18

复位值: 0x0000 0000

该寄存器可以按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD115[1:0]	PUD114[1:0]	PUD113[1:0]	PUD112[1:0]	PUD111[1:0]	PUD110[1:0]	PUD109[1:0]	PUD108[1:0]								
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD107[1:0]	PUD106[1:0]	PUD105[1:0]	PUD104[1:0]	PUD103[1:0]	PUD102[1:0]	PUD101[1:0]	PUD100[1:0]								
rw															

位/位域	名称	描述
31:30	PUD115[1:0]	Pin LINKACTLED0上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
29:28	PUD114[1:0]	Pin LINKACTLED1上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
27:26	PUD113[1:0]	Pin EESIZE上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
25:24	PUD112[1:0]	Pin IRQ上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
23:22	PUD111[1:0]	Pin EESCL上拉或下拉位 该位由软件置位和清除。

		参照 PUD000[1:0]的描述
21:20	PUD110[1:0]	Pin EESDA上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
19:18	PUD109[1:0]	Pin TESTMODE上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
17:16	PUD108[1:0]	Pin IO2上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
15:14	PUD107[1:0]	Pin IO1上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
13:12	PUD106[1:0]	Pin IO0上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
11:10	PUD105[1:0]	Pin WD_TRIG上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
9:8	PUD104[1:0]	Pin SYNC0_LATCH0上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
7:6	PUD103[1:0]	Pin IO9上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
5:4	PUD102[1:0]	Pin IO15上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
3:2	PUD101[1:0]	Pin IO14上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD100[1:0]	Pin IO10上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述

#### 5.4.8. 端口上拉/下拉寄存器 3 (GPIO1\_PUD1)

地址偏移: 0x1C

复位值：0x0000 0000

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:4	保留	必须保持复位值。
3:2	PUD117[1:0]	Pin OUTVALID上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述
1:0	PUD116[1:0]	Pin IO3上拉或下拉位 该位由软件置位和清除。 参照 PUD000[1:0]的描述

#### 5.4.9. EXMC 控制寄存器（EXMC\_CTL）

地址偏移：0x20

复位值：0x0000 0004

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:3	保留	必须保持复位值。
2:1	EXMCHSIZE[1:0]	EXMC hsize 00: 8-bit 01: 16-bit 10: 32-bit 11: 保留
0	EXMCTYPE	EXMC 类型 0: 8-bit EXMC

1: 16-bit EXMC

## 6. 定时器（TIMER）

### 6.1. 基本定时器

#### 6.1.1. 简介

基本定时器模块有一个 16 位计数器，可以用作无符号计数器。基本定时器可以配置为生成中断。基本定时器的分辨率为 100 $\mu$ s。

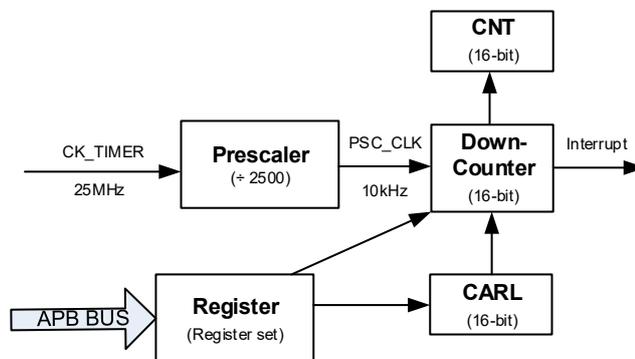
#### 6.1.2. 主要特征

- 计数器宽度：16 位；
- 计数时钟的来源仅为内部时钟；
- 计数模式：向下计数；
- 分辨率：100 微秒；
- 自动重载功能；
- 中断输出：更新事件。

#### 6.1.3. 结构框图

[图6-1. 基本定时器结构框图](#)提供了基本定时器内部配置的细节。

图 6-1. 基本定时器结构框图



#### 6.1.4. 功能说明

##### 时钟源选择

基本定时器的时钟源只能是来自RCU内部25MHz的CK\_TIMER。

CK\_TIMER被进行2500分频生成10kHz的计数器时钟（PSC\_CLK）。计数器的分辨率为100 $\mu$ s。

##### 计数器向下计数模式

当TIMERx\_CTL0寄存器中的CEN位置1时，基本定时器会将TIMER\_CTL0寄存器中的自动加

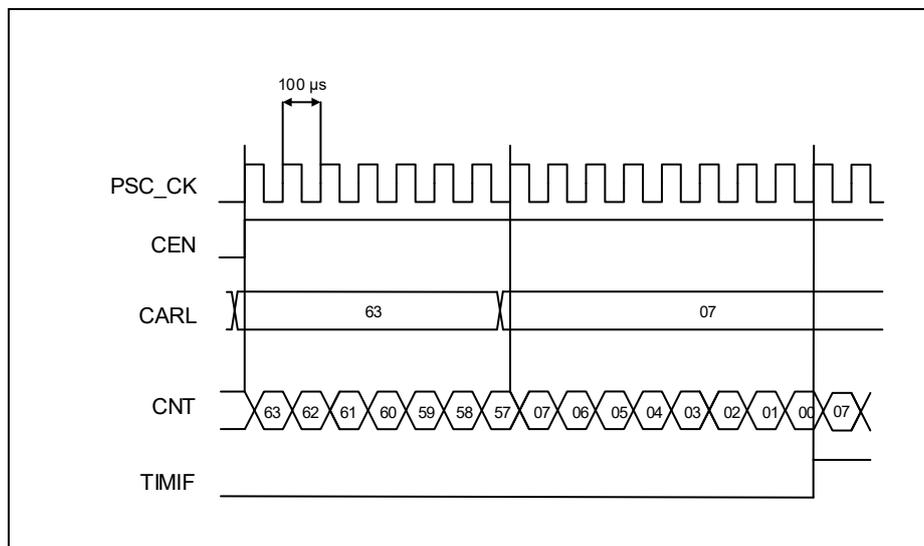
载值 (CARL) 加载到TIMER\_CNT寄存器中。软件可以随时写入新的自动加载值。如果CEN位置1, TIMER\_CNT寄存器将立即被加载到新的自动加载值, 并从该值开始继续向下计数。

如果发生复位或TIMER\_CTL0寄存器中的CEN位从1变为0, 则自动加载值将初始化为0xFFFF。在复位时, 计数值 (CNT) 将初始化为0xFFFF。

当TIMER\_CTL0寄存器中的CEN位置1时, 计数值主要会从自动加载值持续递减到0。一旦计数器达到0, 计数值会回绕到0xFFFF, 并且INTC\_FLAG寄存器中的TIMIF位置1。如果INTC\_EN寄存器中的TIMIE位置1, 基本定时器就会产生中断。计数值会继续从自动加载值开始递减。如果INTC\_FLAG寄存器中的TIMIF位置1, 它只能通过写入1来清除该位。

[图6-2. 向下计数时序图](#)展示了当自动加载值从0x63转换为0x07时计数器行为的一个示例。

图 6-2. 向下计数时序图



### 6.1.5. 寄存器定义

基本定时器基地址: 0x0000 3800

#### 控制寄存器 0 (TIMER\_CTL0)

地址偏移: 0x00

复位值: 0x0000 FFFF

该寄存器只能按字 (32位) 访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CEN	保留												
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

位/位域	名称	描述
------	----	----

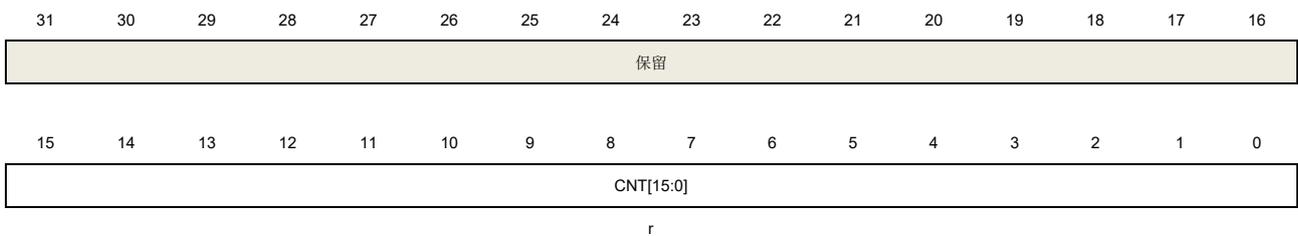
31:30	保留	必须保持复位值。
29	CEN	计数器使能 0: 计数器禁能 1: 计数器使能
28:16	保留	必须保持复位值。
15:0	CARL[15:0]	计数器自动重载值 这些位定义了计数器的自动重载值。 当CEN位置1, 计数器从当前值开始向下计数。 <b>注意:</b> 当CEN位从1到0, 这些位将被转化为0xFFFF。

### 计数器寄存器 (TIMER\_CNT)

地址偏移: 0x04

复位值: 0x0000 FFFF

该寄存器只能按字 (32位) 访问。



位/位域	名称	描述
31:16	保留	必须保持复位值。
15:0	CNT[15:0]	这些位是当前的计数值。

## 6.2. 独立运行计数器

### 6.2.1. 简介

独立运行计数器有一个32位的计数器, 可以用作无符号计数器。计数器时钟为25MHz。

### 6.2.2. 主要特征

- 计数器宽度: 32 位;
- 计数时钟的来源仅为内部时钟;
- 计数模式: 向上计数。

### 6.2.3. 结构框图

[图6-3. 独立运行计数器结构框图](#)提供了FRC内部配置的细节。

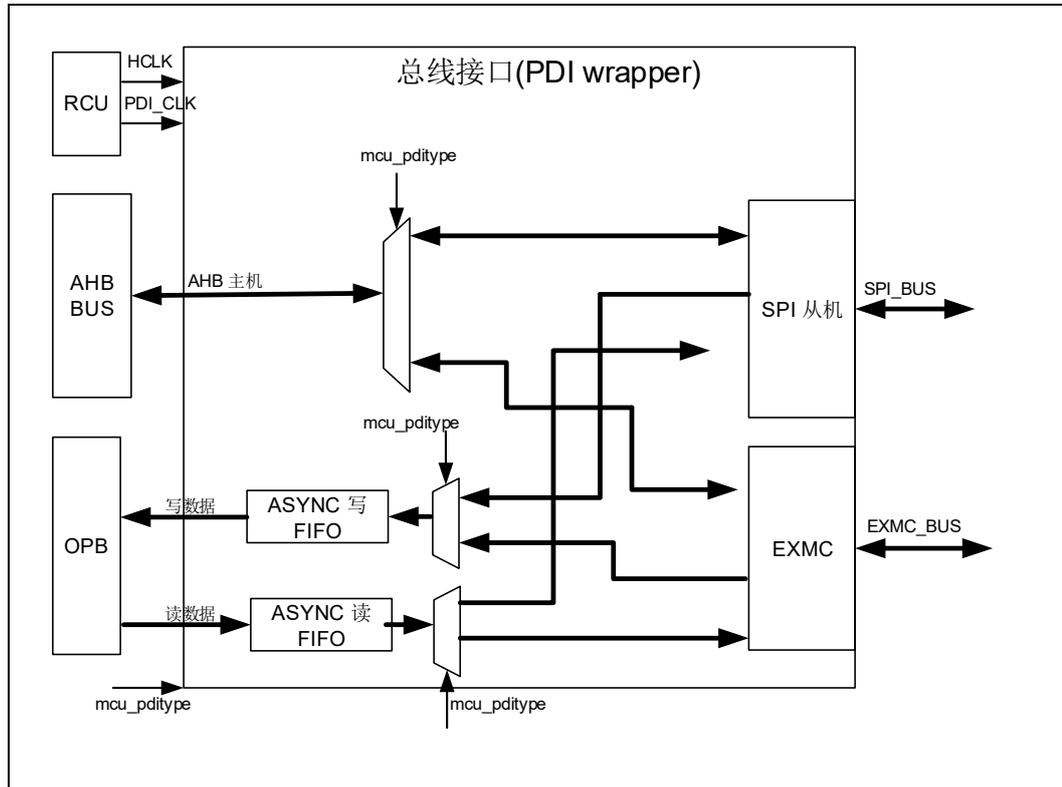


**注意：**计数器在复位事件后最多需要 160 纳秒才能清除。

## 7. 总线接口 (PDI Wrapper)

在 GDSCN 中, EXMC 和 SPI SLAVE 被封装成一个用于系统集成的包装器。PDI Wrapper 用于在 SPI 和 EXMC 之间选择数据。内部集成了两个异步 FIFO, 每个都是 16X32 位。SPI SLAVE 和 EXMC 一次只能工作一个, 由 MCU\_PDITYPE 引脚选择。

图 7-1. PDI Wrapper 框图



SPI 从机和 EXMC 不能同时有效。SPI 从机或 EXMC 通过 AHB 通道访问寄存器, 并通过异步读 FIFO 和异步写 FIFO 访问内核 RAM 数据。EXMC 的数据路径与 SPI 从机相同。

MCU\_PDITYPE 引脚选择工作访问接口模块。当 MCU\_PDITYPE 引脚为 0 时, 只有 SPI 从机可以访问内部数据。当 MCU\_PDITYPE 引脚为 1 时, 只有 EXMC 可以访问内部数据。

PDI\_CLK 为 SPI\_SLAVE、EXMC 和 ASYNC\_FIFO 提供时钟。当 MCU\_PDITYPE 的引脚为 0 时, PDI\_CLK 来自 SPI\_SCK。当 MCU\_PDITYPE 的引脚为 1 时, PDI\_CLK 来自 EXMC\_CLK。HCLK 是一个 100MHz 的系统时钟, 为 ASYNC\_FIFO、SPI\_SLAVE 和 EXMC 提供时钟。

### 7.1. SPI / QSPI / OSPI 从机

#### 7.1.1. 简介

EtherCAT 支持 SPI / QSPI / OSPI 从模块。

#### 7.1.2. 主要特征

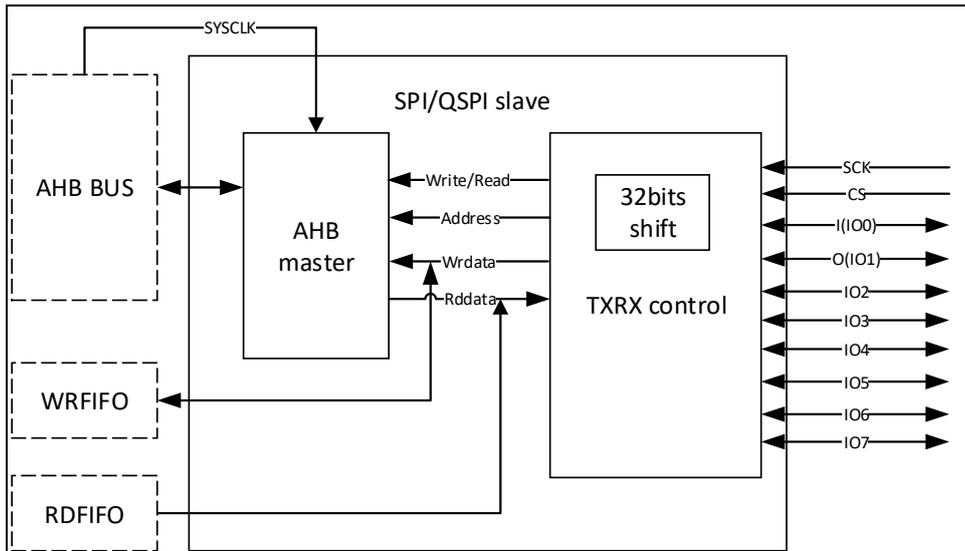
- 支持最高SPI时钟频率为100MHz;

- 仅支持从模式；
- 采样必须在时钟上升沿采样；
- 支持先进先出缓冲区访问。

### 7.1.3. 模块框图

SPI 的模块框图如 [图 7-2. SPI 的结构图](#) 所示。

图 7-2. SPI 的结构图



### 7.1.4. SPI 信号描述

#### 引脚描述

SPI / QSPI 从机模块包含两种引脚模式：4 线模式和 6 线模式。所有模式都包含公共引脚，SCK 和 CS。

表 7-1. 4 线模式

引脚名称	描述
SCK	SPI 时钟
CS	从机片选
I	输入引脚，接收 SPI / QSPI 主机数据
O	输出引脚，传输数据到 SPI / QSPI 主机

表 7-2. 6 线模式

引脚名称	描述
SCK	SPI / QSPI CLK
CS	从机片选
SIO0	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO1	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到

	SPI / QSPI 主机
SIO2	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO3	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机

**表 7-3. 8 线模式下的 OSPI**

引脚名称	描述
SCK	SPI / QSPI 时钟
CS	从机片选
SIO0	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO1	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO2	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO3	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO4	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO5	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO6	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机
SIO7	输入输出引脚，接收 SPI / QSPI 主机数据和传输数据到 SPI / QSPI 主机

### 7.1.5. SPI/QSPI/OSPI 从机控制器

#### 概述

SPI 从机接口可以使用较少的引脚访问寄存器和 FIFO。在 SPI 模式下支持单、双和四线位通道，时钟频率高达 100MHz。QSPI 模式始终使用四位通道，并且也以高达 80MHz 的速率运行。OSPI 模式始终使用八位通道，并且也以高达 80MHz 的速率运行。

#### 描述

以下是 SPI / QSPI / OSPI 客户端提供的功能概述：

- **快速读取：4 线**（时钟、选择、数据输入和数据输出）读取速度高达 80MHz。串行命令、地址和数据。首次访问使用 **dummy** 字节。支持单次和多次寄存器读取，具有递增、递减或静态寻址功能。
- **双 / 四输出读取：4 线或 6 线**（时钟、选择、数据输入/输出）以高达 80MHz 的速度读取。串行命令和地址，数据并行。首次访问使用 **dummy** 字节。具有递增、递减或静态寻址的单次和多次寄存器读取。
- **双 / 四 I/O 读取：4 线或 6 线**（时钟、选择、数据输入/输出）以高达 80MHz 的速度读取。串行命令，并行地址和数据。首次访问时的 **dummy** 字节（或字节）。具有递增、递减或静态寻址的单一和多重寄存器读取。
- **QSPI 读取：6 线**（时钟、选择、数据输入/输出）最高可达 80MHz 的写入速度。并行命令、地址和数据。首次访问时使用哑字节。支持单次和多次寄存器读取，具有递增、递减或静态寻址功能。
- **OSPI 读取：10 线**（时钟、选择、数据输入/输出）最高可写入 80MHz。并行命令、地址和数据。首次访问时的 **dummy** 字节。具有增量、递减或静态寻址的单次和多次寄存器读取。
- **四线**（时钟、选择、数据输入和数据输出）写入速度高达 80MHz。串行命令、地址和数据。具有递增、递减或静态寻址的单寄存器和多寄存器写入。
- **双 / 四数据写入：4 或 6 线**（时钟、选择、数据输入/输出）以高达 80MHz 的速度写入。串行命令和地址，平行数据。具有递增、递减或静态寻址的单寄存器和多寄存器写入。
- **双/四地址/数据写入：4 或 6 线**（时钟、选择、数据输入/输出）以高达 80MHz 的速度写入。串行命令，并行地址和数据。单次和多次寄存器写入，具有递增、递减或静态寻址。
- **QSPI 写入：6 线**（时钟、选择、数据输入/输出）最高可达 80MHz。并行命令、地址和数据。支持单次和多次寄存器写入，具有递增、递减或静态寻址功能。
- **OSPI 写：10 线**（时钟、选择、数据输入/输出）最高写入速度可达 80MHz。并行命令、地址和数据。支持单次和多次寄存器写入，具有递增、递减或静态寻址功能。

## 操作描述

在 **SCK** 输入时钟的上升沿，**IO [7:0]**引脚上的输入数据被采样。在 **IO [7:0]**引脚上，输出数据在时钟的下降沿被驱动。**SCK** 输入时钟可以是高电平脉冲或低电平脉冲。当 **CS** 片选输入为高电平时，**IO [7:0]**输入被忽略，**IO [7:0]**输出被置为三态。

在 **SPI** 模式下，8 位指令在 **CS** 有效电平后的第一个输入时钟上升沿开始。指令总是通过 **I/O** 串行输入。

对于读写指令，两个地址字节跟在指令字节后面。根据指令的不同，地址字节可以串行输入，或者每个时钟输入 2 位或 4 位。尽管所有寄存器都以双字节形式访问，但地址字段被认为是字节地址。地址字段的 15 位和 14 位指定地址是自动递减（10b）还是自动递增（01b）以进行连续访问。（如果访问内部 **FIFO**，将忽略 15 位和 14 位）

对于所有读指令，地址字节之后跟有 **dummy** 字节周期。在 **dummy** 字节周期期间，设备不驱动输出。**dummy** 字节（们）是串行输入的，或者是每时钟 2 位、4 位或 8 位。数据是串行输入的，或者是每时钟 2 位或 4 位。

对于读写指令，一个或多个 32 位数据字段跟随在 **dummy** 字节之后（如果有的话，否则它们跟随地址字节）。数据是串行输入的，或者每个时钟周期输入 2 位、4 位或 8 位数据。

QSPI 模式是通过启用四线 I/O (EQIO) 指令从 SPI 模式进入的。一旦进入 QSPI 模式，所有后续的命令、地址、空字节和数据字节都是每个时钟 4 位。QSPI 模式可以使用重置四线 I/O (RSTQIO) 指令退出。

OSPI 模式是通过 SPI 使用“启用八位 I/O (EOIO)”指令进入的。一旦进入 OSPI 模式，所有进一步的命令、地址、dummy 字节和数据字节都是每个时钟 8 位。OSPI 模式可以使用“重置八位 I/O (RSTQIO)”指令退出。

所有指令、地址和数据都以最高有效位 (MSB) 或最高有效二位 (di-bit) 或最高有效半字节 (nibble) 为先进进行传输。地址以最高有效字节 (MSB) 为先进进行传输。数据以最低有效字节 (LSB) 为先进进行传输 (小端模式)。

SPI 接口支持高达 100MHz 的输入时钟。(例外：对于 QSPI 指令，访问的数据字节数为 4，速度为 100MHz。如果主机想要访问更多的数据字节，主机可以使用较低的速度 (小于或等于 60MHz))。

SPI 接口支持连续命令之间至少 50 纳秒的时间 (CS 至少 50 纳秒的非有效时间)。

[表 7-4. SPI 指令](#) 列出了 SPI 模式下支持的指令。[表 7-5. QSPI 指令](#) 列出了 QSPI 指令。[表 7-6. OSPI 指令](#) 列出了 OSPI 指令。不支持的指令不得使用。

**表 7-4. SPI 指令**

指令	描述	位宽	指令码值	地址字节	dummy 字节	数据字节	最大频率
配置							
EQIO	使能 QSPI	1-0-0	38h	0	0	0	100Mhz
EOIO	使能 OSPI	1-0-0	3Ah	0	0	0	100Mhz
RSTIO	使能 SPI	1-0-0	FFh	0	0	0	100Mhz
读							
READ	读	1-1-1	0Bh	2	1	4 to ∞	100Mhz
SDOR	SPI 双线输出读	1-1-2	3Bh	2	1	4 to ∞	100Mhz
SDIOR	SPI 双线 I/O 读	1-2-2	BBh	2	2	4 to ∞	100Mhz
SQOR	SPI 四线输出读	1-1-4	6Bh	2	1	4 to ∞	100Mhz
SQIOR	SPI 四线 I/O 读	1-4-4	EBh	2	4	4 to ∞	100Mhz
写							
WRITE	写	1-1-1	02h	2	0	4 to ∞	100Mhz
SDDW	SPI 双线输出写	1-1-2	32h	2	0	4 to ∞	100Mhz
SDADW	SPI 双线 I/O 写	1-2-2	B2h	2	0	4 to ∞	100Mhz
SQDW	SPI 四线输出写	1-1-4	62h	2	0	4 to ∞	100Mhz
SQADW	SPI 四线 I/O 写	1-4-4	E2h	2	0	4 to ∞	100Mhz

**表 7-5. QSPI 指令**

指令	描述	位宽	指令码值	地址字节	dummy 字节	数据字节	最大频率
配置							
RSTQIO	复位 QSPI	4-0-0	FFh	0	0	4 to ∞	100Mhz
读							

READ	读	4-4-4	0Bh	2	3	4 to ∞	100Mhz
写							
WRITE	写	4-4-4	02h	2	0	4 to ∞	100Mhz

表 7-6. OSPI 指令

指令	描述	位宽	指令 码值	地址 字节	dummy 字节	数据字 节	最大频率
配置							
RSTOIO	复位 OSPI	8-0-0	FFh	0	0	4 to ∞	100Mhz
读							
READ	读	8-8-8	0Bh	2	8	4 to ∞	100Mhz
写							
WRITE	写	8-8-8	02h	2	0	4 to ∞	100Mhz

**注意：**位宽格式是：命令位宽，地址 / dummy 位宽，数据位宽。例如，1-2-4 表示命令使用 1 条线路，地址 / dummy 使用 2 条线路，数据使用 4 条线路。

## SPI 配置命令

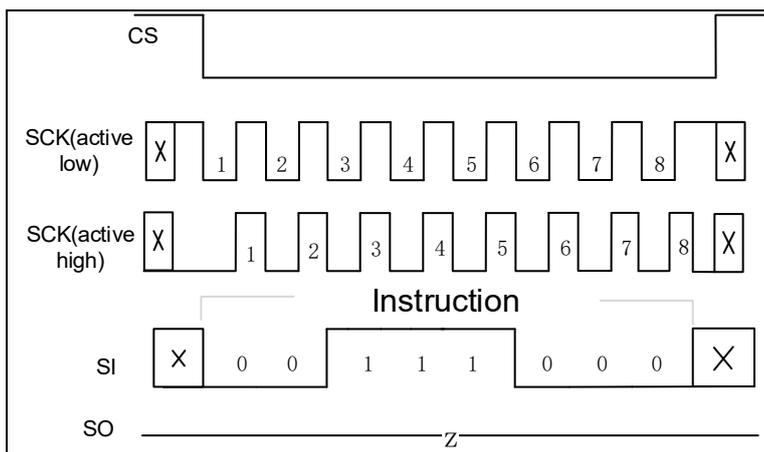
### 使能 QSPI

使能 QSPI 指令将操作模式更改为 QSPI。此指令仅在 SPI 总线协议中支持，并且仅支持最高 80MHz 的时钟频率。此指令在 QSPI 总线协议中不支持。

首先通过将 CS 置为有效信号来选择 SPI 客户端接口。8 位 EQIO 指令，38h，逐位通过 I/O[0] 引脚输入，每个时钟输入一位。将 CS 输入变为无效状态以结束周期。

[图 7-3. 使能 QSPI](#) 展示了使能 QSPI 指令。

图 7-3. 使能 QSPI



### 使能 OSPI

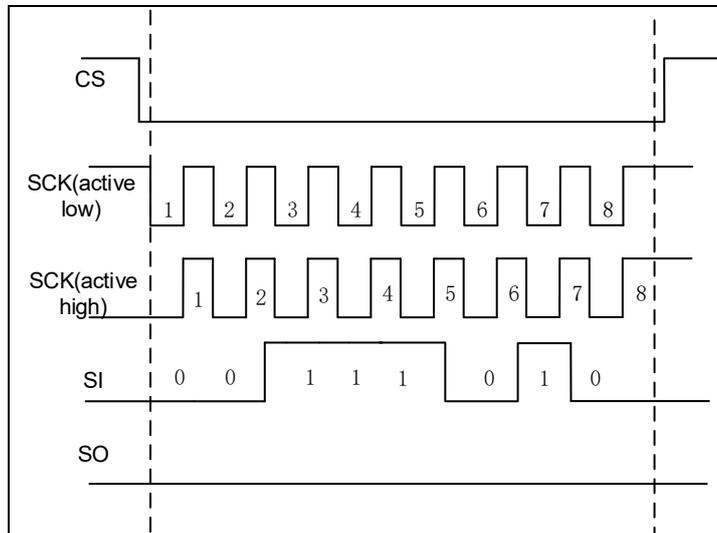
启用 OSPI 指令将操作模式更改为 OSPI。这个指令仅在 SPI 总线协议中支持，且仅支持最高 80MHz 的时钟频率。OSPI 总线协议不支持此指令。

首先通过将 CS 置为有效来选择 SPI 客户端接口。8 位 EOIO 指令，3Ah，逐位输入到 I/O[0]

引脚，每个时钟一位。将 CS 输入变为无效状态以结束周期。

[图 7-4. 使能 OSPI](#) 展示了使能 OSPI 指令。

**图 7-4. 使能 OSPI**



### 复位 QSPI

复位 QSPI / OSPI 指令将操作模式更改为 SPI。这个指令支持在时钟频率高达 80 MHz 的 SPI / QSPI / OSPI 总线协议中使用。

SPI / QSPI / OSPI 客户端接口是通过首先使将 CS 置为有效来选择的。8 位 RSTQIO 指令，FFh，在 SPI 模式下，通过 I/O[0]引脚，每个时钟输入一位；在 QSPI 模式下，通过 IO[3:0]引脚，每个时钟输入四位。CS 输入被设为无效状态以结束周期。

[图 7-5. SPI 模式复位 SPI](#) 展示了 SPI 模式下的重置 SPI 指令。

[图 7-6. QSPI 模式复位 QSPI](#) 展示了 QSPI 模式下的复位 QSPI 指令。

[图 7-7. OSPI 模式复位 OSPI](#) 展示了 OSPI 模式下的复位 OSPI 指令。

**图 7-5. SPI 模式复位 SPI**

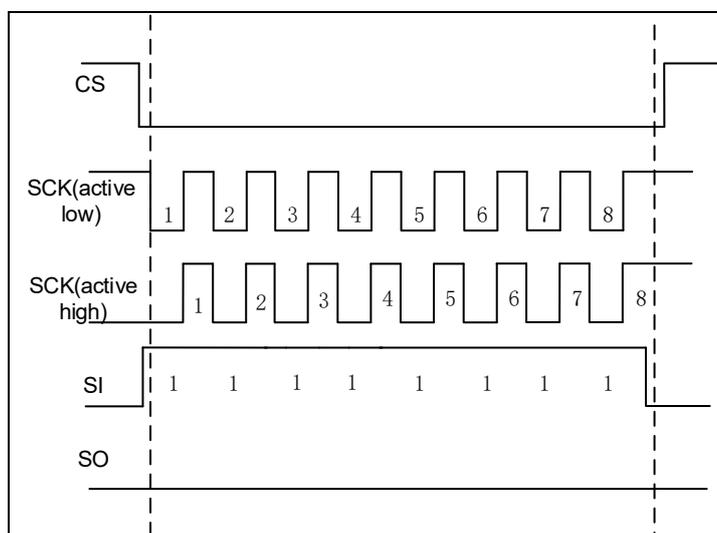


图 7-6. QSPI 模式复位 QSPI

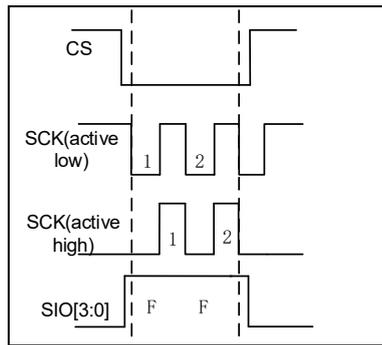
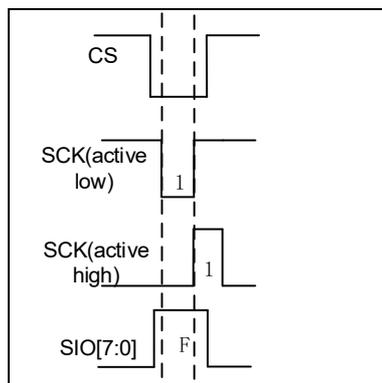


图 7-7. OSPI 模式复位 OSPI



## SPI 读命令

SPI / QSPI 客户端支持各种读命令。以下适用于所有读命令。

### 多次读取

除了首次读取外，通过将 CS 置为有效时继续时钟脉冲来执行额外的读取。地址的高两位指定自动递增（`address[15:14]=01b`）或自动递减（`address[15:14]=10b`）。内部 DWORD 地址根据这些位进行增加、减少或保持不变。保持固定的内部地址对于寄存器轮询很有必要。

### 读取

读指令输入指令代码和地址以及 dummy 字节，每个时钟位输入一个位，并以每个时钟位输出数据。在 QSPI 模式下，指令代码和地址以及 dummy 字节每个时钟位输入四个位，数据以每个时钟位输出四个位。该指令在 SPI 和 QSPI 总线协议中得到支持，时钟频率可达 80 MHz。

SPI / QSPI / OSPI 客户端接口的选择首先通过将 CS 置为有效来实现。在 SPI 模式下，将 8 位的 READ 指令 0Bh 输入到 IO[0] 引脚，然后是两个字节的地址和 1 个字节的 dummy 字节。在 QSPI 模式下，将 8 位的 FASTREAD 指令输入到 IO[3:0] 引脚，然后是两个字节的地址和 3 个字节的 dummy 字节。地址字节指定了设备内的一个字节地址。在 OSPI 模式下，将 8 位的 FASTREAD 指令输入到 IO[7:0] 引脚，然后是两个字节的地址和 8 个字节的 dummy 字节。地址字节指定了设备内的一个字节地址。

在最后一个 dummy 位（或半字节）的上升沿之后的下降沿，O/IO[1] 引脚开始驱动，从所选寄存器的最低有效位的高位开始。对于 QSPI 模式，IO[3:0] 从所选寄存器的最低有效位的高位开始驱

动。对于OSPI模式，IO[7:0]从所选寄存器的最低有效位的高位开始驱动。其余的寄存器位在随后的时钟下降沿上被移出。

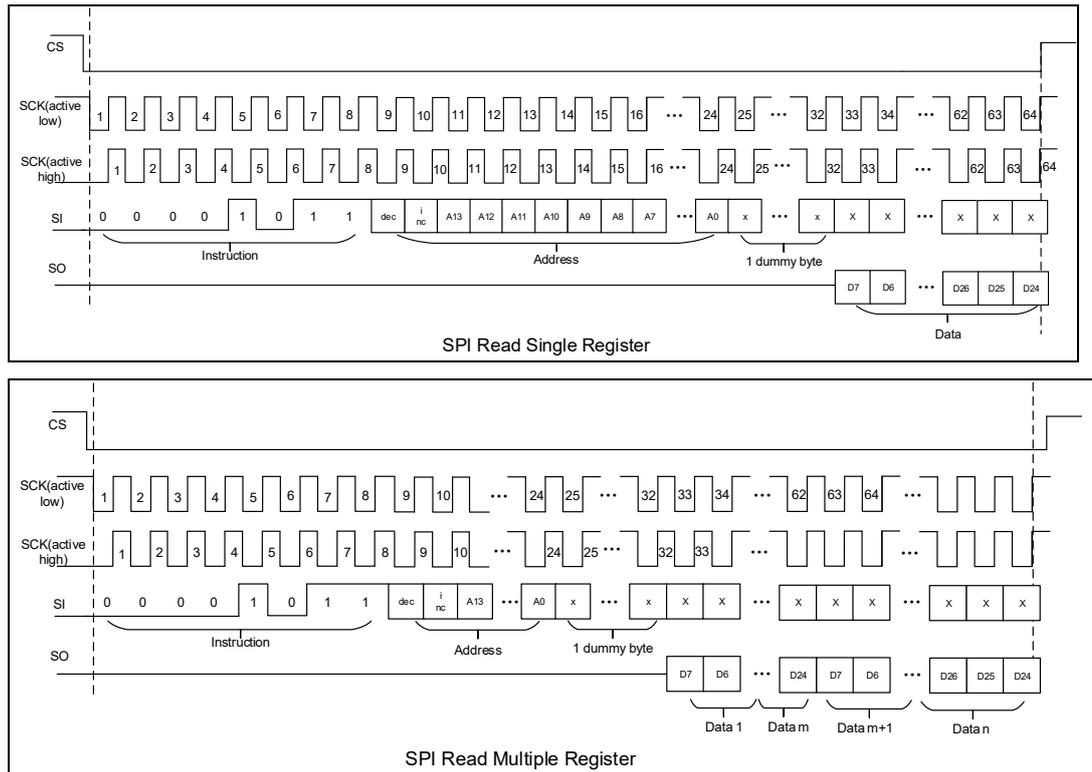
将CS输入被置为无效状态以结束周期。此时，O/IO [7:0] 引脚被置为高阻态。

[图 7-8. SPI 读取](#)展示了 SPI 模式下典型的单个和多个寄存器的快速读取。

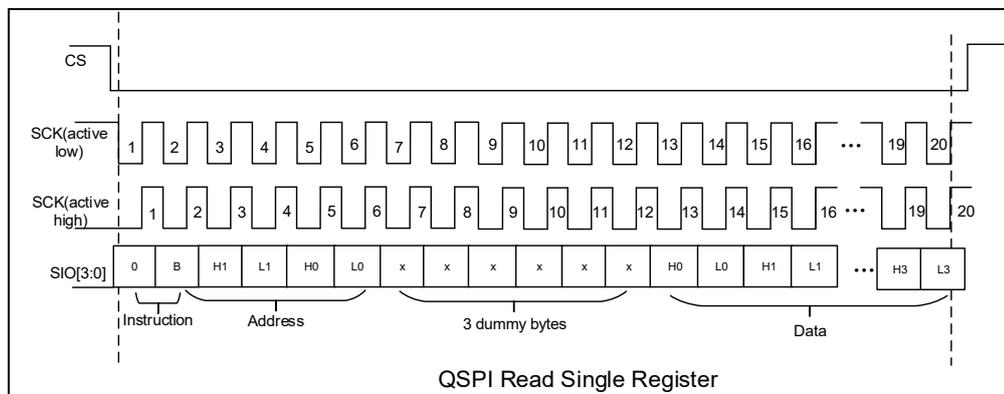
[图 7-9. QSPI 读取](#)展示了 QSPI 模式下典型的单个和多个寄存器的快速读取。

[图 7-10. OSPI 读取](#)展示了 OSPI 模式下典型的单个和多个寄存器的快速读取。

**图 7-8. SPI 读取**



**图 7-9. QSPI 读取**



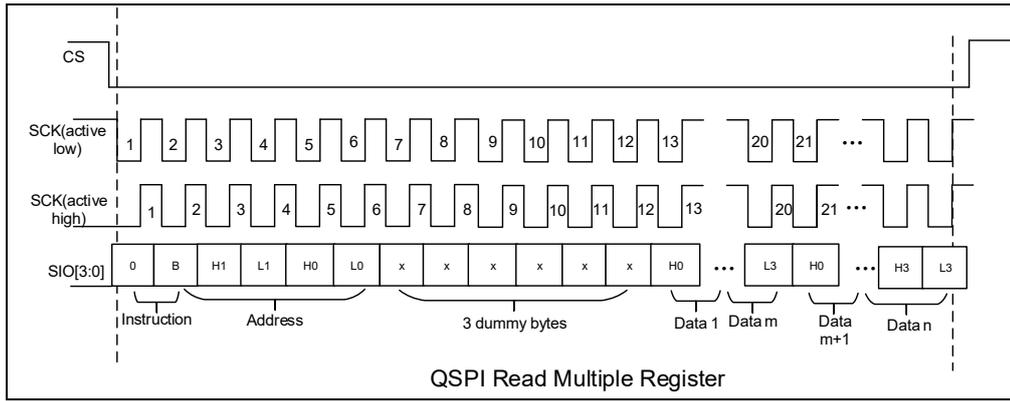
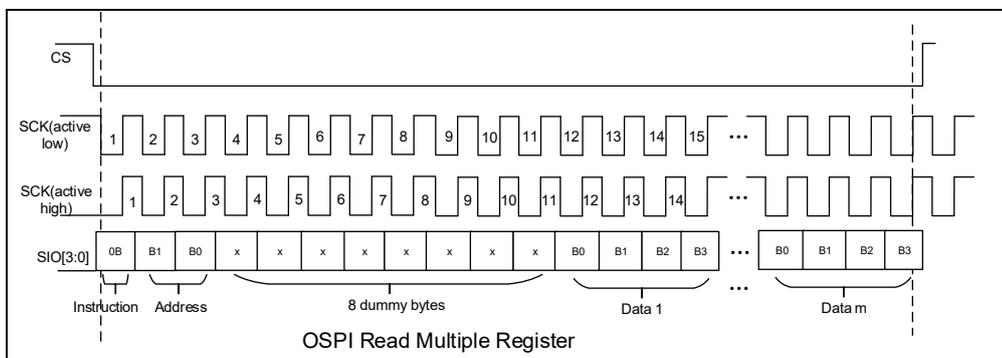
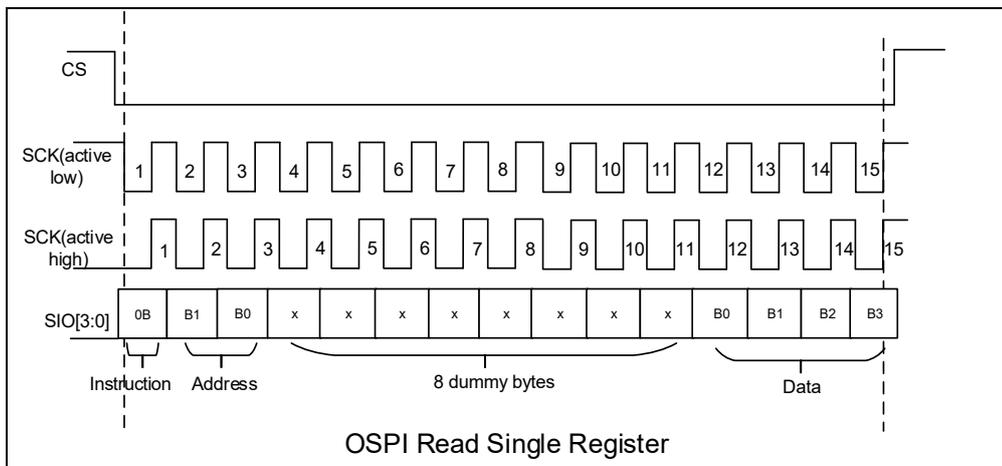


图 7-10. OSPI 读取



### 双输出读取

SPI双输出读取指令以每时钟一位的方式输入指令码和地址以及dummy字节，并以每时钟两位的方式输出数据。此指令仅支持SPI总线协议，并且仅支持高达80MHz的时钟频率。此指令不支持QSPI总线协议。

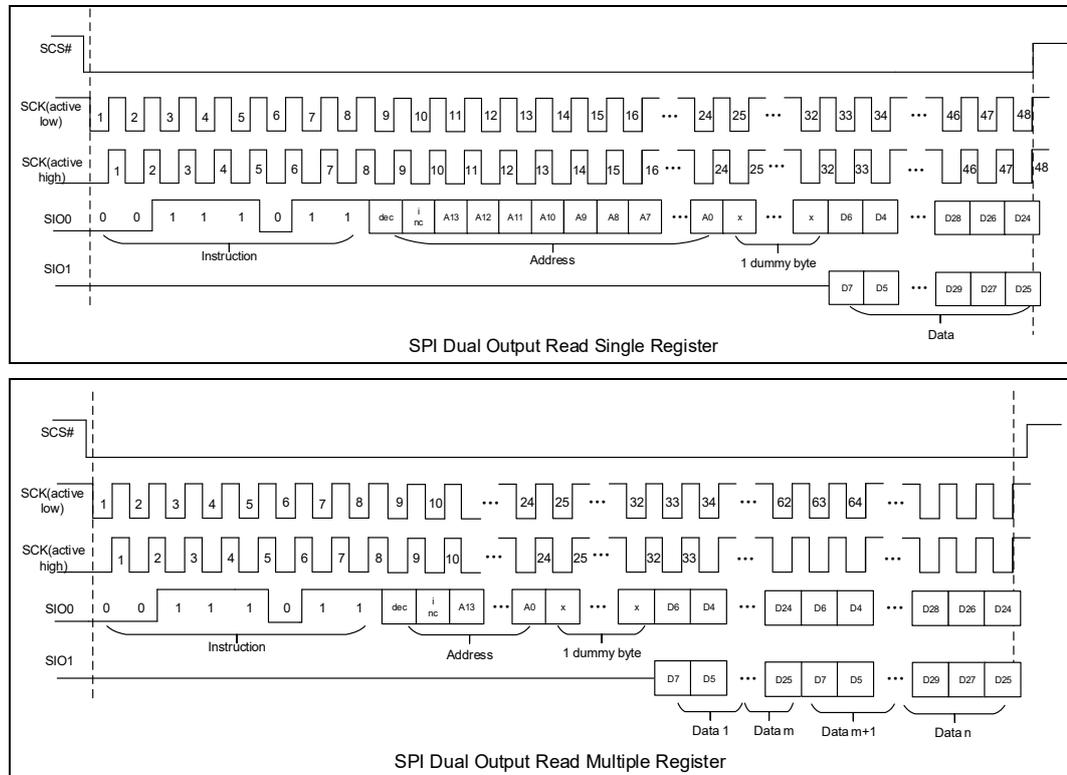
首先通过将CS置为有效信号来选择SPI客户端接口。然后输入8位SDOR指令，即3Bh，到IO [0]引脚，接着是两个字节的地址和一个字节的dummy数据。地址字节指定了设备内的一个字节地址。

在最后一个dummy二进制位的上升沿之后的下降沿，从选定寄存器的最低有效位的最高有效位开始驱动IO[1:0]引脚。其余的寄存器二进制位在随后的下降沿上被移出。

CS输入被置为无效状态以结束周期。此时IO [1:0] 引脚被置为高阻态。

[图 7-11. SPI 双输出读取](#)展示了典型的单个和多个寄存器的双输出读取。

**图 7-11. SPI 双输出读取**



### 四路输出读取

SPI四线输出读取指令以每时钟一位的方式输入指令码和地址以及dummy字节，并以每时钟四位的方式输出数据。这种指令只支持SPI总线协议，且时钟频率最高为80MHz。这种指令在QSPI总线协议中不支持。

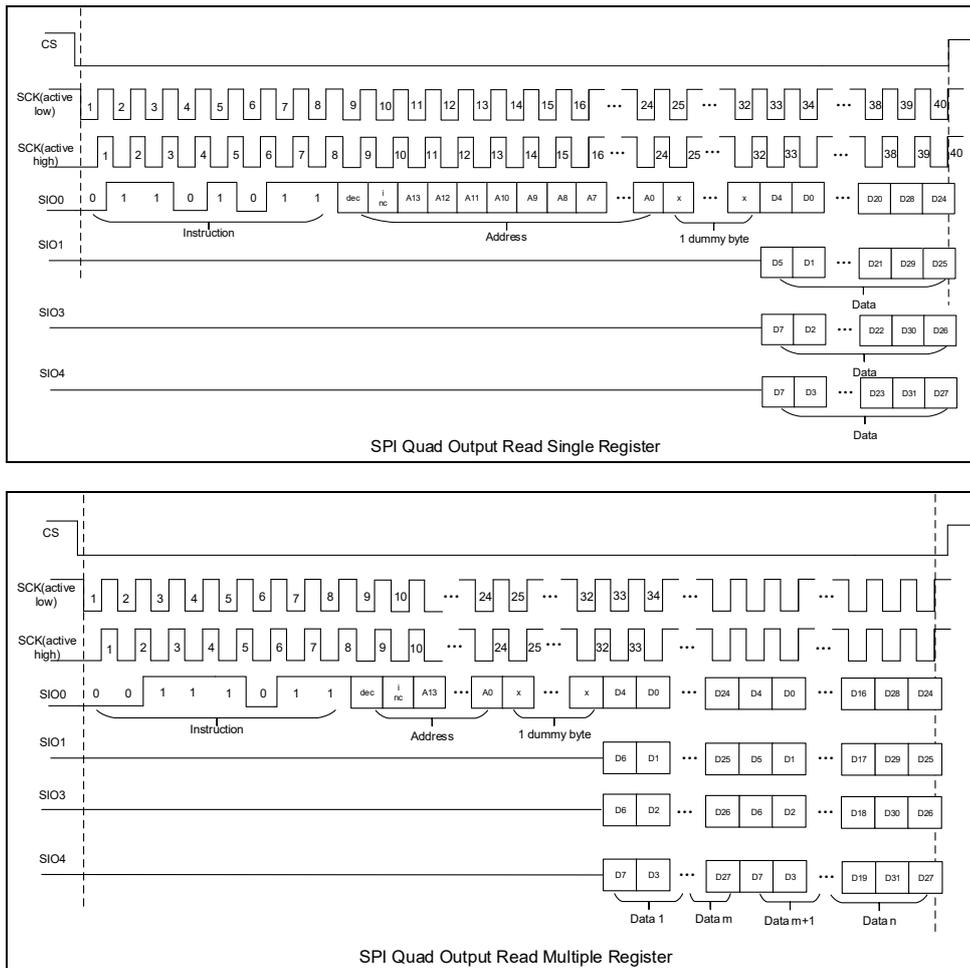
首先通过使CS置为有效状态来选择SPI客户端接口。然后输入8位的SQOR指令，6Bh，到IO [0] 引脚，接着是两个字节的地址和一个字节的dummy码。地址字节指定设备内的字节地址。

在最后一个dummy位的上升沿之后的下降沿，IO [3:0]引脚开始驱动，从选定寄存器的最低有效位的位开始。其余的寄存器半字节被依次移出。

CS输入被置为无效状态以结束周期。此时，IO [3:0]引脚被三态化。

[图 7-12. SPI 四线输出读取](#)展示了典型的单寄存器和多寄存器四线输出读取。

图 7-12. SPI 四线输出读取



### 双输入 / 输出读取

SPI双I/O读指令每时钟输入指令代码一位，地址和dummy字节两位，并每时钟输出数据两位。这种指令只支持在SPI总线协议中，且时钟频率最高可达80MHz。这种指令在QSPI总线协议中不支持。

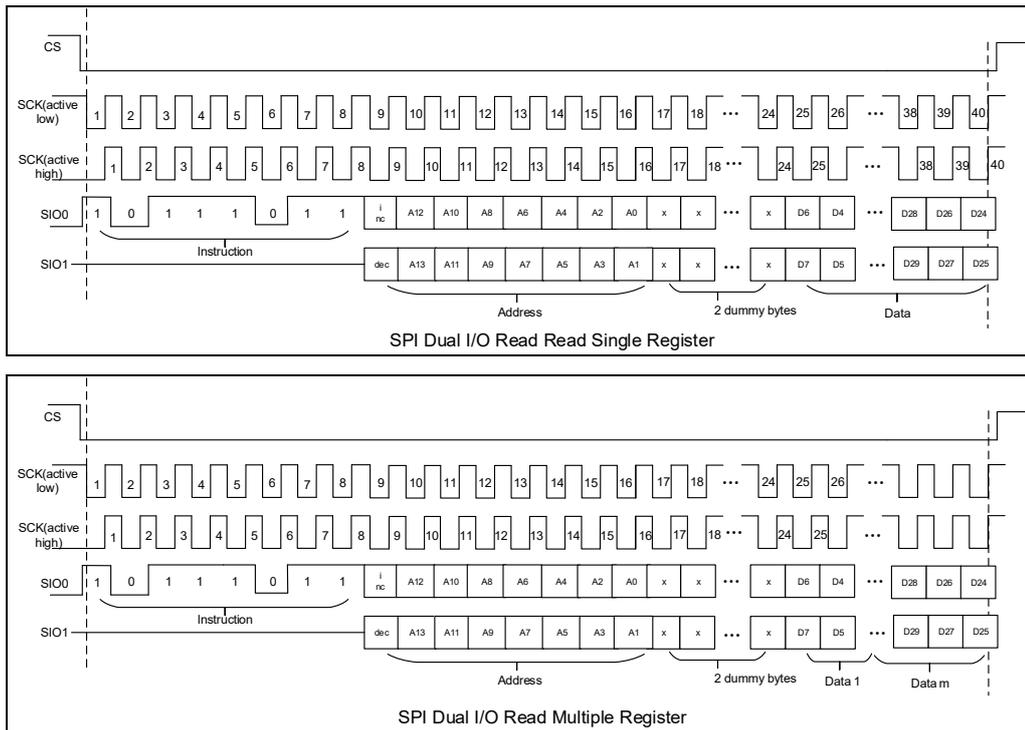
首先通过将CS置为有效来选择SPI客户端接口。然后，将8位的SDIOR指令，即BBh，输入到IO [0]引脚，接着将两个地址字节和2个dummy字节输入到IO [1:0]引脚。这些地址字节指定了设备内的一个字节地址。

在最后一个dummy二进制位的上升沿之后的下降沿上，IO [1:0]引脚开始驱动，从选定寄存器的最低有效位的高位开始。其余的寄存器二进制位在随后的下降时钟沿上被移出。

CS输入被置为无效以结束周期。此时，IO [1:0]引脚处于三态。

[图7-13. SPI 双 I/O 读取](#)展示了典型的单个和多个寄存器的双 I/O 读取。

图 7-13. SPI 双 I/O 读取



#### 四路输入 / 输出读取

SPI四线I/O读取指令以每个时钟一位输入指令码，地址和dummy字节以每个时钟四位输入，并以每个时钟四位输出数据。此指令仅在SPI总线协议中支持，且时钟频率最高为80 MHz。此指令在QSPI总线协议中不支持。

首先通过将CS置为有效来选择SPI客户端接口。然后输入8位的SQIOR指令EBh到IO [0]引脚，紧接着输入两个地址字节和4个dummy字节到IO [3:0]引脚。地址字节指定了设备内的一个字节地址。

在最后一个dummy半字节的上升沿之后的下降沿，IO [3:0] 引脚开始驱动，从所选寄存器的最低有效位的高位开始。其余的寄存器半字节在随后的下降沿上依次移出。

CS输入被置为非有效状态以结束周期。此时，IO [3:0]引脚被置为高阻态。

[图 7-14. SPI 四线 I/O 读取](#)展示了典型的单个和多个寄存器的双输出读取。

图 7-14. SPI 四线 I/O 读取



## SPI 写命令

SPI / QSPI客户端支持多个写命令。以下适用于所有写命令。

### 多次写入

通过保持时钟脉冲和输入数据的激活状态，可以连续进行多次读取。地址的高两位指定自动递增（ $\text{address}[15:14]=01\text{b}$ ）或自动递减（ $\text{address}[15:14]=10\text{b}$ ）。内部DWORD地址根据这些位进行递增、递减或保持不变。保持固定的内部地址可能对寄存器“位操作”或其他重复写入很有用。

### 写入

写指令输入指令代码和地址以及数据字节，每个时钟输入一位。在QSPI模式下，指令代码和地址及数据字节每个时钟输入四位。这种指令在SPI和QSPI总线协议中都得到支持，时钟频率可高达80MHz。

SPI/QSPI客户端接口是通过首先将CS置为有效来选择的。在SPI模式下，8位写入指令02h输入到I/O[0]引脚，然后是两个字节地址。在QSPI模式下，8位写入指令02h输入到IO[3:0]引脚，然后是两个字节地址。在OSPI模式下，8位写入指令02h输入到IO[7:0]引脚，然后是两个字节地址。地址字节指定设备内的一个字节地址。

数据跟随地址字节。对于SPI模式，数据输入到I/O[0]引脚，从最低有效位（LSB）的最高位开

始。对于QSPI模式，数据以半字节（nibble）宽度通过IO[3:0]输入，从最低有效位的最高位开始。对于OSPI模式，数据以半字节宽度通过IO[7:0]输入，从最低有效位的最高位开始。其余的位或半字节在随后的时钟边缘上移入。当32位数据输入完毕后，数据写入寄存器。如果CS返回高电平时，没有写入32位数据，则写入被认为是无效的，寄存器不受影响。

CS 输入被置为非有效状态以结束周期。

[图 7-15. SPI 写入](#)展示了 SPI 模式下典型的单个和多个寄存器写入。

[图 7-16. QSPI 写入](#)展示了 QSPI 模式下典型的单个和多个寄存器写入。

[图 7-17. OSPI 写入](#)展示了 OSPI 模式下典型的单个和多个寄存器写入。

图 7-15. SPI 写入

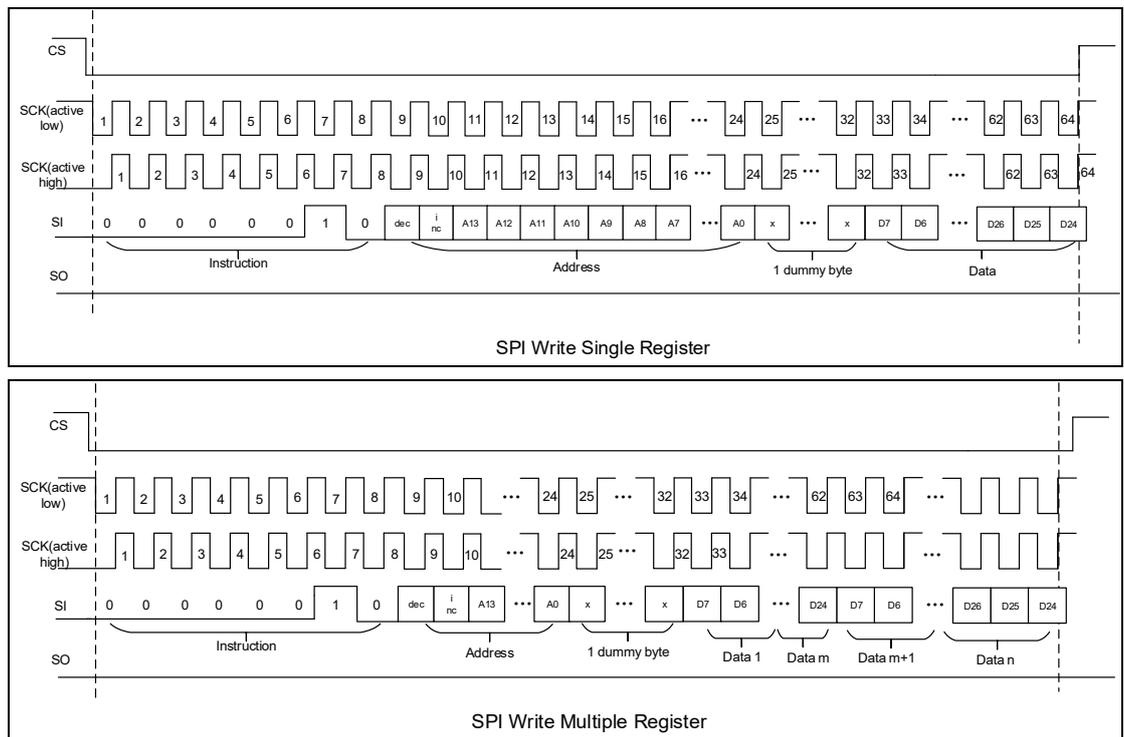
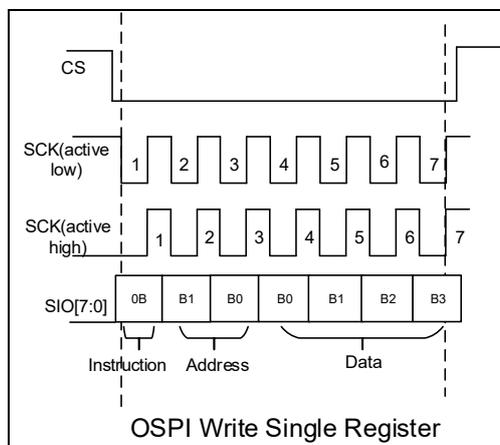


图 7-16. QSPI 写入



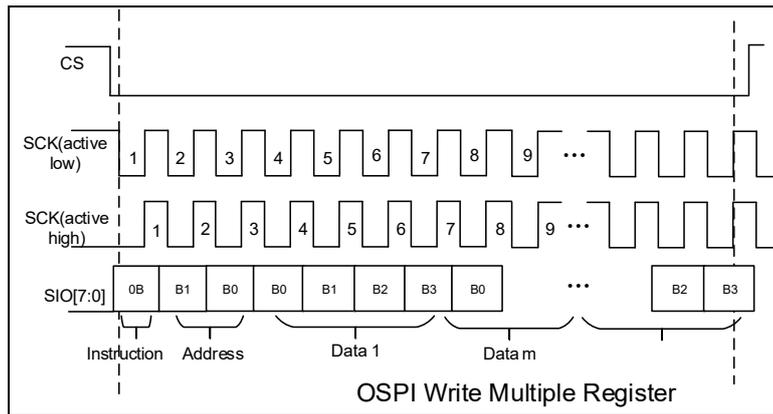
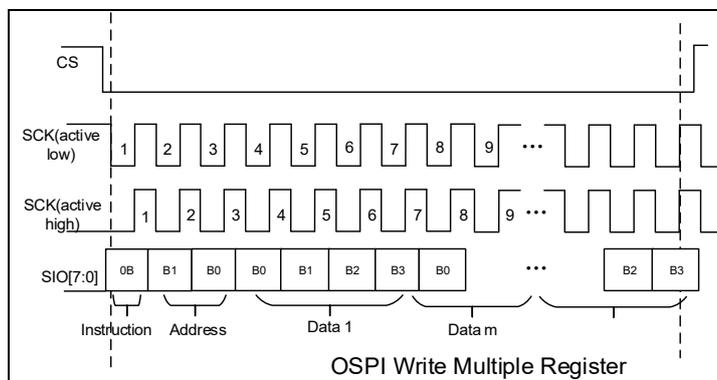
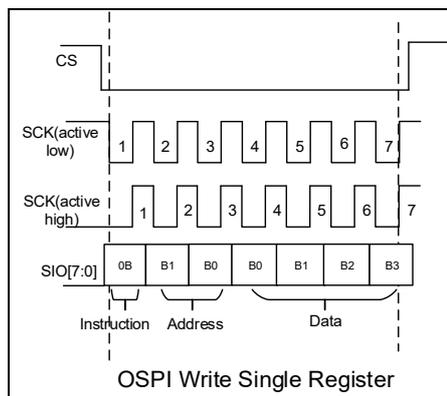


图 7-17. OSPI 写入



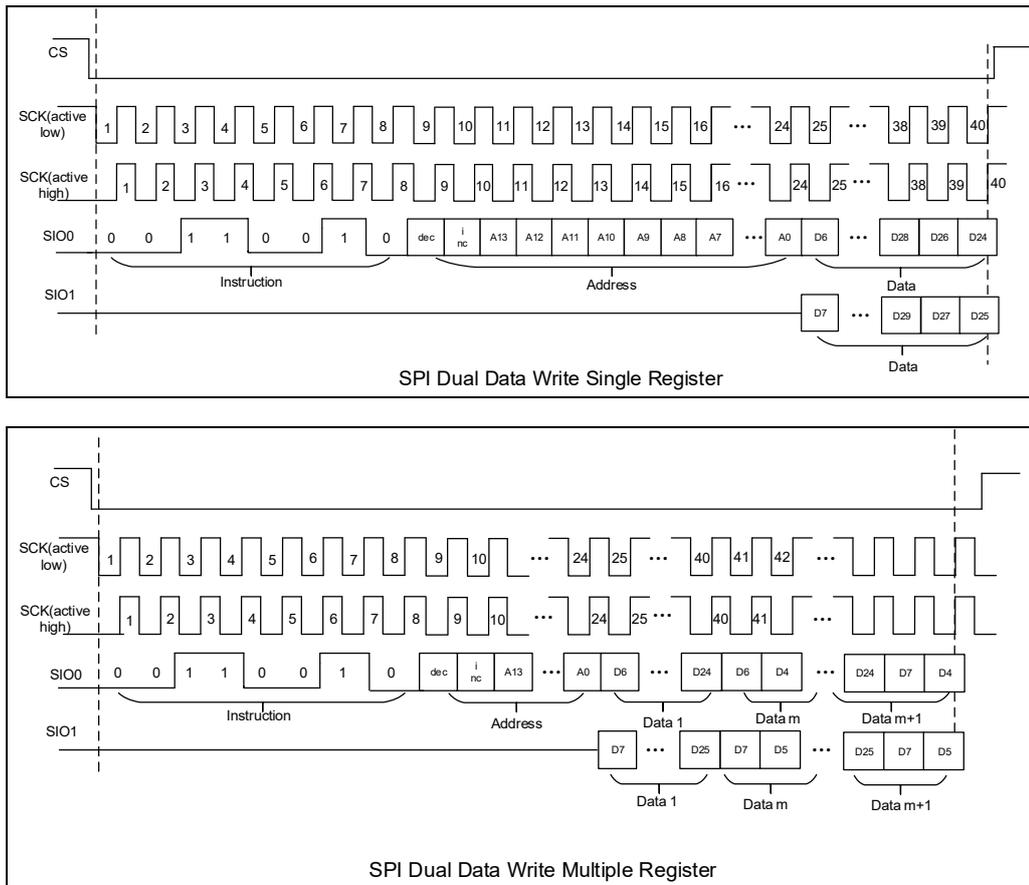
### 双数据读取

SPI双数据写入指令以每个时钟一位输入指令码和地址字节，以每两个时钟位输入数据。该指令仅在SPI总线协议中支持，且时钟频率最高可达80MHz。在QSPI总线协议中不支持此指令。

首先通过将CS置为有效来选择SPI客户端接口。将8位的SDDW指令，32h，输入到IO [0]引脚，然后是两个字节的地址。地址字节指定设备内的字节地址。数据跟在地址字节后面。数据输入到IO [1:0]引脚，从最低有效字节的最左边的位开始。其余的二进制位在随后的时钟边缘上移入。在输入32位后，数据被写入寄存器。如果CS返回高电平时没有写入32位，则写入被认为是无效的，寄存器不受影响。

CS 输入被置为非有效状态以结束周期。

[图7-18. SPI 双数据写入](#)展示了典型的单个和多个寄存器双数据写入。

**图 7-18. SPI 双数据写入**


#### 四线数据读取

SPI四线数据写指令以每个时钟一位输入指令码和地址字节，并以每个时钟四位输入数据。此指令仅在SPI总线协议中支持，且时钟频率最高可达80MHz。此指令在QSPI总线协议中不受支持。

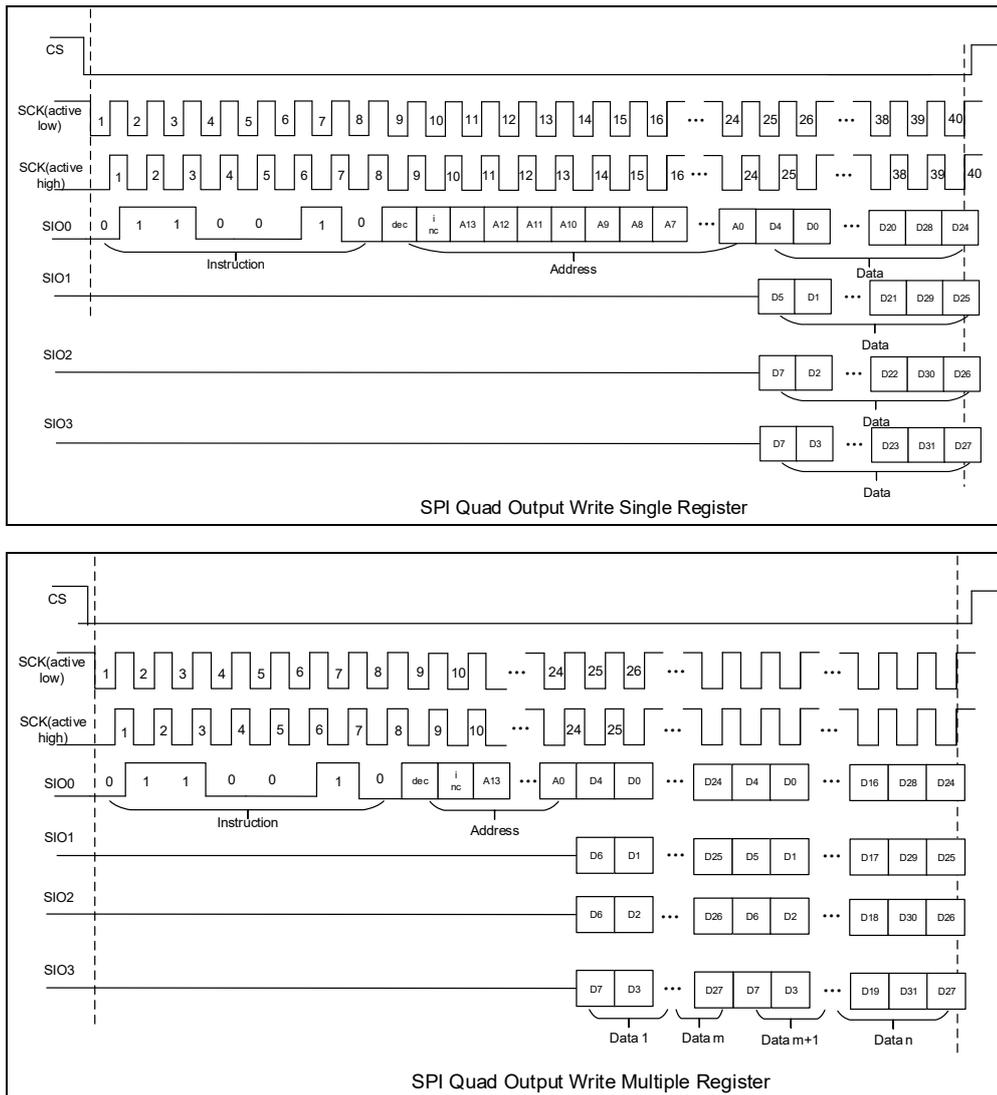
首先通过将CS置为有效来选择SPI从机接口。然后输入8位SQDW指令，62h，到IO[0]引脚，紧接着是两个地址字节。地址字节指定了设备内的一个字节地址。

数据跟随地址字节。数据从LSB的最高位开始，通过IO[3:0]引脚输入。剩余的半字节在随后的时钟边缘被移入。32位数据输入完成后，数据被写入寄存器。如果在CS返回高电平时没有写入32位，则写入被认为是无效的，寄存器不受影响。

CS输入被置为无效状态以结束周期。

[图7-19. SPI 四线数据写入](#)展示了典型的单个和多个寄存器的四线数据写入。

图 7-19. SPI 四线数据写入



### 双地址 / 数据写入

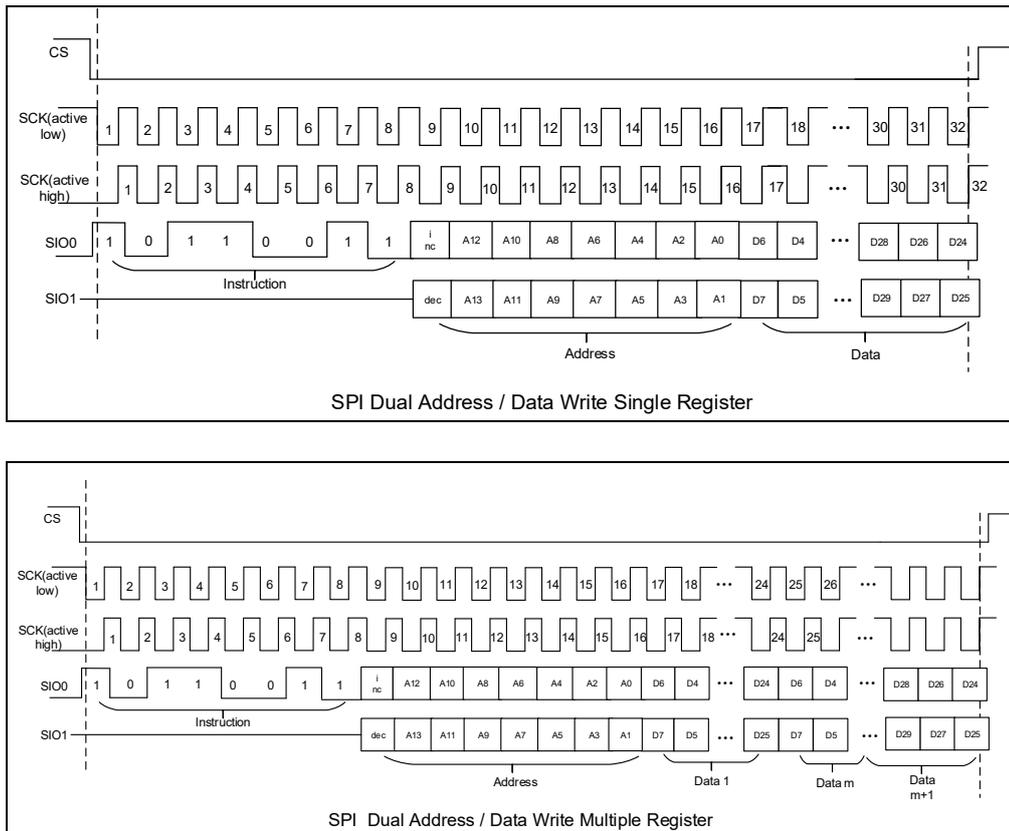
SPI双地址/数据写指令以每个时钟位输入指令码，以每个时钟两位输入地址和数据字节。这种指令只支持SPI总线协议，且时钟频率最高为80 MHz。这种指令在QSPI总线协议中不支持。

首先通过将CS置为有效来选择SPI客户端接口。然后，将8位的SDADW指令，即B2h，输入到IO[0]引脚，接着将两个地址字节输入到IO[1:0]引脚。这些地址字节指定了设备内的一个字节地址。

数据紧随地址字节之后。数据从IO[1:0]引脚输入，从最低有效字节的最高位开始。剩余的二进制位在随后的时钟边沿上被移入。当32位数据输入完成后，数据被写入寄存器。如果CS返回高电平时没有写入32位数据，则认为写入无效，寄存器不受影响。

CS 输入被置为非活动状态以结束周期。

[图 7-20. SPI 双地址/数据写入](#)展示了典型的单个和多个寄存器的双地址/数据写入。

**图 7-20. SPI 双地址/数据写入**


#### 四路地址 / 数据写入

SPI四地址/数据写指令以每时钟一位的方式输入指令码，以及每时钟四位的方式输入地址和数据字节。这种指令仅在SPI总线协议中支持，且仅支持最高80MHz的时钟频率。在QSPI总线协议中不支持此指令。

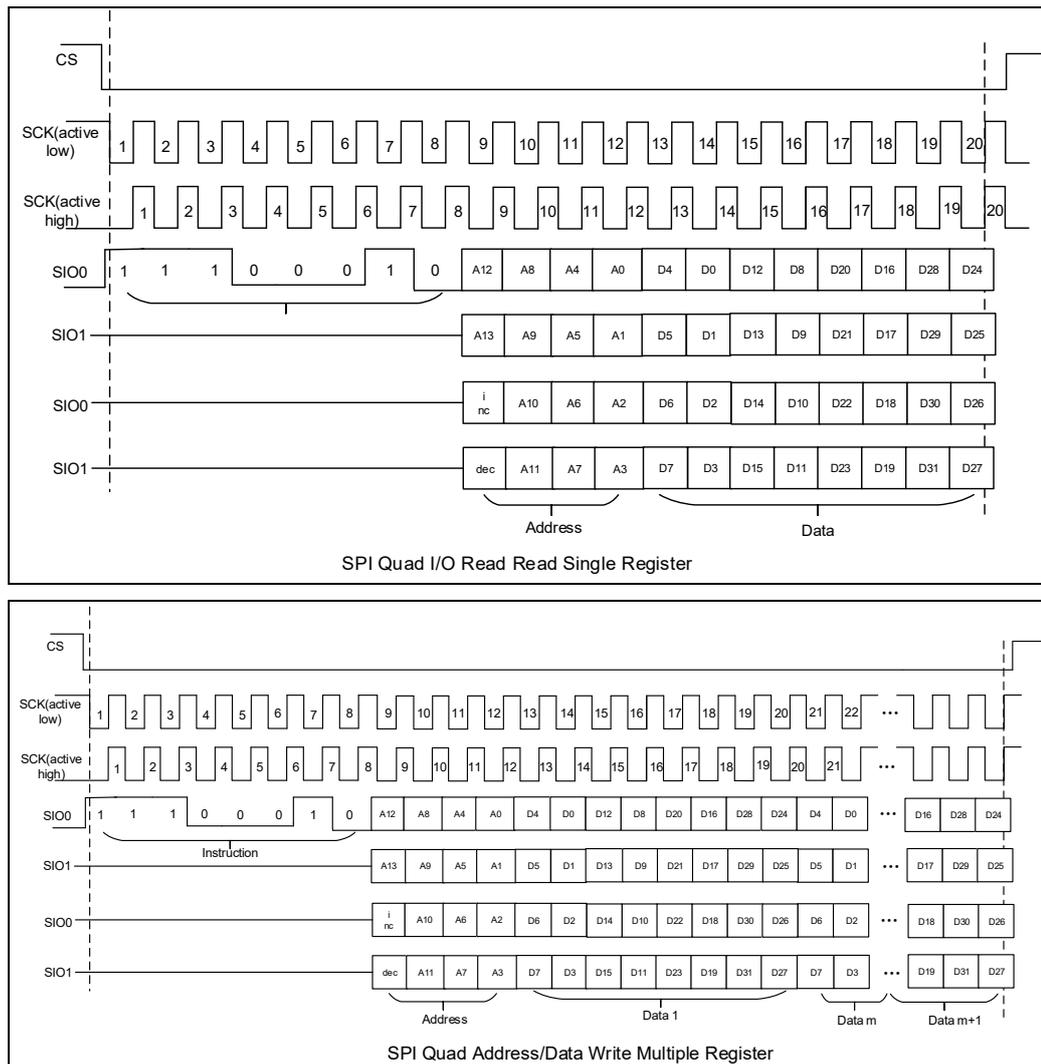
首先通过将CS置为有效来选择SPI客户端接口。然后输入8位的SQADW指令，E2h，到IO[0]引脚，随后输入两个地址字节到IO[3:0]引脚。这些地址字节指定了设备内的字节地址。

数据紧跟在地址字节之后。数据从最低位的高有效位开始，通过IO[3:0]引脚输入。其余的半字节在随后的时钟边缘上被移入。在32位数据输入完毕后，数据被写入寄存器。如果CS返回高电平没有写入32位，则写入被认为是无效的，寄存器不受影响。

CS 输入被置为非有效状态以结束周期。

[图 7-21. SPI 四路地址/数据写入](#)展示了典型的单个和多个寄存器四线地址/数据写入。

图 7-21. SPI 四路地址/数据写入



## SPI 唤醒系统

当芯片进入低功耗模式时，用户可以通过SPI / QSPI / OSPI访问BYTE\_TEST和READY寄存器以退出低功耗模式。

要确定主机接口是否正常工作，应轮询字节序测试寄存器（BYTE\_TEST）。一旦读取到正确的模式，就可以认为接口已经正常工作。此时，可以轮询设备就绪（READY）寄存器，以确定设备是否完全唤醒。

## SPI 访问 FIFO

SPI/QSPI/OSPI支持访问寄存器和FIFO。在访问FIFO的过程中，用户需要确保FIFO在访问时不会超出边界，否则会发生数据丢失。

BUS 模块提供了一些 FIFO 计数寄存器，包括 PRAM\_RD\_AVAIL\_CNT 和 PRAM\_WR\_AVAIL\_CNT，这些寄存器可以通过SPI总线读出。当用户读取TXFIFO时，读取的数据量应该小于或等于PRAM\_RD\_AVAIL\_CNT。当用户写入RXFIFO时，写入的数据量应该小于或等于PRAM\_WR\_AVAIL\_CNT。

在读取 TXFIFO 时，建议先读取 PRAM\_RD\_AVAIL\_CNT，然后再从 TXFIFO 中读取 PRAM\_RD\_AVAIL\_CNT 数量的数据。

在写入 RXFIFO 时，建议首先读取 PRAM\_WR\_AVAIL\_CNT，然后写入 PRAM\_WR\_AVAIL\_CNT 数据到 RXFIFO。

## 7.2. 外部存储器控制器（EXMC）

### 7.2.1. 概述

EXMC 模块提供了一个接口，用于连接 MCU 中的 EXMC 模块和 ESC 从控制器。它可以将 EXMC 的信号转换为 AHB 信号，以实现 MCU 和 ESC 之间的数据传输。

### 7.2.2. 主要特征

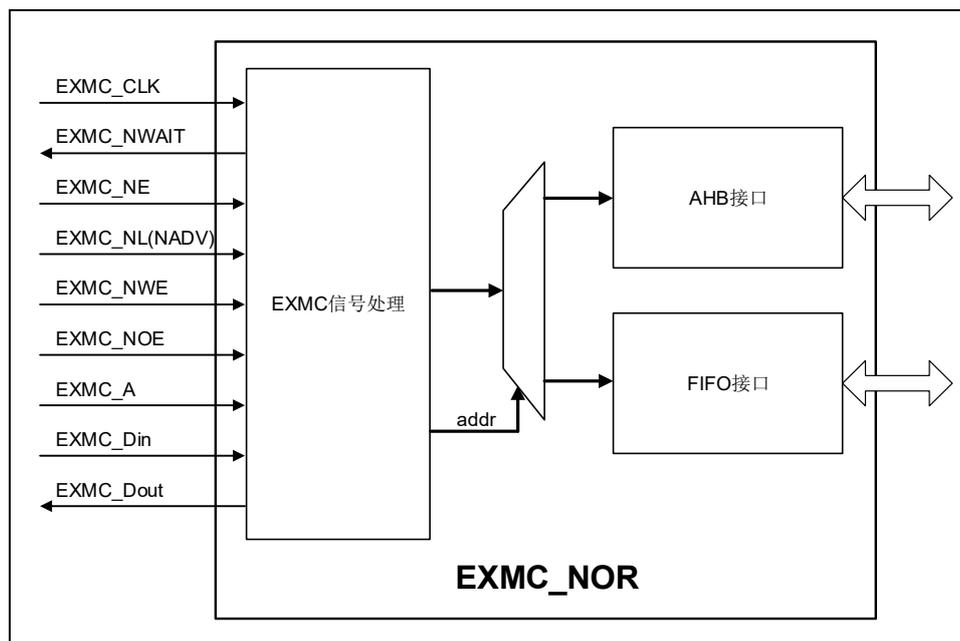
- 将 EXMC 信号转换为 AHB 信号以访问 ESC CCTL。
- 将 EXMC 信号转换为 FIFO 读写信号以访问 ESC 内核过程数据 RAM。
- 支持 EXMC 多路复用模式：8 位和 16 位；AHB：8 位、16 位和 32 位。
- 支持主机微控制器时钟高达 200MHz，从机以太网控制器时钟高达 80MHz。
- 通过外部存储器控制器（EXMC）支持手动设备唤醒。

### 7.2.3. 功能概述

#### 结构框图

EXMC 由五个模块组成：AHB 总线接口、EXMC 配置寄存器、NOR/PSRAM 控制器和外部设备接口。AHB 时钟（HCLK）是参考时钟。

图 7-22. EXMC 模块框图



## 基本传输

EXMC\_NOR 支持异步和同步两种模式。异步模式用于访问 ESC CCTL，同步模式用于访问异步 FIFO 以及在首次使用 EXMC\_NOR 之前配置 MCU 和 ESC 时钟比率。EXMC\_NOR 模块采样 MCU 发送的地址，并根据地址范围判断当前传输是异步还是同步。

### 异步模式

在异步模式下，所有 EXMC 输入在主机 MCU 的 HCLK 上升沿发生变化，而 ESC 将这些信号与系统时钟同步。因此，主机需要保持这些信号足够长的时间，以确保采样正确。

异步模式用于访问内核 ESC CCTL 和系统 ESC CCTL。每个 EXMC 传输可以是 8 位或 16 位，并支持转换为 8 位、16 位、32 位 AHB 传输。

表 7-7. EXMC 引脚和描述

引脚名称	描述
EXMC_NE	片选使能。在 EXMC 传输期间需要保持低电平。
EXMC_NL(NADV)	地址使能。表示发送的地址是有效的。
EXMC_NOE	输出使能。表示由 ESC 发送到 MCU 的数据是有效的。
EXMC_NWE	写使能。表示 MCU 写入到 ESC 的数据是有效的。
EXMC_Din	输入数据。在发送地址期间表示地址，在数据写入期间表示写入到 ESC 的数据。
EXMC_Dout	输出数据。表示由 ESC 发送到 MCU 的数据。

主机微控制器将通过外部存储器控制器(EXMC)向以太网控制器(ESC)发送地址和数据。地址建立时间 ASET 和数据建立时间 DSET 均为四个从机系统时钟周期，地址保持时间 AHLD 为一个从机系统时钟周期，以确保正确采样地址和数据。wdata\_samp\_wait\_cyc 可用于在不同主机和从机频率下增加写数据的采样等待时间。当地址和数据被采样时，就会启动 AHB 传输。在 GPIO 中，可以通过配置 EXMCTYPE 和 EXMCHSIZE 来确定 EXMC 和 AHB 的数据位宽。EXMCTYPE=1 是 EXMC 的 16 位模式，EXMCTYPE=0 是 8 位模式。HSIZE 在 SYSCFG 模块中配置。hsize=00、01 和 10 分别对应 AHB8 位、16 位和 32 位。

EXMC\_NOR 根据 EXMC 和 AHB 的位宽自动确定何时转换 AHB 传输。

当以 EXMCTYPE=0 和 AHB 为 32 位发起写传输时，EXMC\_NOR 将在第四次数据传输后完成 AHB 信号转换。

当 EXMCTYPE=0 时，启动 AHB 32 位读传输，数据在首次 EXMC 传输启动时从 AHB 总线读取，并且在四次传输中，每次向 EXMC\_NOR 主机传输 8 位数据。

**图 7-23. 异步写传输**展示了一个 16 位的 EXMC 写操作被转换为 16 位的 AHB 写操作。当 EXMC\_NOR 从主机采样地址时，拉低 NWAIT 信号，这样进入数据阶段后，主机 MCU 可以采样 NWAIT 信号以保持等待状态，当 AHB 传输结束时，EXMC\_NOR 释放 NWAIT。释放 NWAIT 后，系统时钟释放写功能 EXMC\_NWE 对三个主机启用，然后系统时钟释放片选信号 EXMC\_NE。

图 7-23. 异步写传输

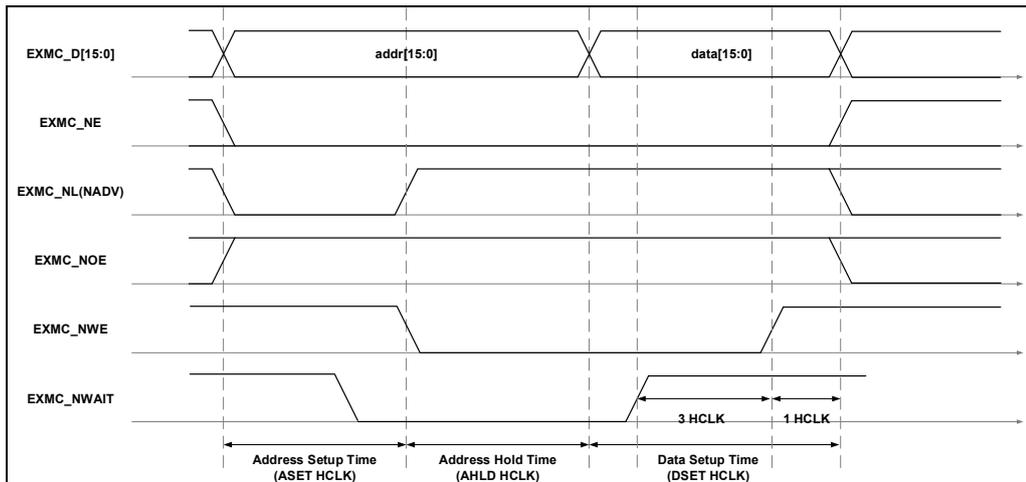


图 7-24. 异步读传输展示了异步读传输的时序图。nwait 机制和 DSET 配置是互斥的。当在创建读数据传输期间 nwait 被上拉时，无论 DSET 的配置如何，EXMC\_NOE 都会在 nwait 被上拉后将主机的系统时钟延长 4 个主机时钟周期。

图 7-24. 异步读传输

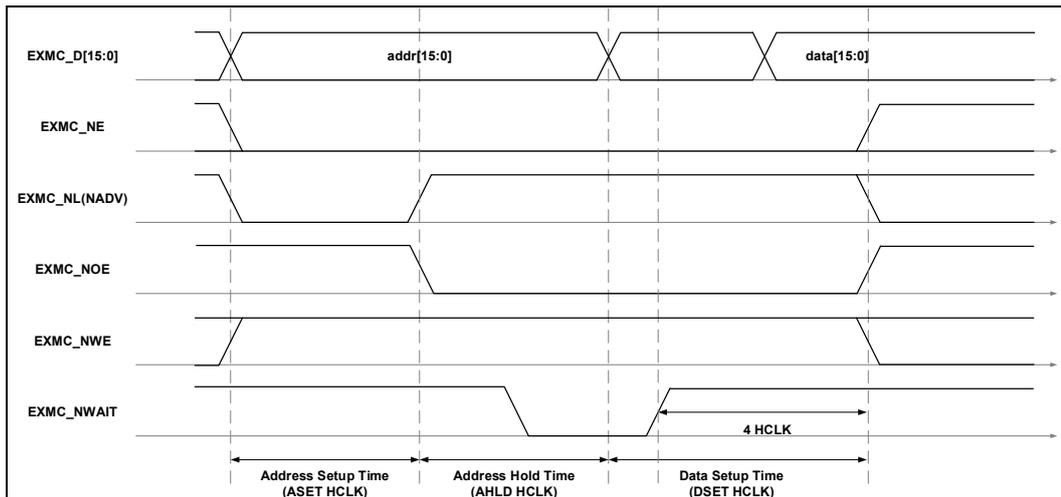
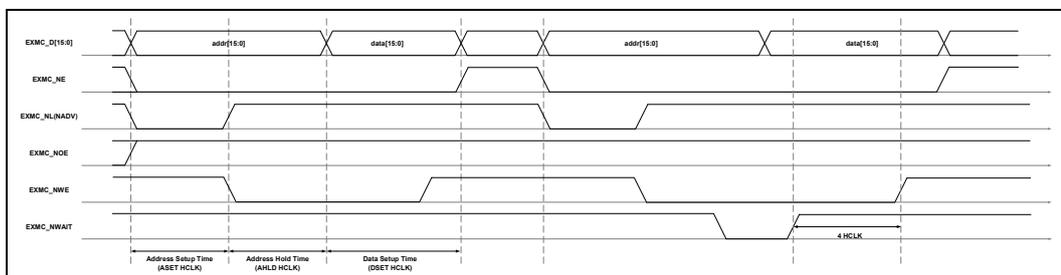


图 7-25. 包含 nwait 的背靠背传输显示包含 nwait 的背靠背传输。

图 7-25. 包含 nwait 的背靠背传输

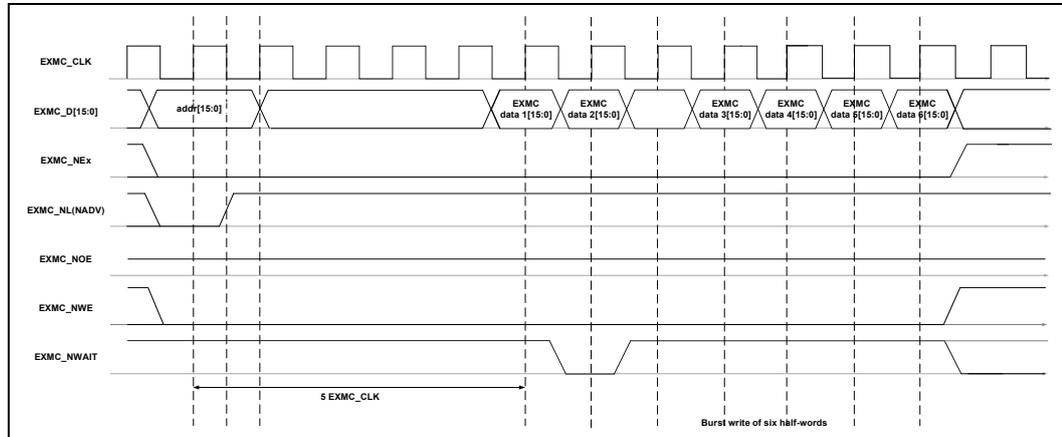


### 同步模式

EXMC\_NOR 使用同步模式访问发送数据到 ESC 内核的异步 FIFO。在同步模式下，所有由 MCU 发送的信号都随着 MCU 提供的 EXMC\_CLK 的下降沿变化。当 FIFO 满或读空时，使用 nwait 信号来保持 MCU 信号，使得 nwait 下拉的信号无效。

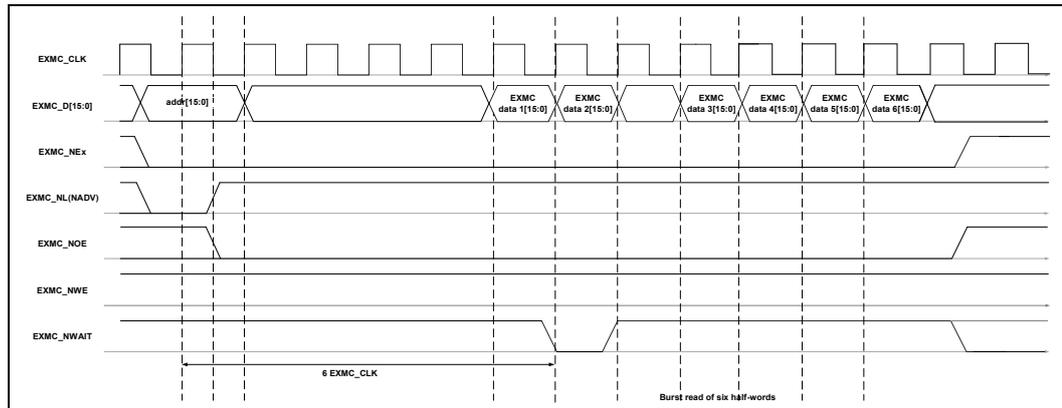
**图 7-26. 同步模式下写传输**显示了同步模式下的写传输。EXMC 写数据在地址采样后的 5 个周期被采样。每两个 EXMC 写操作被拼接成一个 32 位数据，并且启用 FIFO 写操作以完成 FIFO 写入。

**图 7-26. 同步模式下写传输**



**图 7-27. 同步模式下读传输**显示了同步模式下的读传输。当地址被采样时，主机 MCU 开始在六个周期后采样 EXMC 读取数据。在 16 位模式下，每个 FIFO 的读数据被分成两个 16 位数据，由 EXMC\_NOR 连续发送。

**图 7-27. 同步模式下读传输**



## 使用过程

在使用 EXMC\_NOR 之前，需要轮询电源管理单元（PMU）的数据接口参考值寄存器（PMU\_PDIREFVAL），以检查 EXMC\_NOR 是否可用。如果唯一值是 0x87654321，EXMC\_NOR 就可以正常使用。然后轮询 PMU 模块中的 READY 位。当该位设置为 1 时，整个设备就准备好使用了。

在使用 EXMC\_NOR 接口首次访问设备内部的寄存器和 FIFO 之前，需要将主机 MCU 系统时钟与从设备时钟的比例以同步模式写入到 SYSCFG 模块的 MCU\_HCLK\_FREQ 寄存器中。在同步模式下，系统发起一个写传输，地址为：0x3902，数据为：时钟比例。然后，可以在异步模式和同步模式下进行数据的读写。

## 唤醒功能

当 ESC 进入低功耗模式 D1、D2H 或 D3 时，可以通过 EXMC\_NOR 接口发起对寄存器 BYTE\_TEST 的写操作来手动唤醒设备。在此操作过程中，EXMC 的 AHLD 需要延长到 3 个主机 MCU 系统时钟。发送唤醒操作后，可以通过轮询 [数据接口参考值寄存器 \(PMU PDIREFVAL\)](#) 和 [控制寄存器 0 \(PMU CTL0\)](#) 的 Bit0 来确定 EXMC\_NOR 和整个设备是否可用。

## 8. 以太网 PHYS

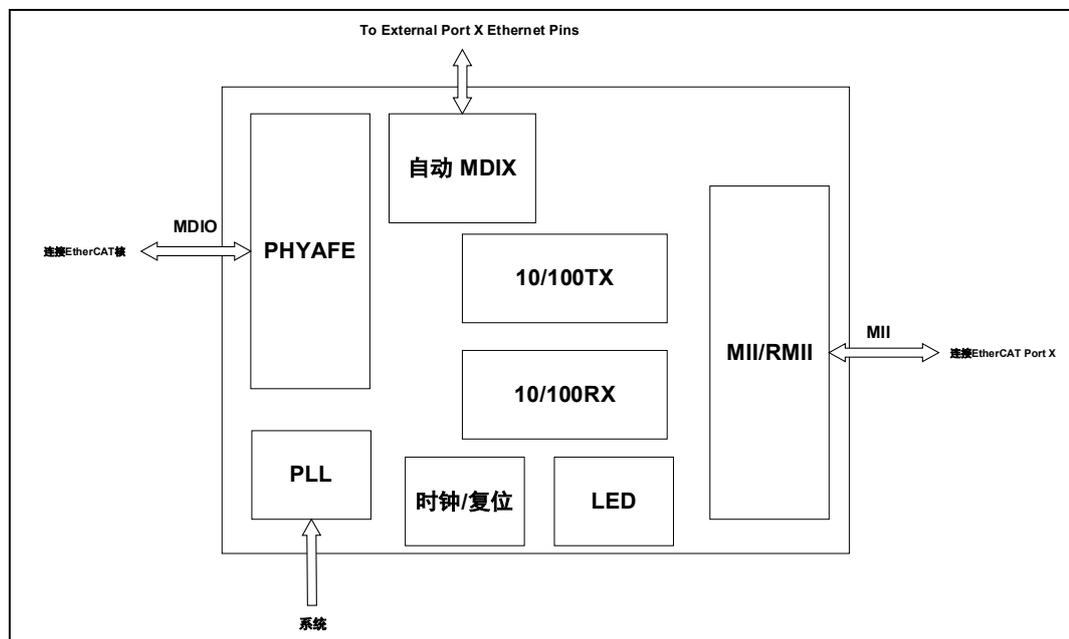
### 8.1. 简介

GDSCN 包含两个电压型 PHYA 和 B，它们在功能上是相同的。PHYA 连接到 EtherCAT 端口 0 或 2。PHY B 连接到 EtherCAT 端口 1。这些 PHY 通过内部 MII 接口与它们各自的 MAC 进行接口。可以配置为全双工 100 Mbps（100BASE-TX）以太网操作。所有 PHY 寄存器遵循 IEEE 802.3 规定的 MII 管理寄存器，并且是完全可配置的。

### 8.2. 主要特征

- 完全符合 IEEE 802.3 100 Base-TX 标准，并支持EEE；
- 自动协商和并行检测能力，用于自动速度和双工选择；
- 支持MII接口；
- 10Base-T的自动极性校正；
- 支持即插即用的Auto-MDIX功能；
- 可编程环回模式用于诊断；
- 支持不同应用的可编程LED输出，开机自检LED；
- 支持WOL（Wake-On-Lan）功能。

图 8-1. PHY 功能模块图



## 8.3. 功能概述

### 8.3.1. 操作模式

在 100BASE-TX 中,发送端将来自 MAC 接口的数据流进行 4B/5B 编码、串行化、扰乱和 MLT3 编码。在接收端,来自介质的数据流被恢复,从 MLT3 解码,去扰乱,并行化,并进行 5B/4B 解码,解码成 4 位数据。

当没有数据需要传输时,系统会通过低功耗空闲(LPI)信号通知其链路伙伴,然后进入发送器省电模式。另一方面,当从链路伙伴接收到 LPI 信号时,系统会进入接收机省电模式。只有周期的信令用于保持链路活跃。

### 8.3.2. MII 接口

MII (Media Independent Interface)是 IEEE 802.3u 中定义的 MAC 和 PHY 之间的接口。对于 10Mbps 传输,时钟频率为 2.5MHz。对于 100Mbps 传输,时钟频率为 25MHz。MAC 以 TXCLK 和 RXCLK 同步传输和接收数据,这些时钟由 PHY 生成。

TXEN 生效时, TXD[3:0]被 PHY 接受进行传输。在 TXEN 有效时, TXER 有效表示传输编码错误。TXEN 无效时, TXER 生效, TXD[3:0]等于 0001 的组合显示请求进入(或保持)低功耗状态。TXEN、TXER 和 TXD[3:0]与 TXCLK 同步采样。

当 RXDV 生效时, RXD[3:0]将从 PHY 恢复的数据传输到 MAC。RXER 表示接收错误。RXDV 无效, RXER 有效,且 RXD[3:0]等于 0001 的组合,通知 LPI 客户端(例如 MAC)链路伙伴处于低功耗状态。CRS 在 PHY 发送或接收时生效。COL 在 PHY 检测到冲突时生效。RXDV、RXER 和 RXD 与 RXCLK 同步。

### 8.3.3. SMI 接口

串行管理接口(SMI)可用于在站点管理(STA)和物理层(PHY)之间传输控制和状态信息。用户还可以通过 SMI 访问 PHY 的内部寄存器设置。MDIO 是双向信号,主要由命令(读/写)字段和数据字段组成,并与 MDC 同步。当没有驱动信号时,MDIO 引脚应该上拉。

### 8.3.4. 自动 MDI/MDIX 和极性配置

自动 MDI/MDIX 配置目的在消除两个设备之间外部交叉电缆的需要。PHY 可以自动进行 MDI/MDIX 配置,以使传输和接收正常工作。MDI/MDIX 配置也可以通过手动设置寄存器来确定。

PHY 可以自动校正电缆对上的极性错误。

### 8.3.5. 环回模式

环回模式提供了发送和接收的诊断功能,可以对发送和接收的数据路径进行测试。

### 8.3.6. Wake-On-LAN

Wake-On-LAN 通过使用一种特殊的网络消息来实现，这种消息被称为“魔术包”。魔术包包含目标设备的 MAC 地址。监听设备等待一个合法的、针对它的魔术包，然后激活系统唤醒程序。

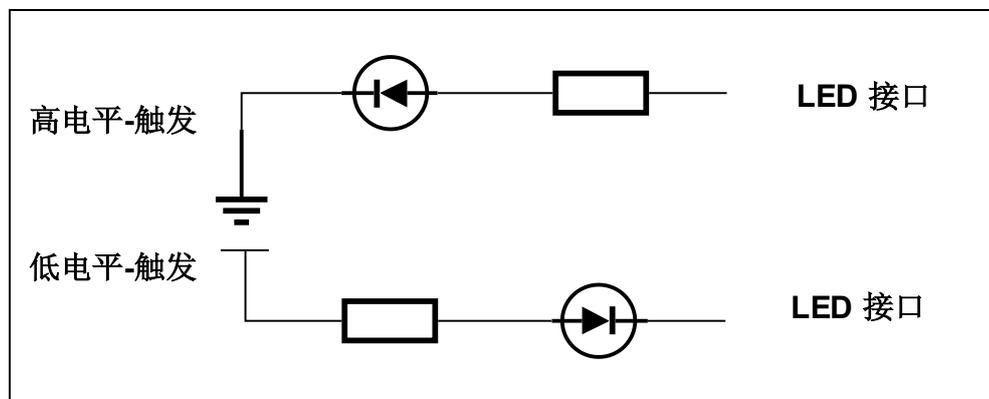
当启用了远程唤醒功能时，物理层在接收到合法的魔术包后会发送一个中断。

### 8.3.7. LED 模式

有 3 个 LED 接口用于控制链路状态、速度和双工模式指示的 LED 状态。

有两种 LED 连接类型：高电平触发和低电平触发，如图 82 所示：

图 8-2. LED 连接图



当 LED 接口被连接为高电平触发类型，接口 PHY\_LED\_POL 应该被置为 0。

当 LED 接口被连接为低电平触发类型，接口 PHY\_LED\_POL 应该被置为 1。

LED 状态信息的定义如下：

- 链接LED：
  - 打开：链接已激活
  - 关闭：链路已断开
  - 刷新：数据传输
- 速度LED
  - 打开：100M
  - 关闭：10M
  - 刷新：N/A
- 双工LED
  - 打开：全双工
  - 关闭：半双工
  - 刷新：碰撞

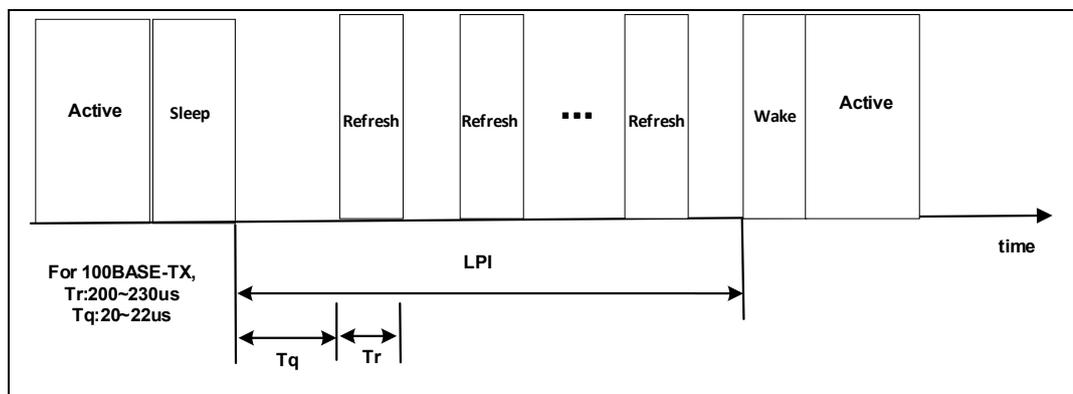
LED 的刷新周期是 33 毫秒。

### 8.3.8. LPI 信令

当 LPI 客户端发出 LPI 请求时，物理层会发送 **Sleep** 符号，以告知其链路伙伴本地物理层即将进入 LPI 状态。物理层在发送 **Sleep** 符号后进入 LPI 状态。在 LPI 状态下，只定期传输刷新符号。当 LPI 客户端请求退出 LPI 状态时，物理层会发送唤醒符号，以请求链路伙伴唤醒以进行进一步传输。

当接收到来自其链路伙伴的 **Sleep** 符号时，PHY 知道远端 PHY 将进入 LPI 状态。在远端 PHY 停止传输后，本地 PHY 可以关闭一些电路以节省电力。在低 LPI 状态下，物理层使用 **Refresh** 符号来更新其过滤系数并调整时序。当从其链路伙伴接收到 **Wake** 符号时，物理层在指定的恢复时间之前从 LPI 状态回到正常操作。

图 8-3.100Base-TX LPI



## 8.4. PHY 寄存器

### 8.4.1. Page0 寄存器

#### PHY 控制寄存器 (PHY\_MII\_CTL)

地址偏移: 0x00

复位值: 0x3100

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR_MAIN_REST	LOOPBACK_EN	FORCE_SLEEP(B)	MR_AUTO_NEG_EN	POWERDOWN	保留	MR_RESET_ONEG	FORCE_DUPLEX	COL_TEST	FORCE_SLEEP(B)	UNIDIRECTIONAL_EN	保留				
r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w					

位/位域	名称	描述
15	MR_MAIN_REST	主要复位 将 PHY 的状态和控制寄存器复位为默认值并自清除 1: 复位

		0: 正常
14	LOOPBACK_EN	环回使能 在环回模式下，工作模式将被设置为全双工模式，自动关闭自协商功能
13	FORCE_SPEED(LSB)	LSB 强制速度 位 13 和位 6 结合速度选择。它仅在 MR_AUTONEG_EN = 0 时有效。启用光纤模式后，内部速率自动设置为 100M，忽略该字段设置。 00: 10M 01: 100M 1X: 保留
12	MR_AUTONEG_EN	自协商使能 该位的结果应与 I_FXEN 和 EN_FX 进行 OR 'ed，以确定内部最终的自动协商使能信号(自动协商将在光纤中禁用模式)。注意 LOOPBACK_EN 设置为 1
11	POWERDOWN	下电模式 高电平有效编程 PHY 进入断电(断电状态下行模拟 TX，模拟 RX，模拟 AD)
10	保留	必须保持复位值。
9	MR_RESTART_AUTONEG	重启自协商 此位写入 1 时，该为由 PHY 自动清除，写入 0 时不受影响 1: 重启自协商 0: 正常
8	FORCE_DUPLEX	强制工作模式 此位仅在 MR_AUTONEG_EN = 0 时有效。该位的结果应与 LOOPBACK_EN 进行匹配，以确定内部最终双工能力 1: 全双工（默认） 0: 半双工
7	COL_TEST	冲突测试 当该位被置为有效时，PHY 响应 TX_EN 有效将响应 512BT 内的 COL 信号，当连接到 MII 或连接到 GMII 时，响应 TX_EN 的有效将在 4BT 或 16BT 内生效
6	FORCE_SPEED(MSB)	MSB 强制速度 参考 bit13
5	UNIDIRECTIONAL_EN	单向使能 启用从 MII / GMII 接口编码和传输数据的能力，而不管 PHY 是否确定已建立有效链路。此功能仅在禁用“自动协商”且工作模式为全双工时有效
4:0	保留	必须保持复位值。

## PHY 状态寄存器 (PHY\_MII\_STATUS)

地址偏移: 0x01

复位值: 0x79C9

该寄存器可以按半字（16位）。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
100BASE _T4	100BASE- X_FULL_ DPX	100BASE- X_HALF_ DPX	10BASE- T_FULL_D PX	10BASE- T_HALF_ DPX	100BASE _T2_FULL_ DPX	100BASE T2_HALF_ DPX	EXTENDE D_STATU S	UNIDIREC TIONAL_A BLT	MF_PRB_ SUP	MR_AUTO NEG_CPL T	REMOTE_ FAULT	AUTONE G_ABLT	LINK_STA D_TUS	JABBER_ DETECT	EXTENDE D_CAP
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位/位域	名称	描述
15	100BASE_T4	100BASE T4性能 不支持，始终保持为0
14	100BASE-X_FULL_DPX	支持100BASE TX全双工 1: PHY支持100BASE TX全双工 0: PHY不支持100BASE TX全双工
13	100BASE-X_HALF_DPX	支持100BASE TX半双工 1: PHY支持100BASE TX半双工 0: PHY不支持100BASE TX半双工
12	10BASE-T_FULL_DPX	支持10BASE-T全双工 1: PHY支持10BASE-T全双工 0: PHY不支持10BASE-T全双工
11	10BASE-T_HALF_DPX	支持10BASE-T半双工 1: PHY支持10BASE-T半双工 0: PHY不支持10BASE-T半双工
10	100BASE_T2_FULL_DPX	100BASE T2全双工 不支持，始终保持为0
9	100BASE_T2_HALF_DPX	100BASE T2半双工 不支持，始终保持为0
8	EXTENDED_STATUS	扩展状态 1: 寄存器15中的扩展状态信息 0: 在寄存器15中没有扩展状态信息
7	UNIDIRECTIONAL_ABLT	单向能力 1: PHY能够从MII/GMII传输，无论PHY是否确定已经建立了有效的链路 0: 只有当PHY确定已经建立了有效的链路时，PHY才能从MII/GMII传输
6	MF_PRB_SUP	消除前导码能力 1: PHY可以接受带有前导码消除的管理帧 0: PHY不接受带有前导码消除的管理帧
5	MR_AUTONEG_CPLT	完成自协商 1: 自协商完成 0: 自协商未完成
4	REMOTE_FAULT	远程故障检测

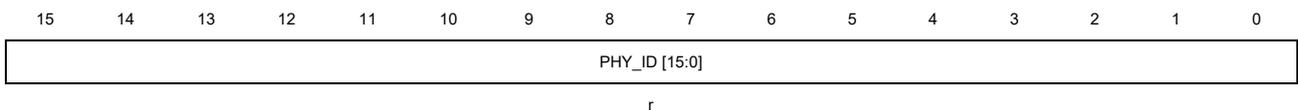
		1: 远程故障 0: 正常
3	AUTONEG_ABLT	自协商能力 1: PHY 可以自协商 0: PHY 不可以自协商
2	LINK_STATUS	连接状态 1: 建立连接 0: 断开连接 注意: 当PHY被设置为环回模式, <a href="#">PHY控制寄存器 (PHY MII CTL)</a> BIT14 (LOOPBACK_EN被设置为1), PHY被强制连接。为了使MAC正常工作, 链路状态应根据此位进行更新。这个位是用一个高锁存功能来实现的, 这样, 链路故障条件的发生将导致这个位被清除并保持清除, 直到它被读取。
1	JABBER_DETECT	检测到无效情况 1: 检测到 RX 和 TX 无效情况 0: 正常
0	EXTENDED_CAP	扩展寄存器功能 1: 扩展寄存器功能 0: 仅基本寄存器设置功能

### PHY 标识寄存器 (PHY\_ID\_REG)

地址偏移: 0x02

复位值: 0x0044

该寄存器可以按半字 (16位)。



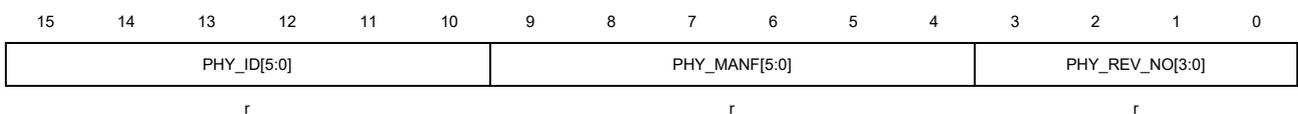
位/位域	名称	描述
15:0	PHY_ID [15:0]	PHY标识bit位[31-16] OUI (bits 3-18). OUI =00-11-05

### PHY 版本寄存器 (PHY\_VER\_REG)

地址偏移: 0x03

复位值: 0x1400

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识 bit 位 [5-0] OUI bits 19-24
9:4	PHY_MANF[5:0]	制造商型号 制造商的型号(bit5-0)，其中[5:4]为架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0)寄存器 <a href="#">PHY 标识寄存器 (PHY ID REG)</a> ，bit 0 是 PHY 标识符 LS 位

### 自动协商广播寄存器 (PHY\_AUTONEG\_ADV)

地址偏移: 0x04

复位值: 0x0DE1

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE_EN	保留	REMOTE_FAULT_EN	EXTENDED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASE_TX_FULL_DPX	100BASE_TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SLCT_FLD[4:0]				
r/w		r/w	r/w	r/w	r/w	r	r/w	r/w	r/w	r/w	r				

位/位域	名称	描述
15	NEXT_PAGE_EN	使能下一页 1: 设置使用下一页 0: 未使用下一页
14	保留	必须保持复位值
13	REMOTE_FAULT_EN	使能远程故障检测 1: 检测到自动协商故障 0: 无远程故障
12	EXTENDED_NEXT_PAGE	扩展下一页 PHY 不支持时，此位始终保持为 0
11	ASYM_PAUSE	非对称式暂停功能 A6 技术能力 1: 有非对称式暂停能力 0: 无非对称式暂停能力
10	PAUSE	暂停功能 A5 技术能力 1: 有暂停能力 0: 无暂停能力
9	100BASE_T4	100BASE-T4 功能 PHY 不支持时，此位始终保持为 0
8	100BASE_TX_FULL_DPX	100BASE-X 全双工功能 1: 支持全双工

- 0: 不支持全双工  
 注意: 当自动协商被禁用时, 该位上的值将反映在 [PHY 控制寄存器\(PHY MII CTL\)](#) BIT6, BIT13 (FORCE\_SPEED) 和 BIT8(FORCE\_DUPLEX) 中的值, 当 FORCE\_SPEED 为 01 和 FORCE\_DUPLEX 为 1 时, 该位将为 1, 反之亦然。
- 7      100BASETX\_HALF\_DPX    100BASE-X 半双工功能  
 1: 支持半双工  
 0: 不支持半双工  
 注意: 当自动协商被禁用时, 该位上的值将反映在 [PHY 控制寄存器\(PHY MII CTL\)](#) BIT6, BIT13 (FORCE\_SPEED) 和 BIT8(FORCE\_DUPLEX) 中的值, 当 FORCE\_SPEED 为 01 和 FORCE\_DUPLEX 为 0 时, 该位将为 1, 反之亦然。
- 6      10BASE-T\_FULL\_DPX      10BASE-T 全双工功能  
 1: 支持全双工  
 0: 不支持全双工  
 注意: 当自动协商被禁用时, 该位上的值将反映在 [PHY 控制寄存器\(PHY MII CTL\)](#) BIT6, BIT13 (FORCE\_SPEED) 和 BIT8(FORCE\_DUPLEX) 中的值, 当 FORCE\_SPEED 为 00 和 FORCE\_DUPLEX 为 1 时, 该位将为 1, 反之亦然。
- 5      10BASE-T\_HALF\_DPX      10BASE-T 半双工功能  
 1: 支持半双工  
 0: 不支持半双工  
 注意: 当自动协商被禁用时, 该位上的值将反映在 [PHY 控制寄存器\(PHY MII CTL\)](#) BIT6, BIT13 (FORCE\_SPEED) 和 BIT8(FORCE\_DUPLEX) 中的值, 当 FORCE\_SPEED 为 00 和 FORCE\_DUPLEX 为 0 时, 该位将为 1, 反之亦然。
- 4:0      SLCT\_FLD[4:0]            识别消息类型  
 始终保持为 5'h01

## 自动协商链路伙伴(LP)能力寄存器 (PHY\_LP\_ABILITY)

地址偏移: 0x05

复位值: 0x0000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	ACKNOWLEDGE	REMOTE_FAULT	EXTENDED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASE_TX_FULL_DPX	100BASE_TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SELECTOR_FIELD[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r				

位/位域	名称	描述
15	NEXT_PAGE	链路伙伴下一页请求 1: 链路伙伴请求下一页功能 0: 基本页请求
14	ACKNOWLEDGE	链路伙伴确认收到 1: 链路伙伴确认收到成功

		0: 收到失败
13	REMOTE_FAULT	链路伙伴检测到远端故障 1: 检测到自适应故障 0: 无远端故障
12	EXTENDED_NEXT_PAGE	扩展下一页
11	ASYM_PAUSE	链路伙伴非对称式暂停功能 A6 技术能力 1: 有非对称式暂停能力 0: 无非对称式暂停能力
10	PAUSE	链路伙伴对称暂停功能 A5 技术能力 1: 有对称式暂停能力 0: 无对称式暂停能力
9	100BASET4	A4 技术能力 链路伙伴 100BASE-T4 功能
8	100BASETX_FULL_DPX	链路伙伴 100BASE-X 全双工功能 1: 支持全双工 0: 不支持全双工 注意: 当自适应关闭时, 该bit的值始终为1
7	100BASETX_HALF_DPX	链路伙伴 100BASE-X 半双工功能 1: 支持半双工 0: 不支持半双工 注意: 当自适应关闭时, 该bit的值始终为1
6	10BASE-T_FULL_DPX	链路伙伴 10BASE-T 全双工功能 1: 支持全双工 0: 不支持全双工 注意: 当自适应关闭时, 该bit的值始终为1
5	10BASE-T_HALF_DPX	链路伙伴 10BASE-T 半双工功能 1: 支持半双工 0: 不支持半双工 注意: 当自适应关闭时, 该bit的值始终为1
4:0	SELECTOR_FIELD[4:0]	链路伙伴识别消息类型 始终保持为 5'h01

### 自动协商扩展寄存器 (PHY\_AUTONEG\_EXP)

地址偏移: 0x06

复位值: 0x0064

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									NXT_PAG E_LOC_A BLE	NXT_PAG E_STO_L OC	PARA_DE T_FAULT	LNK_PTN ER_NP_A BLE	MR_NP_A BLE	PAGE_RE CEIVED	LNK_PTN ER_AN_A BLE
									r	r	r	r	r	r	r

位/位域	名称	描述
15:7	保留	必须保持复位值。
6	NXT_PAGE_LOC_ABLE	接收下一页位置功能 1: 收到下一页存储位置由bit 5确定 0: 收到下一页存储位置不由bit 5确定
5	NXT_PAGE_STO_LOC	收到的下一页存储位置 1: 链接伙伴下一页存储在 <a href="#">自动协商下一页接收寄存器 (PHY_AUTONEG_NEXT_PAGE_RECEIVE)</a> 0: 链接伙伴下一页存储在 <a href="#">自动协商链路伙伴(LP)能力寄存器 (PHY_LP_ABILITY)</a>
4	PARA_DET_FAULT	Parallel Detect Fault并行故障检测 1: 本地设备检测到并行故障 0: 无故障
3	LNK_PTNER_NP_ABLE	链接伙伴下一页功能 1: 链接伙伴是下一页的功能 0: 链接伙伴不是下一页的功能
2	MR_NP_ABLE	下一页功能 1: 当前设备是下一页功能 0: 当前设备不是下一页功能
1	PAGE_RECEIVED	收到新页面 1: 已收到新页面 0: 未收到新页面
0	LNK_PTNER_AN_ABLE	链接伙伴自适应功能 1: 链接伙伴是自适应功能 0: 链接伙伴不是自适应功能

## 自动协商下一页发送寄存器 (PHY\_AUTONEG\_NEXT\_PAGE\_TRANSMIT)

地址偏移: 0x07

复位值: 0x2001

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_P AGE	保留	MESSAG E_PAGE	ACKNOW LEDGE2	TOGGLE	MSG_UFMT_CODE_FIELD[10:0]										
r/w		r/w	r/w	r	r/w										

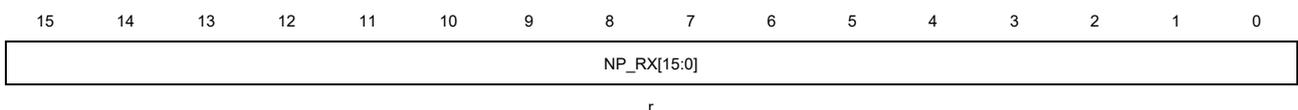
位/位域	名称	描述
15	NEXT_PAGE	最后一页 证明这是否是最后一页 1: 附加的下一页在这之后 0: 下一页是最后一页
14	保留	必须保持复位值
13	MESSAGE_PAGE	消息页或未格式化页 指示这是消息页或未格式化的页 1: 消息页 0: 未格式化页
12	ACKNOWLEDGE2	执行信息功能 1: 执行信息 0: 不执行信息
11	TOGGLE	切换 切换位由硬件自动计算，软件可忽略
10:0	MSG_UFMT_CODE_FIELD[10:0]	消息/未格式化代码字段

### 自动协商下一页接收寄存器 (PHY\_AUTONEG\_NEXT\_PAGE\_RECEIVE)

地址偏移: 0x08

复位值: 0x0000

该寄存器可以按半字 (16位)。



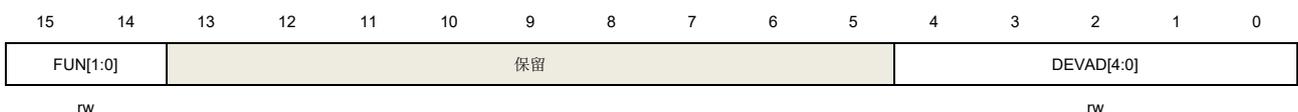
位/位域	名称	描述
15:0	NP_RX[15:0]	从链路伙伴接收到下一页

### MMD 访问控制寄存器 (MMD\_CTL)

地址偏移: 0x0D

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
------	----	----

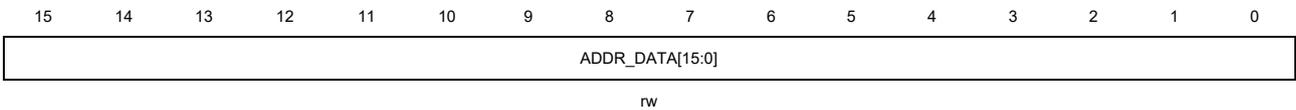
15:14	FUN[1:0]	功能 00: 地址 01: 数据, 没有后增量 10: 数据, 读和写的后增量 11: 数据, 只对写操作进行后增量
13:5	保留	必须保持复位值。
4:0	DEVAD[4:0]	设备地址

### MMD 访问数据地址寄存器 (MMD\_ADDR\_DATA)

地址偏移: 0x0E

复位值: 0x0000

该寄存器可以按半字 (16位)。



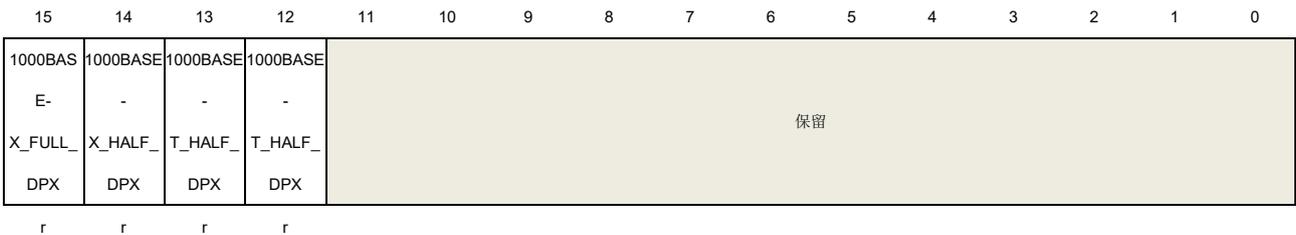
位/位域	名称	描述
15:0	ADDR_DATA[15:0]	数据地址 当bit 13.15:14==0时, 为地址寄存器, 否则为数据寄存器。

### PHY 扩展状态寄存器 (PHY\_EXTENDED\_STATUS)

地址偏移: 0x0F

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15	1000BASE-X_FULL_DPX	1000BASE TX全双工功能 1: PHY是1000BASE-X全双工 0: PHY不是1000BASE-X全双工
14	1000BASE-X_HALF_DPX	1000BASE TX半双工功能 1: PHY是1000BASE-X半双工 0: PHY不是1000BASE-X半双工
13	1000BASE-T_HALF_DPX	1000BASE-T全双工功能

		1: PHY是1000BASE-T全双工 0: PHY不是1000BASE-T全双工
12	1000BASE-T_HALF_DPX	1000BASE-T半双工功能 1: PHY是1000BASE-T半双工 0: PHY不是1000BASE-T半双工
11:0	保留	必须保持复位值。

### 中断状态寄存器 (INT\_STS)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位)。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LNK_STS	MGC_PKT	TX_LPI_R	RX_LPI_R	保留											
	_CHG_IN	_DET_INT	CV_INT	CV_INT												
	T															
	rc_w1	rc_w1	rc_w1	rc_w1												

位/位域	名称	描述
15	LNK_STS_CHG_INT	连接状态变化中断 0: 正常 1: 连接状态改变
14	MGC_PKT_DET_INT	魔术包检测中断 0: 正常 1: 检测到魔术包
13	TX_LPI_RCV_INT	TX LPI接收中断 0: 正常 1: TX LPI接收
12	RX_LPI_RCV_INT	RX LPI接收中断 0: 正常 1: RX LPI接收
11:0	保留	必须保持复位值。

### 中断掩码寄存器 (INT\_MASK)

地址偏移: 0x11

复位值: 0x0000

该寄存器可以按半字 (16位)。

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
--	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

LNK_STS	MGC_PKT	TX_LPI_R	RX_LPI_R	保留
_CHG_IN	_DET_INT	CV_INT_	CV_INT_M	
T_MSK	_MSK	MSK	SK	
rw	rw	rw	rw	

位/位域	名称	描述
15	LNK_STS_CHG_INT_MSK	连接状态变化中断掩码 当此位设置为1且INT_STS寄存器中的LINK_STS_CHG_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了连接状态变化中断。 1: 正常 0: 中断掩码
14	MGC_PKT_DET_INT_MSK	魔术包检测中断掩码 当此位设置为1且INT_STS寄存器中的MGC_PKT_DET_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了魔术包检测中断。 1: 正常 0: 中断掩码
13	TX_LPI_RCV_INT_MSK	TX LPI接收中断掩码 当此位设置为1且INT_STS寄存器中的TX_LPI_RCV_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了TX LPI接收中断。 1: 正常 0: 中断掩码
12	RX_LPI_RCV_INT_MSK	RX LPI接收中断掩码 当此位设置为1且int_sts寄存器中的RX_LPI_RCV_INT位被设置为1，INT_SMI_INT_N/EXT_SMI_INT_N接口被设置为0(低电平触发)，表示发生了RX LPI接收中断。 1: 正常 0: 中断掩码
11:0	保留	必须保持复位值。

### 环回控制寄存器 (PHY\_LB\_CTL)

地址偏移: 0x12

复位值: 0x0000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_LB_X	保留	RG_LB_P	RG_LB_P	RG_LB_A	RG_LB_E	保留			RG_LB_M	保留					
MII2MAC		CS2MAC	MA2MAC	FE2MAC	PG2EPC				MII2PHY						
rw	rw	rw	rw	rw	rw				rw						

位/位域	名称	描述
15	RG_LB_XMII2MAC	XMII2MAC环回使能
14	保留	必须保持复位值。
13	RG_LB_PCS2MAC	PCS2MAC环回使能
12	RG_LB_PMA2MAC	PMA2MAC环回使能
11	RG_LB_AFE2MAC	AFE2MAC环回使能
10	RG_LB_EPG2EPC	EPG2EPC环回使能
9:8	保留	必须保持复位值。
7	RG_LB_MMII2PHY	RMII2PHY 环回使能 仅在使用内部phy时有效。
6:0	保留	必须保持复位值。

### PHY 全局配置寄存器 (PHY\_GLOBAL\_CONFIG)

地址偏移: 0x13

复位值: 0x0102

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		XMII_TXC _INV	XMII_RXC _INV	EN_FX	EN_WOL	FORCE_LI NK	RG_WOL RCV_BC	RG_WOL CHK_PS WD	保留						MDI_MD[1:0]
		rw	rw	rw	rw	rw	rw	rw							rw

位/位域	名称	描述
15:14	保留	必须保持复位值。
13	XMII_TXC_INV	XMII TXCLK 反转 0: TXCLK 在 XMII 接口上没有反转 1: TXCLK在XMII接口上反转
12	XMII_RXC_INV	XMII RXCLK 反转 0: RXCLK 在 XMII 接口上没有反转 1: RXCLK在XMII接口上反转
11	EN_FX	光纤使能 该位将与 EPHY_FXEN 进行 OR 处理以确定 EPHY 的介质类型。下面显示了 OR 'ed 结果的定义 1: 光纤模式 0: 双绞线模式 注意: 当 EPHY 被编程为光纤模式时, Auto MDIX 应该被自动禁用, 并强制进入 MDI 模式

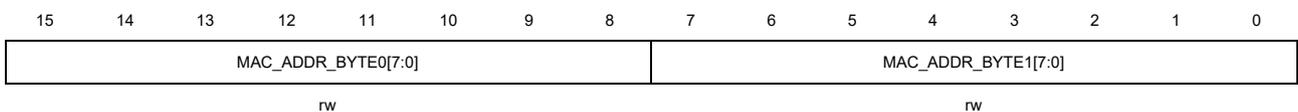
10	EN_WOL	使能 Wake-On-Lan Wake-On-Lan
9	FORCE_LINK	强制建立连接 1: 强制两个10/100M模块连接 0: 是否强制建立连接分别取决于10/100M配置寄存器中的FORCE_LINK_10和/或FORCE_LINK_100 注意: FORCE LINK仅在PHY编程为强制10或100M模式时有效。当PHY被编程为自动协商模式时, 程序1'b1到该位没有作用(如果在自动协商开启之前将该位编程为1'b1, 则H/W应自动禁用该位)
8	RG_WOL_RCV_BC	使能WOL模式接收广播魔术包 1: 使能接收广播魔术包 0: 其他
7	RG_WOL_CHK_PSWD	使能WOL模式下的SecureOn密码检查功能 1: 使能SecureOn密码检查 0: 其他
6:2	保留	必须保持复位值。
1:0	MDI_MD[1:0]	MDI/MDIX 模式 DUPCOLLED 和 RXER 将会被锁存 上电复位并存储在这两个 bit 位 00: 强制 MDI 模式 01: 模式 10: 自动 MDI/MDIX 检测(默认) 11: 保留 当 PHY 处于环回模式或光纤模式时,MDI/MDIX 模式将自动设置为“强制 MDI 模式”

### MAC 地址寄存器 0 (RG\_MAC\_AADR\_0)

地址偏移: 0x16

复位值: 0x0000

该寄存器可以按半字 (16位)。



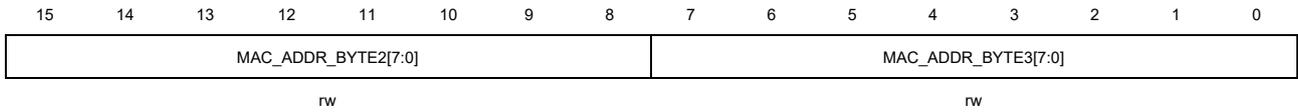
位/位域	名称	描述
15:8	MAC_ADDR_BYTE0[7:0]	传输顺序中的 MAC 地址 Byte0
7:0	MAC_ADDR_BYTE1[7:0]	传输顺序中的 MAC 地址 Byte1

### MAC 地址寄存器 1 (RG\_MAC\_AADR\_1)

地址偏移: 0x17

复位值：0x0000

该寄存器可以按半字（16位）。



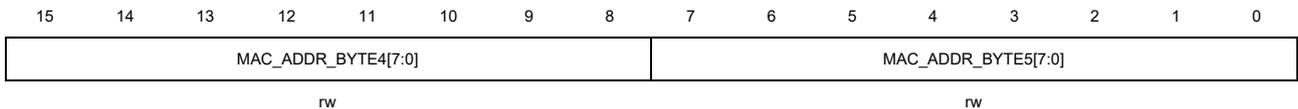
位/位域	名称	描述
15:8	MAC_ADDR_BYTE2[7:0]	传输顺序中的 MAC 地址 Byte2
7:0	MAC_ADDR_BYTE3[7:0]	传输顺序中的 MAC 地址 Byte3

### MAC 地址寄存器 2 (RG\_MAC\_AADR\_2)

地址偏移：0x18

复位值：0x0000

该寄存器可以按半字（16位）。



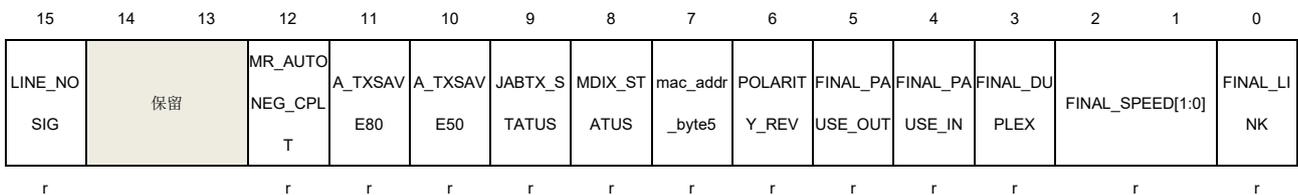
位/位域	名称	描述
15:8	MAC_ADDR_BYTE4[7:0]	传输顺序中的 MAC 地址 Byte4
7:0	MAC_ADDR_BYTE5[7:0]	传输顺序中的 MAC 地址 Byte5

### PHY 状态寄存器 (PHY\_STATUS)

地址偏移：0x19

复位值：0x0800

该寄存器可以按半字（16位）。



位/位域	名称	描述
15	LINE_NOSIG	无媒介信号 0: 媒介中检测到信号 1: 媒介中未检测到信号
14:13	保留	必须保持复位值
12	MR_AUTONEG_CPLT	完成自协商 1: 完成

		0: 未完成
11	A_TXSAVE80	10M 发送 80%振幅状态 1: 80%TX 振幅状态 0: 正常振幅
10	A_TXSAVE50	10M 发送 50%振幅状态 1: 50%TX 振幅状态 0: 正常振幅
9	JABRX_STATUS	实时 RX Jabber 状态 1: RX Jabber 0: 没有 RX Jabber
8	JABTX_STATUS	实时 TX Jabber 状态 1: TX Jabber 0: 没有 RX Jabber
7	MDIX_STATUS	MDIX 状态 1: MDIX 0: MDI
6	POLARITY_REV	极性状态 1: 保留 0: 正常
5	FINAL_PAUSE_OUT	暂停输出功能 启用自动协商后, 该位由 <a href="#">自动协商广播寄存器 (PHY AUTONEG ADV)</a> BIT [11:10] 和 <a href="#">自动协商链路伙伴(LP)能力寄存器 (PHY LP ABILITY)</a> BIT[11:10]在连接后确定。 当自动协商被禁用时, 这个位始终被设置为 0。 1: 存在暂停输出功能 0: 不存在暂停输出功能
4	FINAL_PAUSE_IN	暂停输入功能 启用自动协商后, 该位由 <a href="#">自动协商广播寄存器 (PHY AUTONEG ADV)</a> BIT [11:10] 和 <a href="#">自动协商链路伙伴(LP)能力寄存器 (PHY LP ABILITY)</a> BIT[11:10]在连接后确定。 当自动协商被禁用时, 这个位始终被设置为 0。 1: 存在暂停输入功能 0: 不存在暂停输入功能
3	FINAL_DUPLEX	双工状态 在连接之前, S/W 应该忽略这个状态此时无意义 1: 全双工 0: 半双工
2:1	FINAL_SPEED[1:0]	速度状态 在连接之前, S/W 应该忽略这个状态此时无意义 1: 100M 0: 10M

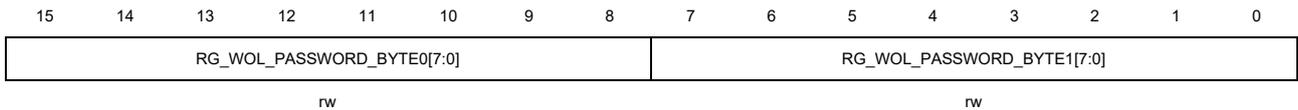
0	FINAL_LINK	连接状态 1: 建立连接 0: 断开连接
---	------------	----------------------------

### Wake-On-Lan 安全密码寄存器 0 (RG\_WOL\_PASSWORD\_0)

地址偏移: 0x1A

复位值: 0x0000

该寄存器可以按半字 (16位)。



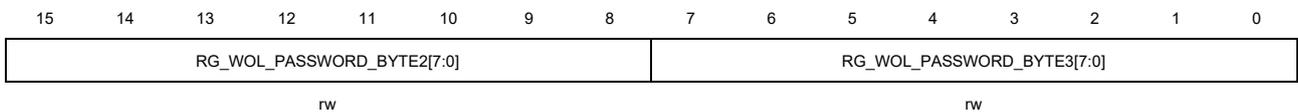
位/位域	名称	描述
15:8	RG_WOL_PASSWORD_BYTE0[7:0]	SecureON 传输顺序中的密码 Byte0
7:0	RG_WOL_PASSWORD_BYTE1[7:0]	SecureON 传输顺序中的密码 Byte1

### Wake-On-Lan 安全密码寄存器 1 (RG\_WOL\_PASSWORD\_1)

地址偏移: 0x1B

复位值: 0x0000

该寄存器可以按半字 (16位)。



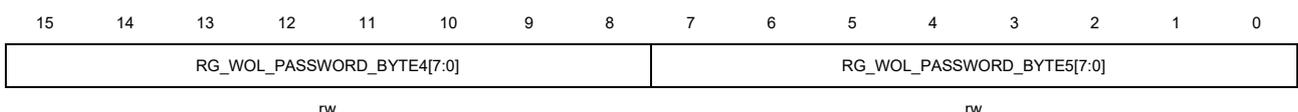
位/位域	名称	描述
15:8	RG_WOL_PASSWORD_BYTE2[7:0]	SecureON 传输顺序中的密码 Byte2
7:0	RG_WOL_PASSWORD_BYTE3[7:0]	SecureON 传输顺序中的密码 Byte3

### Wake-On-Lan 安全密码寄存器 2 (RG\_WOL\_PASSWORD\_2)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:8	RG_WOL_PASSWORD_BYTE4[7:0]	SecureON 传输顺序中的密码 Byte4

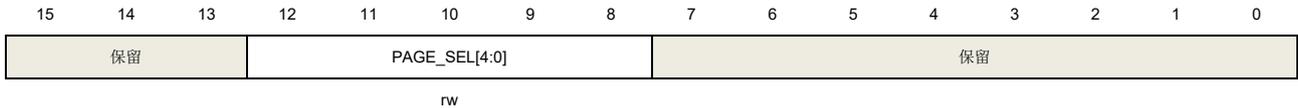
7:0 RG\_WOL\_PASSWORD\_BYTE5[7:0] SecureON 传输顺序中的密码 Byte5

### Page 选择寄存器 (PHY\_PAGE\_SEL)

地址偏移: 0x1F

复位值: 0x003D

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:13	保留	必须保持复位值。
12:8	PAGE_SEL[4:0]	页选择寄存器
7:0	保留	必须保持复位值。

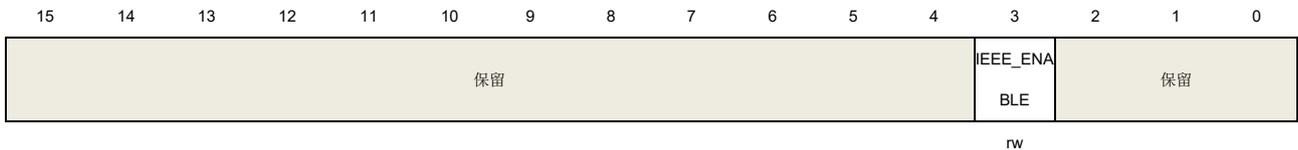
## 8.4.2. Page1 寄存器

### EEE 配置寄存器 (EEE\_CFG)

地址偏移: 0x17

复位值: 0x0033

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:4	保留	必须保持复位值。
3	IEEE_ENABLE	使能 EEE 使能 EPHY TX 在无数据传输时自动进入 LPI 状态
2:0	保留	必须保持复位值。

## 8.4.3. Page2 寄存器

### 10M 节能控制寄存器 (PHY\_10M\_PWRSERVE)

地址偏移: 0x17

复位值: 0x04C8

该寄存器可以按半字（16位）。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYPASS_TXSAVE 80	BYPASS_TXSAVE5 0	TEST_TX_SAVE80	TEST_TX_SAVE50	LINKPULSE_DLY_TH[3:0]				SAVE_80_PERCENT	TX10SAVE_E_MODE	保留	SAVE_ON_DLY_TH[4:0]				
rw	rw	rw	rw	rw				rw	rw		rw				

位/位域	名称	描述
15	BYPASS_TXSAVE80	分流器正常省电 80%通道
14	BYPASS_TXSAVE50	分流器正常省电 50%通道
13	TEST_TX_SAVE80	省电 80%测试输入 仅 BYPASS_TXSAVE80 使能时有效
12	TEST_TX_SAVE50	省电 50%测试输入 仅 BYPASS_TXSAVE50 使能时有效
11:8	LINKPULSE_DLY_TH[3:0]	链路脉冲延时产生阈值 TX 链路脉冲延迟传输的阈值。 延迟的时间是编程的值 x 2 个周期，其中 1 个周期等于 40ns。 这里设置的默认值是延迟 320ns
7	SAVE_80_PERCENT	省电 80% 1: 使能 0: 未使能
6	TX10SAVE_MODE	10M Power Saving Mode10M 省电模式 1: 10M 省电模式使能 0: 正常模式没有省电
5	保留	必须保持复位值。
4:0	SAVE_ON_DLY_TH[4:0]	省电模式关于延时阈值

## 模拟传输数据测试和控制寄存器（PHY\_TXDATA\_CTRL）

地址偏移：0x18

复位值：0x1000

该寄存器可以按半字（16位）。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	FIR_10_SEL[2:0]			保留	BYPASS_TX_10_DATA_TA	BYPASS_TX_10_DATA_TA	TEST_TX_100_DATA [1:0]		保留	TEST_TX_10_DATA [4:0]					
	rw				rw	rw	rw			rw					

位/位域	名称	描述
------	----	----

15	保留	必须保持复位值。
14:12	FIR_10_SEL[2:0]	10M TX 滤波选择
11:10	保留	必须保持复位值。
9	BYPASS_TX_100_DATA	分流器 100M 数据传输 1: 将 100M 分流器数据设为模拟并强制 TEST_TX_100_DATA 为模拟 0: 正常 100M 数据发送到模拟块
8	BYPASS_TX_10_DATA	分流器 10M 数据传输 1: 将 10M 分流器数据设为模拟并强制 TEST_TX_10_DATA 为模拟 0: 正常 10M 数据发送到模拟块
7:6	TEST_TX_100_DATA[1:0]	100M 测试数据 当 BYPASS_TX_100_DATA 设置为 1 时, 反馈到模拟块
5	保留	必须保持复位值。
4:0	TEST_TX_10_DATA[4:0]	10M 测试数据 当 BYPASS_TX_10_DATA 设置为 1 时, 反馈到模拟块

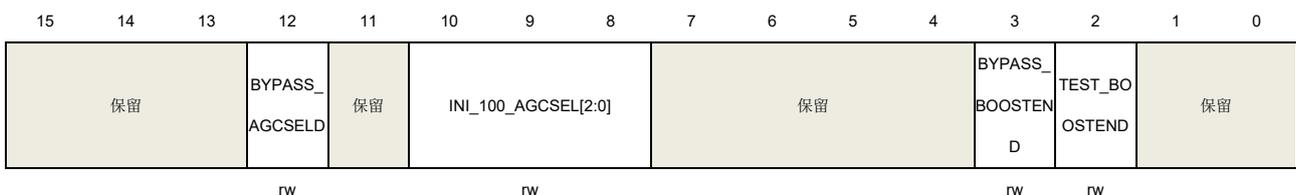
#### 8.4.4. Page3 寄存器

##### DSPSM 控制寄存器 (PHY\_DSPSM\_CTRL)

地址偏移: 0x11

复位值: 0x8510

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:13	保留	必须保持复位值。
12	BYPASS_AGCSELD	AGCSEL 分流器 1: 分流器正常 agcseID 0: 正常
11	保留	必须保持复位值。
10:8	INI_100_AGCSEL[2:0]	100M 模式 AGCSEL 初始值 这 3 位的功能根据 BYPASS_AGCSELD BYPASS 的设置不同 AGCSELD=1: 内部 AGCSEL 固定为 INI_100_AGCSEL BYPASS AGCSELD=0:在 DSPRST 和内部 boost 为 0 时加载 INI_100_AGCSEL, 当内部 boost 为低时保持不变。

因此，DSPSM 可以覆盖 [100,000] 和 [1,000] 到 [1,111] 范围内的 [boosten, AGCSEL]。

7:4	保留	必须保持复位值。
3	BYPASS_BOOSTEND	BoostenD 分流器模式 内部 BoostenD 将被跳过和 TEST_BOOSTEND 将用于调试目的的控制 1: 分流器模式 0: 正常模式
2	TEST_BOOSTEND	BoostenD 测试输入 当 BYPASS_BOOSTEND=1 时，强制 BoostenD 为该值
1:0	保留	必须保持复位值。

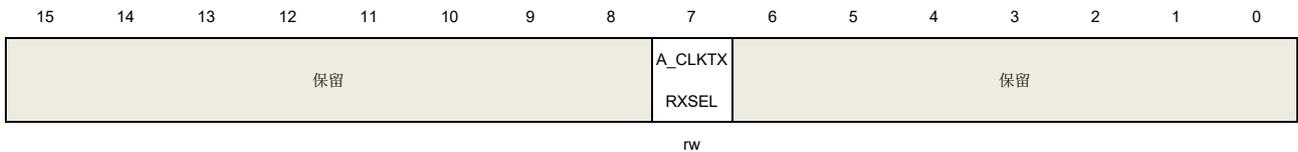
## 8.4.5. Page6 寄存器

### 模拟 ADC 控制寄存器 (PHY\_ADC\_CTL)

地址偏移: 0x10

复位值: 0x5563

该寄存器可以按半字 (16位)。



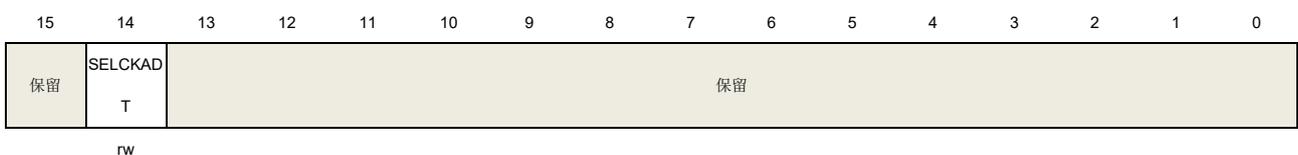
位/位域	名称	描述
15:8	保留	必须保持复位值。
7	A_CLKTXRXSEL	RXCLK125 到 TXCLK125 分流器 1: 内部 RXCLK125 选择 TXCLK125 0: 内部 RXCLK125 选择 RXCLK125(默认)
6:0	保留	必须保持复位值。

### 模拟 Pre-Gain 和 PLL 配置寄存器 (PHY\_PGPLL\_CTL)

地址偏移: 0x12

复位值: 0x0D00

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15	保留	必须保持复位值。
14	SELCKADT	测试模式 ADC 时钟选择 该位仅在 PHY 处于 AFE 测试模式时可用，非 AFE 测试模式时，向该位写入值无效 1: 选择 CKADTEST 作为 ADC 输入时钟 0: 选择 RXCLK125 作为 ADC 输入时钟
13:0	保留	必须保持复位值。

#### 8.4.6. Page9 寄存器

##### 内置包生成器和校验命令寄存器 (EPGC\_CMD)

地址偏移: 0x10

复位值: 0x0000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_EPG_EN	RG_EPG_DATA_TYPE[2:0]			RG_EPG_IPG_LEN_FIXED	RG_EPG_DASA_FIXED	RG_EPG_MODE[1:0]		RG_EPG_LPI_INDICATION	RG_EPG_LOC	保留	RG_EPG_CLR_CNT	RG_EPG_CLR_CNT	RG_EPG_PAUSE	RG_EPG_GO	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	

位/位域	名称	描述
15	RG_EPG_EN	使能内置包发生器 此位用于使能内置包生成器以进行调试。当它被使用时，内部tx数据路径将切换到内置包生成器
14:12	RG_EPG_DATA_TYPE[2:0]	内置包发生器模式 3'b000: 全0; 3'b001: 全1; 3'b010: 全5s; 3'b011: 全As; 3'b100: 字节增量; 3'b101: 随机; 3'b110: 字节减量;
11	RG_EPG_IPG_LEN_FIXED	固定包间间隙长度 1: 报文间隙长度由RG_EPG_IPG_LEN确定 0: 包间间隙长度随机由硬件
10	RG_EPG_DASA_FIXED	固定DA/SA 1: DA固定为00-01-02-03-04-05和SA固定为0a-0b-0c-0d-0e-0f 0: DA/SA 由 RG EPG DATA TYPE 配置
9:8	RG_EPG_MODE[1:0]	内置包发生器模式

		2'b00: 单发模式;
		2'b01: 突发模式;
		2'b1x: 持续模式;
7	RG_EPG_LPI_INDICATION	TX LPI指示 1: 传输LPI 0: 正常
6	RG_EPC_LOC	EPC位置 1: EPC位于TX路径 0: EPC 位于 RX 路径
5:4	保留	必须保持复位值。
3	RG_EPC_CLR_CNT	内置包检查计数器清除 高位有效清除 rx 包检查统计计数器，包括总包数计数器和 crc 错误计数器，它是自清除。
2	RG_EPG_CLR_CNT	内置包生成计数器 清除高位有效清除tx包生成统计计数器，包括总包生成计数器，它是自清除。
1	RG_EPG_PAUSE	内置包生成器结合RG_EPG_GO控制包生成，{ RG_EPG_PAUSE, RG_EPG_GO } = 2'b01: 开始 2'b11: 暂停 2'b00: 结束 注意：在单发模式下(RG_EPG_MODE[1:0]= 2'b0)，只有start命令有效。在突发模式下(RG_EPG_MODE[1:0]= 2'b01)，这三个命令都是有效的，pause命令将暂停包的生成，内部突发包数计数器将保持，随后的start命令将继续当前突发的生成。在继续模式下(RG_EPG_MODE[1:0]= 2'b10)，所有三个命令都是有效的，并且暂停命令的行为将与停止命令相同。
0	RG_EPG_GO	内置包生成器结合RG_EPG_PAUSE控制包生成，请参考RG_EPG_PAUSE获取控制命令定义。 注意：在单发模式和连续模式下，当包生成任务完成时，该位将自动清除。当处于连续模式时，只有将0写入该位才能清除它。

### 内置包生成器包长度寄存器 (EPG\_PKT\_LEN)

地址偏移: 0x11

复位值: 0x0000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				RG_EPG_PKT_LEN_FIXED	RG_EPG_PKT_LEN[10:0]										

rw

rw

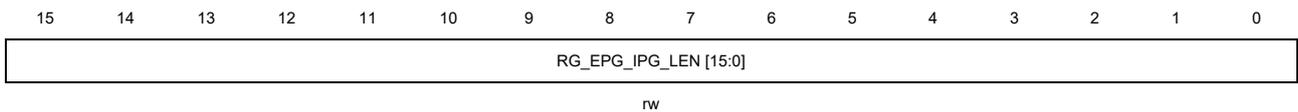
位/位域	名称	描述
15:12	保留	必须保持复位值。
11	RG_EPG_PKT_LEN_FIXED	固定报文长度使能 1: 数据包长度固定, 由 rg_epg_pkt_len 决定 0: 包长度随机由硬件决定
10:0	RG_EPG_PKT_LEN[10:0]	报文长度 包的总长度, 包括 DA/SA, 数据和 FCS。只有当 rg_epg_pkt_len_fixed 为 1(按字节计数)时才有效, 否则, 数据包长度将随机由硬件决定

### 内置包生成器包间间隔寄存器 (EPG\_IPG\_CFG)

地址偏移: 0x12

复位值: 0x0000

该寄存器可以按半字 (16位)。



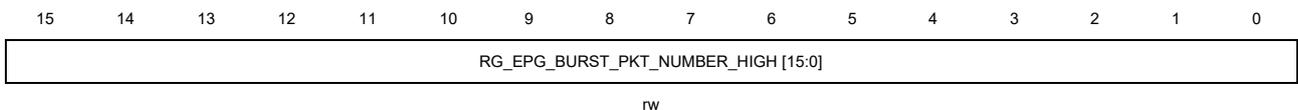
位/位域	名称	描述
15:0	RG_EPG_IPG_LEN [15:0]	Inter-Packet-Gap 长度

### 内置包生成器突发数量高位数据寄存器(EPG\_BURST\_PKT\_NUM\_CFG\_HIGH)

地址偏移: 0x13

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:0	RG_EPG_BURST_PKT_NUMBER_HIGH [15:0]	突发高位数据只在突发模式下有效 (RG_EPG_MODE==10)

### 内置包生成器突发数量低位数据寄存器(EPG\_BURST\_PKT\_NUM\_CFG\_LOW)

地址偏移: 0x14

复位值: 0x0000

该寄存器可以按半字 (16位)。



RG\_EPG\_BURST\_PKT\_NUMBER\_LOW [15:0]

rw

位/位域	名称	描述
15:0	RG_EPG_BURST_PKT_NUMBER_LOW [15:0]	突发低位数据只在突发模式下有效 (RG_EPG_MODE==10)

### TX 字节计数器高位数据寄存器 (TX\_BYTE\_CNT\_HIGH)

地址偏移: 0x15

复位值: 0x0000

该寄存器可以按半字 (16位)。

15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0

TX\_BYTE\_CNT\_HIGH[15:0]

r

位/位域	名称	描述
15:0	TX_BYTE_CNT_HIGH[15:0]	内置包发生器包生成字节计数器高位数据(bit31:16) 该计数器将通过 RG_EPG_CLR_CNT 来清除

### TX 字节计数器低位数据寄存器 (TX\_BYTE\_CNT\_LOW)

地址偏移: 0x16

复位值: 0x0000

该寄存器可以按半字 (16位)。

15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0

TX\_BYTE\_CNT\_LOW[15:0]

r

位/位域	名称	描述
15:0	TX_BYTE_CNT_LOW[15:0]	内置包发生器包生成字节计数器低位数据(bit[15:0]) 该计数器将通过 RG_EPG_CLR_CNT 来清除

### TX 总数据包计数器高位寄存器 (TX\_PKT\_CNT\_HIGH)

地址偏移: 0x17

复位值: 0x0000

该寄存器可以按半字 (16位)。

15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0

TX\_PKT\_CNT\_HIGH[15:0]

r

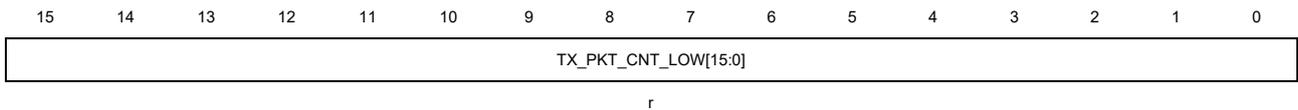
位/位域	名称	描述
15:0	TX_PKT_CNT_HIGH[15:0]	内置包生成器总包生成计数器高位数据(bit31:16) 该计数器将通过 RG_EPG_CLR_CNT 来清除

### TX 总数据包计数器低位寄存器 (TX\_PKT\_CNT\_LOW)

地址偏移: 0x18

复位值: 0x0000

该寄存器可以按半字 (16位)。



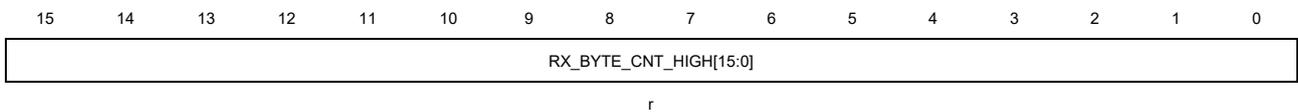
位/位域	名称	描述
15:0	TX_PKT_CNT_LOW[15:0]	内置包生成器总包生成计数器低位数据(bit15:0) 该计数器将通过 RG_EPG_CLR_CNT 来清除

### RX 字节计数器高位数据寄存器 (RX\_BYTE\_CNT\_HIGH)

地址偏移: 0x19

复位值: 0x0000

该寄存器可以按半字 (16位)。



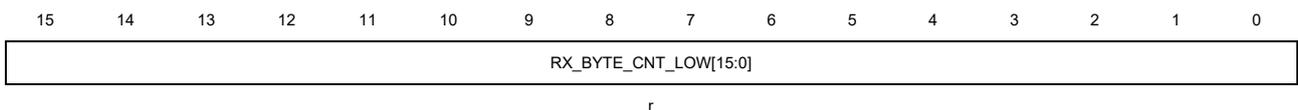
位/位域	名称	描述
15:0	RX_BYTE_CNT_HIGH[15:0]	数据包接收字节计数器高位数据(bit31:16) 该计数器将通过 RG_EPC_CLR_CNT 来清除

### RX 字节包计数器低位数据寄存器 (RX\_BYTE\_CNT\_LOW)

地址偏移: 0x1A

复位值: 0x0000

该寄存器可以按半字 (16位)。



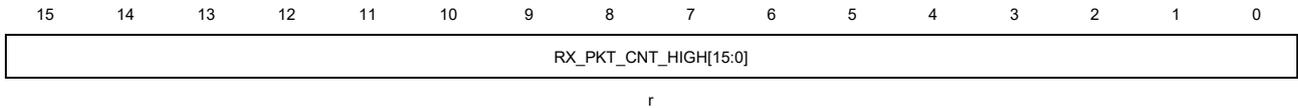
位/位域	名称	描述
15:0	RX_BYTE_CNT_LOW[15:0]	数据包接收字节计数器低位数据(bit15:0) 该计数器将通过 RG_EPC_CLR_CNT 来清除

### RX 总数据包计数器高位寄存器 (RX\_PKT\_CNT\_HIGH)

地址偏移: 0x1B

复位值: 0x0000

该寄存器可以按半字 (16位)。



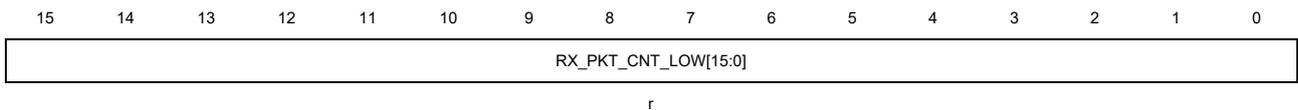
位/位域	名称	描述
15:0	RX_PKT_CNT_HIGH[15:0]	总包接收计数器高位数据(bit31:16) 该计数器将通过 RG_EPC_CLR_CNT 来清除

### RX 总数据包计数器低位寄存器 (RX\_PKT\_CNT\_LOW)

地址偏移: 0x1C

复位值: 0x0000

该寄存器可以按半字 (16位)。



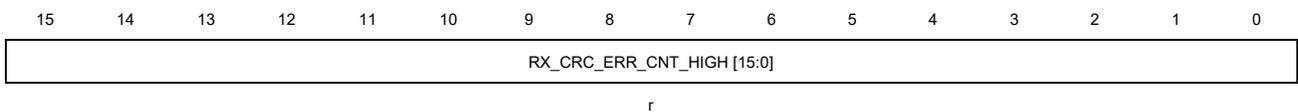
位/位域	名称	描述
15:0	RX_PKT_CNT_LOW[15:0]	总包接收计数器低位数据(bit15:0) 该计数器将通过 RG_EPC_CLR_CNT 来清除

### RX CRC 错误数据包计数器高位数据寄存器 (RX\_CRC\_ERR\_CNT\_HIGH)

地址偏移: 0x1D

复位值: 0x0000

该寄存器可以按半字 (16位)。



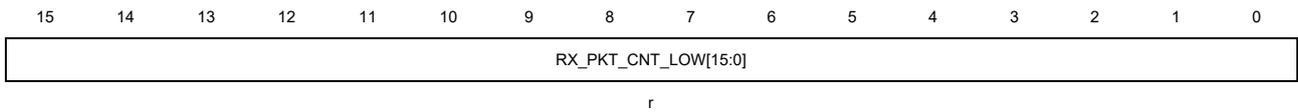
位/位域	名称	描述
15:0	RX_CRC_ERR_CNT_HIGH[15:0]	错误包接收计数器高位数据(bit31:16) 该计数器将通过 RG_EPC_CLR_CNT 来清除

### RX CRC 错误数据包计数器低数据寄存器 (RX\_CRC\_ERR\_CNT\_LOW)

地址偏移: 0x1E

复位值: 0x0000

该寄存器可以按半字（16位）。



位/位域	名称	描述
15:0	RX_CRC_ERR_CNT_LOW[15:0]	错误包接收计数器低位数据(bit15:0) 该计数器将通过 RG_EPC_CLR_CNT 来清除

### 8.4.7. MDIO 寄存器

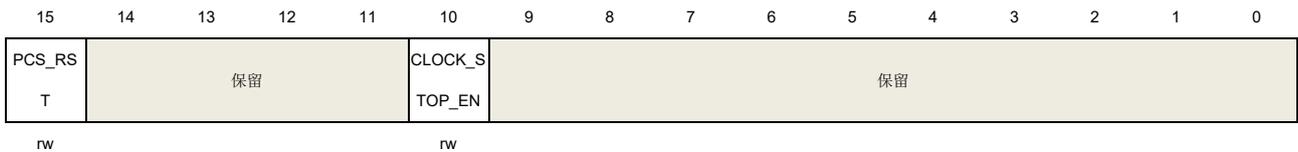
#### PCS 控制寄存器 1 (PCS\_CTL\_1)

设备地址：0x3

地址偏移：0x00

复位值：0x0400

该寄存器可以按半字（16位）。



位/位域	名称	描述
15	PCS_RST	PCS 复位 将 AN 和 PCS MMD 寄存器复位为默认值，并在 PHY 中引起软件复位，复位完成后将自动清除
14:11	保留	必须保持复位值
10	CLOCK_STOP_EN	使能接收时钟停止 设置为 1 表示在发送 LPI 信号时停止接收 xMII 时钟，否则它将保持时钟活动
9:0	保留	必须保持复位值

#### PCS 状态寄存器 1 (PCS\_STS\_1)

设备地址：0x3

地址偏移：0x01

复位值：0x0040

该寄存器可以按半字（16位）。



位/位域	名称	描述
15:12	保留	必须保持复位值。
11	TX_LPI_REC	TX PCS 收到 LPI 它是用一个锁存高功能实现的，这样 TX LPI 信号使这个位变成 1 并保持 1 直到它被读取。 1: TX PCS 收到 LPI 0: LPI 未接收
10	RX_LPI_REC	RX PCS 收到 LPI 它是用一个锁存高功能实现的，这样 RX LPI 信号使这个位变成 1 并保持 1 直到它被读取。 1: RX PCS 收到 LPI 0: LPI 未接收
9	TX_LPI_INT	TX PCS 接收 LPI 指示 1: TX PCS 目前正在接收 LPI 0: PCS 目前没有收到 LPI
8	RX_LPI_INT	RX PCS 接收 LPI 指示 1: RX PCS 目前正在接收 LPI 0: PCS 目前没有收到 LPI
7	保留	必须保持复位值。
6	CLOCK_STOP_CAP	传输 xMII 时钟停止能力 1: RS 可能会在 LPI 期间停止发送 xMII 时钟 0: 传输 xMII 时钟不可停止
5:0	保留	必须保持复位值。

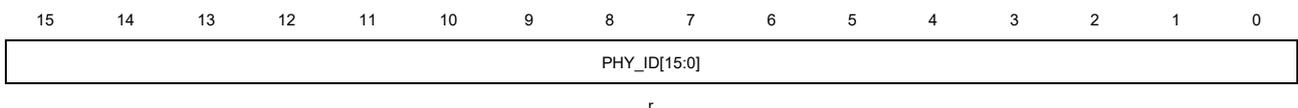
### PCS 设备标识符寄存器 (PCS\_ID)

设备地址: 0x3

地址偏移: 0x02

复位值: 0x0044

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:0	PHY_ID[15:0]	PHY 标识 bit[31-16] OUI (bits 3-18) .OUI =00-11-05

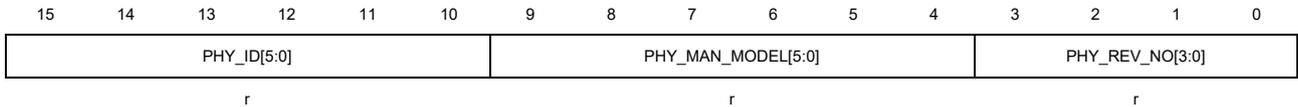
### PCS 设备版本寄存器 (PCS\_VER)

设备地址: 0x3

地址偏移: 0x03

复位值: 0x1400

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识 bit[15-10] OUI bits 19-24
9:4	PHY_MAN_MODEL[5:0]	制造商型号 制造商的型号(bits 5-0), 其中[5:4]=架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0) <a href="#">PCS 设备标识符寄存器 (PCS_ID)</a> bit 0 是 PHY 标识符的 LS 位

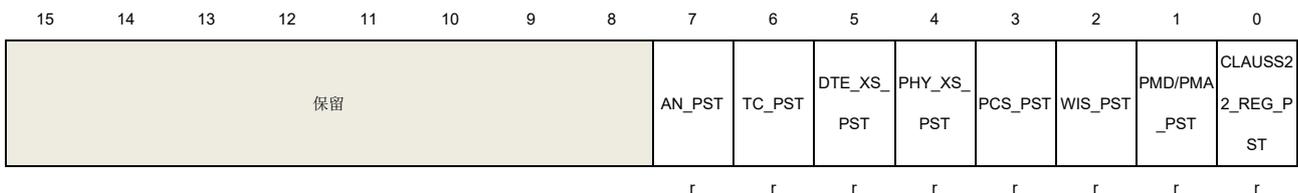
### PCS 数据包寄存器 0 (PCS\_PKG\_0)

设备地址: 0x3

地址偏移: 0x05

复位值: 0x0089

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:8	保留	必须保持复位值。
7	AN_PST	自动协商存在于包中 始终为 1
6	TC_PST	TC 存在于包中 始终为 0
5	DTE_XS_PST	DTE XS 存在于包中 始终为 0
4	PHY_XS_PST	PHY XS 存在于包中 始终为 0

3	PCS_PST	PCS 存在包中 始终为 0
2	WIS_PST	WIS 存在包中 始终为 0
1	PMD/PMA_PST	PMD/PMA 存在包中 始终为 0
0	CLAUSS22_REG_PST	自动协商存在于包中 始终为 1

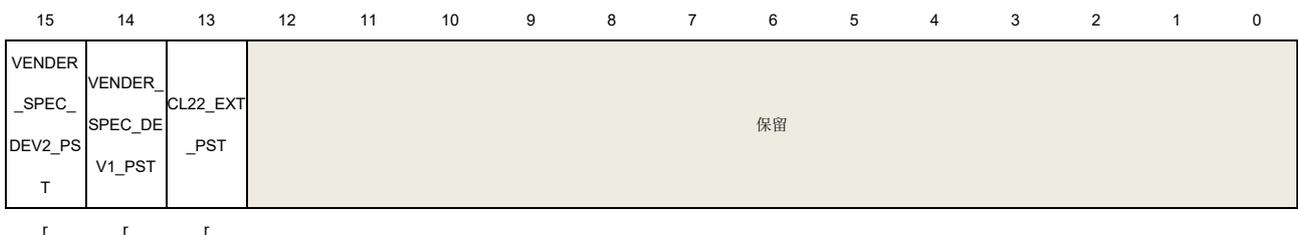
### PCS 数据包寄存器 1 (PCS\_PKG\_1)

设备地址: 0x3

地址偏移: 0x06

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15	VENDER_SPEC_DEV2_PST	特殊设备 2 存在于包中 始终为 0
14	VENDER_SPEC_DEV1_PST	特殊设备 1 存在于包中 始终为 0
13	CL22_EXT_PST	Clause 22 扩展在于包中 始终为 0
12:0	保留	必须保持复位值

### EEE 能力寄存器 (EEE\_CAP)

设备地址: 0x3

地址偏移: 0x14

复位值: 0x0003

该寄存器可以按半字 (16位)。



保留	10GBASE_KR_EEE	10GBASE_KX4_EEE E	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	保留
	r	r	r	r	r	rw	

位/位域	名称	描述
15:7	保留	必须保持复位值。
6	10GBASE_KR_EEE	10GBASE-KR EEE 不支持，始终为 0
5	10GBASE_KX4_EEE	10GBASE-KR EEE 不支持，始终为 0
4	1000BASE_KX_EEE	10GBASE-KR EEE 不支持，始终为 0
3	10GBASE_T_EEE	10GBASE-KR EEE 不支持，始终为 0
2	1000BASE_T_EEE	1000Base-T EEE 1: EEE支持1000BASE-T 0: EEE不支持1000Base-T
1	100BASE_TX_EEE	100Base-TX EEE 1: EEE支持100BASE-TX 0: EEE不支持100BASE-TX
0	保留	必须保持复位值。

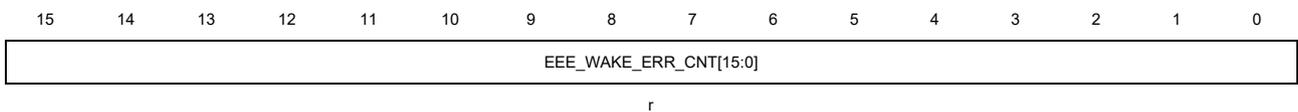
### EEE 唤醒错误计数器寄存器 (EEE\_WAKE\_ERR\_CNT)

设备地址: 0x3

地址偏移: 0x16

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:0	EEE_WAKE_ERR_CNT[15:0]	EEE唤醒错误计数器 用于统计PHY未在指定类型的时间内完成正常的序列时的时间故障。它被读取时被清除

### AN 控制寄存器 (AN\_CTL)

设备地址: 0x7

地址偏移: 0x00

复位值: 0x2000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AN_RST	保留	XNP_CTL	AN_EN	保留	保留	RESTART _AUTONE G	保留								
rw		r	rw			rw									

位/位域	名称	描述
15	AN_RST	AN 复位 将 AN 和 PCS MMD 寄存器复位为默认值，并在 PHY 中引起软件复位，复位完成后将自动清除
14	保留	必须保持复位值。
13	XNP_CTL	扩展下一页控制 0 不支持，保留 0
12	AN_EN	使能自动协商 它是 bit12 在 <a href="#">PHY 控制寄存器 (PHY MII CTL)</a> 1: 使能自动协商 0: 未使能自动协商
11:10	保留	必须保持复位值。
9	RESTART_AUTONEG	重启自动协商 它是 bit9 在 <a href="#">PHY 控制寄存器 (PHY MII CTL)</a> ，该位为自清除 1: 重启自动协商 0: 正常运行
8:0	保留	必须保持复位值。

### AN 状态寄存器 (AN\_STS)

设备地址: 0x7

地址偏移: 0x01

复位值: 0x0008

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留										PAGE_RE CEIVED	AN_COM PLETE	AN_ABILI TY	AN_ABILI TY	LINK_STA TUS	保留	LP_AN_A BILITY
										r	r	r	r	r		r

位/位域	名称	描述
15:7	保留	必须保持复位值。
6	PAGE_RECEIVED	页接收 设置为1表示已经接收到一个新的链接码字并将其存储在 <code>an_lp_xnp_ability</code> 寄存器中。在读取AN状态寄存器或自动协商扩展寄存器时，页面接收位将复位为0。它是 <a href="#">自动协商扩展寄存器 (PHY AUTONEG EXP)</a> 中的bit1。 1: 已收到一个新的链接码字 0: 没有收到链接码字
5	AN_COMPLETE	完成自动协商 设置为1表示自动协商已经完成，清除 <a href="#">AN控制寄存器 (AN CTL)</a> 中的第12位使能自动协商时返回0。
4	AN_ABILITY	远程故障 设置为1，表示检测到远程故障情况。当启用光纤模式时，远端故障定义为远端故障。它实现了一个锁存高功能，这样一个远程故障的发生导致位4成为设置并保持设置，直到它被清除。在读取AN状态寄存器或状态寄存器时，它将被复位为0。它是 <a href="#">PHY状态寄存器 (PHY MII STATUS)</a> 中的bit4。
3	AN_ABILITY	自动协商能力 在双绞线模式下总是设置为1，表示PHY具有自动协商的能力。它是 <a href="#">PHY状态寄存器 (PHY MII STATUS)</a> 中的bit3。
2	LINK_STATUS	连接状态 设置为1，表示建立了有效的链路。它将通过一个锁存低功能来实现，这样链路失败条件的发生会导致链路状态位被清除并保持清除，直到它被读取。它是 <a href="#">PHY状态寄存器 (PHY MII STATUS)</a> 中的bit2。
1	保留	必须保持复位值。
0	LP_AN_ABILITY	链路伙伴自动协商能力 设置为1，表示链路伙伴能够参与自动协商功能。

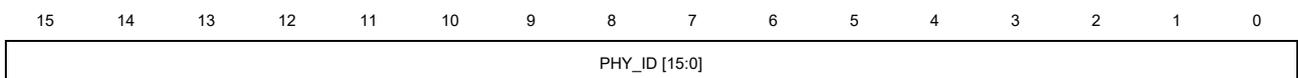
### 自动协商设备标识寄存器(AN\_ID)

设备地址: 0x7

地址偏移: 0x02

复位值: 0x0044

该寄存器可以按半字（16位）。



r

位/位域	名称	描述
15:0	PHY_ID[15:0]	PHY标识符bit[31-16] OUI (bits 3-18). OUI =00-11-05

### 自动协商设备版本寄存器 (AN\_VER)

设备地址: 0x7

地址偏移: 0x03

复位值: 0x1400

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID[5:0]						PHY_MAN_MODE[5:0]						PHY_REV_NO[3:0]			
r						r						r			

位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识符 bit [15-10] OUI bits 19-24
9:4	PHY_MAN_MODE[5:0]	制造商型号 制造商的型号(bits 5-0), 其中[5:4]=架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0) <a href="#">PCS 设备标识符寄存器 (PCS ID)</a> bit 0 是 PHY 标识符的 LS 位

### AN 数据包寄存器 0 (AN\_PKG\_0)

设备地址: 0x7

地址偏移: 0x05

复位值: 0x0089

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								AN_PST	TC_PST	DTE_XS_ PST	PHY_XS_ PST	PCS_PST	WIS_PST	PMD/PMA _PST	CLAUSS2 2_REG_P ST
								r	r	r	r	r	r	r	r

位/位域	名称	描述
15:8	保留	必须保持复位值。
7	AN_PST	自动协商存在于包中 始终为 1
6	TC_PST	TC 存在于包中 始终为 0

5	DTE_XS_PST	DTE XS 存在包中 始终为 0
4	PHY_XS_PST	PHY XS 存在包中 始终为 0
3	PCS_PST	PCS 存在包中 始终为 0
2	WIS_PST	WIS 存在包中 始终为 0
1	PMD/PMA_PST	PMD/PMA 存在包中 始终为 0
0	CLAUSS22_REG_PST	自动协商存在于包中 始终为 1

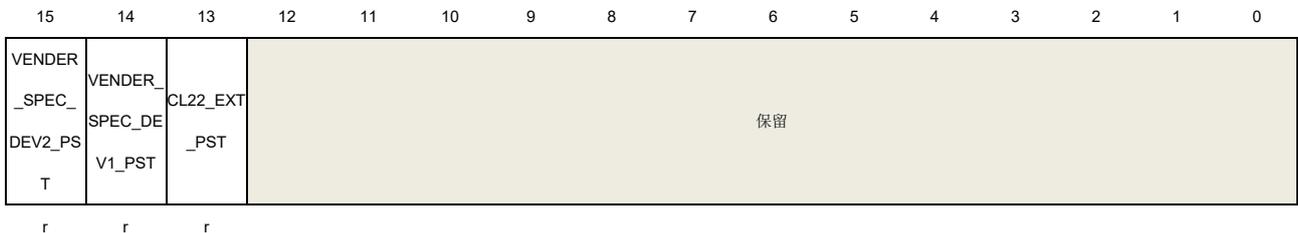
### AN 数据包寄存器 1 (AN\_PKG\_1)

设备地址: 0x7

地址偏移: 0x06

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15	VENDER_SPEC_DEV2_PST	特殊设备 2 存在于包中 始终为 0
14	VENDER_SPEC_DEV1_PST	特殊设备 1 存在于包中 始终为 0
13	CL22_EXT_PST	Clause 22 扩展在于包中 始终为 0
12:0	保留	必须保持复位值。

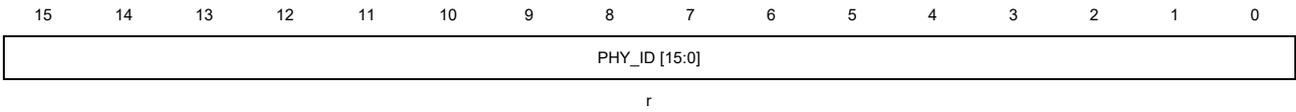
### 自动协商设备标识寄存器 (AN\_ID)

设备地址: 0x7

地址偏移: 0x0E

复位值: 0x0044

该寄存器可以按半字（16位）。



位/位域	名称	描述
15:0	PHY_ID[15:0]	PHY标识符bit[31-16] OUI (bits 3-18). OUI =00-11-05

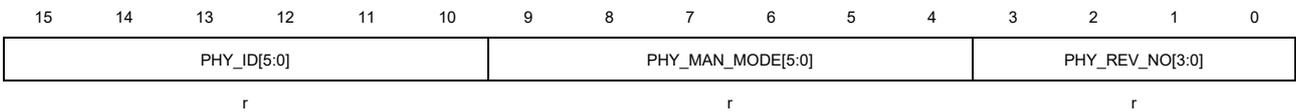
### 自动协商设备版本寄存器（AN\_VER）

设备地址：0x7

地址偏移：0x0F

复位值：0x1400

该寄存器可以按半字（16位）。



位/位域	名称	描述
15:10	PHY_ID[5:0]	PHY 标识符 bit [15-10] OUI bits 19-24
9:4	PHY_MAN_MODE[5:0]	制造商型号 制造商的型号(bits 5-0)，其中[5:4]=架构版本
3:0	PHY_REV_NO[3:0]	版本号(bits3-0) <a href="#">PCS 设备标识符寄存器 (PCS_ID)</a> bit 0 是 PHY 标识符的 LS 位

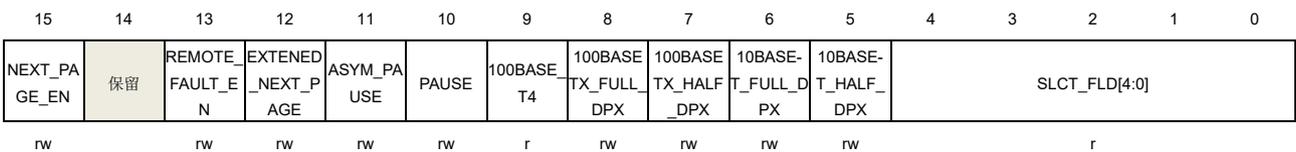
### 自动协商广播寄存器（AN\_ADV）

设备地址：0x7

地址偏移：0x10

复位值：0x0DE1

该寄存器可以按半字（16位）。



位/位域	名称	描述
15	NEXT_PAGE_EN	下一页使能 1: 设置为使用下一页

		0: 不要使用下一页
14	保留	必须保持复位值。
13	REMOTE_FAULT_EN	远程故障检测使能 1: 自动协商故障检测 0: 没有远程故障
12	EXTENDED_NEXT_PAGE	扩展下一页 PHY 不支持, 应该始终写 0
11	ASYM_PAUSE	非对称暂停功能 A6 技术能力 1: 支持非对称暂停能力 0: 不支持非对称暂停
10	PAUSE	暂停功能 A5 技术能力 1: 支持暂停功能 0: 不支持暂停功能
9	100BASE_T4	100BASE-T4 功能 PHY 不支持, 应该始终写 0
8	100BASETX_FULL_DPX	100BASE-X 全双工功能 1: 支持全双工 0: 不支持全双工 注意: 当自动协商被禁用时, 该位上的值将反映在 <a href="#">PHY 控制寄存器 (PHY MII CTL)</a> BIT6, BIT13 (FORCE_SPEED)和 BIT8 (FORCE_DUPLEX)中编程的值, 当 FORCE_SPEED 为 2'b01 和 FORCE_DUPLEX 是 1'b1, 那么这个位将是 1'b1, 反之亦然。
7	100BASETX_HALF_DPX	100BASE-X 半双工功能 1: 支持半双工 0: 不支持半双工 注意: 当自动协商被禁用时, 该位上的值将反映在 <a href="#">PHY 控制寄存器 (PHY MII CTL)</a> BIT6, BIT13 (FORCE_SPEED)和 BIT8 (FORCE_DUPLEX)中编程的值。当 FORCE_SPEED 为 2'b01, FORCE_DUPLEX 为 1'b0 时, 此位将为 1'b1, 反之亦然。
6	10BASE-T_FULL_DPX	10BASE-T 全双工功能 1: 支持全双工 0: 不支持全双工 注意: 当自动协商被禁用时, 该位上的值将反映在 <a href="#">PHY 控制寄存器 (PHY MII CTL)</a> BIT6, BIT13 (FORCE_SPEED)和 BIT8 (FORCE_DUPLEX)中编程的值。当 FORCE_SPEED 为 2'b00, FORCE_DUPLEX 为 1'b1 时, 此位将为 1'b1, 反之亦然。
5	10BASE-T_HALF_DPX	10BASE-T 半双工功能

- 1: 支持半双工
- 0: 不支持半双工

注意:当自动协商被禁用时,该位上的值将反映在 [PHY 控制寄存器 \(PHY MII CTL\)](#) BIT6, BIT13 (FORCE\_SPEED)和 BIT8 (FORCE\_DUPLEX)中编程的值。当 FORCE\_SPEED 为 2'b00, FORCE\_DUPLEX 为 1'b0 时,此位将为 1'b1,反之亦然。

4:0 SLCT\_FLD[4:0] 识别消息类型  
始终强制为 5'h01

### 自动协商链路伙伴(LP)性能寄存器 (AN\_LP\_ABILITY)

设备地址: 0x7  
地址偏移: 0x13  
复位值: 0x0000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	ACKNOWLEDGE	REMOTE_FAULT	EXTENDED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASE_TX_FULL_DPX	100BASE_TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SELECTOR_FIELD[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r				

位/位域	名称	描述
15	NEXT_PAGE	链接伙伴下一页请求 1: 链接伙伴正在请求下一页功能 0: 请求基页
14	ACKNOWLEDGE	链接伙伴确认接收 1: 链接伙伴确认接收成功 0: 没有接收
13	REMOTE_FAULT	链接伙伴检测到远程故障 1: 自动协商故障检测 0: 没有远程故障
12	EXTENDED_NEXT_PAGE	扩展下一页
11	ASYM_PAUSE	链接伙伴非对称暂停功能 A6技术能力 1: 支持非对称暂停能力 0: 不支持非对称暂停能力
10	PAUSE	链接伙伴对称暂停功能 A5技术能力 1: 支持对称暂停能力 0: 不支持对称暂停能力
9	100BASE_T4	A4技术能力

		链接伙伴 100BASE-T4 能力
8	100BASETX_FULL_DPX	链接伙伴100BASE-X全双工能力 1: 支持全双工 0: 不支持 注意: 当自动协商被禁用时, 该位的值将一直被设置为1'b1
7	100BASETX_HALF_DPX	链接伙伴100BASE-X半双工能力 1: 支持半双工 0: 不支持 注意: 当自动协商被禁用时, 该位的值将一直被设置为1'b1
6	10BASE-T_FULL_DPX	链接伙伴10BASE-T全双工能力 1: 支持全双工 0: 不支持 注意: 当自动协商被禁用时, 该位的值将一直被设置为1'b1
5	10BASE-T_HALF_DPX	链接伙伴10BASE-T半双工能力 1: 支持半双工 0: 不支持 注意: 当自动协商被禁用时, 该位的值将一直被设置为1'b1
4:0	SELECTOR_FIELD[4:0]	链路伙伴标识消息类型应该是5'h01

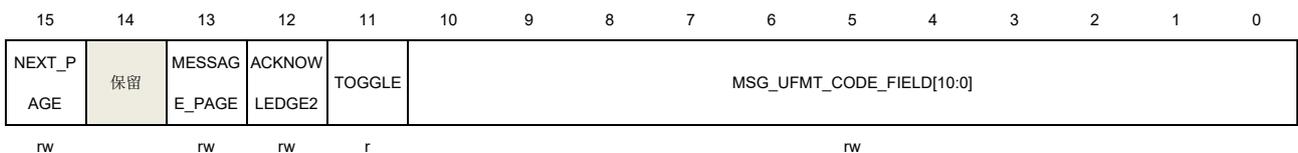
### 自动协商 XNP 发送寄存器 (AN\_XNP\_TRANSMIT)

设备地址: 0x7

地址偏移: 0x16

复位值: 0x2001

该寄存器可以按半字 (16位)。



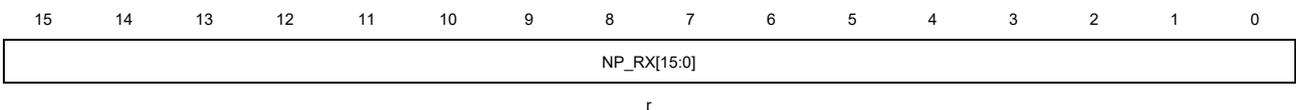
位/位域	名称	描述
15	NEXT_PAGE	最后一页 指示这是否是最后一页 1: 附加的下一页将紧随其后 0: 下一页是最后一页
14	保留	必须保持复位值
13	MESSAGE_PAGE	消息页或未格式化页 指示这是消息页或未格式化的页 1: 消息页面 0: 无格式页面

12	ACKNOWLEDGE2	依从信息的能力 1: 依从信息 0: 无法依从信息
11	TOGGLE	切换 切换位由SW自动计算，软件可以忽略
10:0	MSG_UFMT_CODE_FIELD[10:0]	消息/未格式化代码字段

### 自动协商链路伙伴 XNP 性能寄存器 (AN\_LP\_XNP\_ABILITY)

设备地址: 0x7  
地址偏移: 0x19  
复位值: 0x0000

该寄存器可以按半字 (16位)。

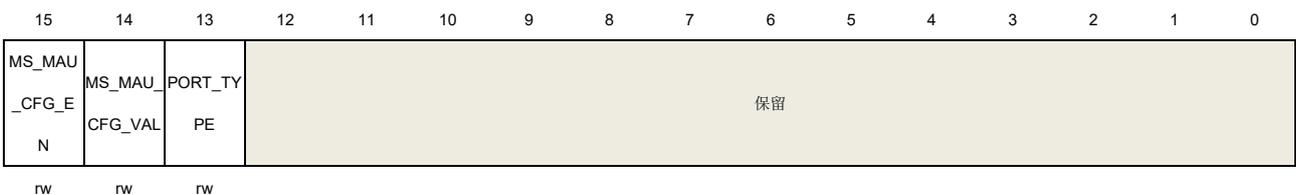


位/位域	名称	描述
15	NP_RX[15:0]	从链路伙伴收到下一页

### 主-从控制寄存器(MS\_CTL)

设备地址: 0x7  
地址偏移: 0x20  
复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15	MS_MAU_CFG_EN	启用主从手动配置 1: 启用主从手动配置 0: 关闭主从手动配置
14	MS_MAU_CFG_VAL	主从手动配置值 1: 配置PHY为主设备 0: 配置PHY为从设备
13	PORT_TYPE	端口类型 1: 多端口设备

0: 单端口设备

12:0 保留 必须保持复位值

### 主-从状态寄存器(MS\_STS)

设备地址: 0x7

地址偏移: 0x21

复位值: 0x0000

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS_CFG_FAULT	MS_CFG_RES	保留													
r	r														

位/位域	名称	描述
15	MS_CFG_FAULT	检测到主从配置错误 1: 检测到主从配置错误 0: 未检测到主从配置错误
14	MS_CFG_RES	主从配置解析 1: 本地配置解析为主 0: 本地配置解析为从
13:0	保留	必须保持复位值

### EEE 广播寄存器 (EEE\_ADV)

设备地址: 0x7

地址偏移: 0x3C

复位值: 0x0003

该寄存器可以按半字 (16位)。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留									10GBASE_KR_EEE	10GBASE_KX4_EEE	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	保留
									r	r	r	r	rW	rW	

位/位域	名称	描述
15:7	保留	必须保持复位值。
6	10GBASE_KR_EEE	10GBASE-KR EEE 不支持, 始终为0
5	10GBASE_KX4_EEE	10GBASE-KR EEE

		不支持，始终为0
4	1000BASE_KX_EEE	10GBASE-KR EEE 不支持，始终为0
3	10GBASE_T_EEE	10GBASE-KR EEE 不支持，始终为0
2	1000BASE_T_EEE	1000Base-T EEE 1: 广播1000BASE-T具有EEE能力 0: 不广播1000BASE-T具有EEE能力
1	100BASE_TX_EEE	100Base-TX EEE 1: 广播100BASE-TX具有EEE能力 0: 不广播100BASE-TX具有EEE能力
0	保留	必须保持复位值。

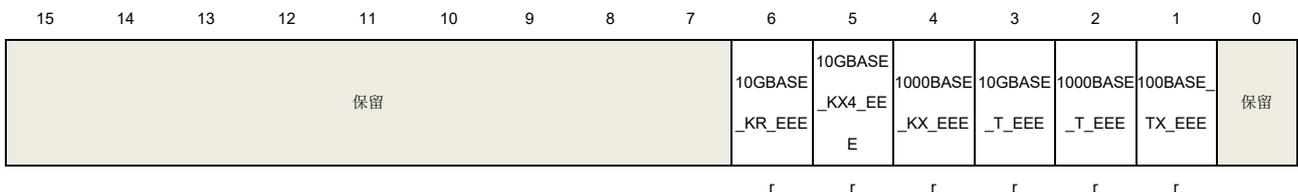
### EEE 链路伙伴(LP)性能寄存器 (EEE\_LP\_ABILITY)

设备地址: 0x7

地址偏移: 0x3D

复位值: 0x0000

该寄存器可以按半字 (16位)。



位/位域	名称	描述
15:7	保留	必须保持复位值。
6	10GBASE_KR_EEE	10GBASE-KR EEE 1: 链接伙伴广播10GBASE-KR的EEE功能 0: 链接伙伴不广播10GBASE-KR的EEE功能
5	10GBASE_KX4_EEE	10GBASE-KX4 EEE 1: 链接伙伴广播10GBASE-KX4的EEE功能 0: 链接伙伴不广播10GBASE-KX4的EEE功能
4	1000BASE_KX_EEE	1000BASE-KX EEE 1: 链接伙伴广播1000BASE-KX的EEE功能 0: 链接伙伴不广播1000BASE-KX的EEE功能
3	10GBASE_T_EEE	10GBASE-T EEE 1: 链接伙伴广播10GBASE-T的EEE功能 0: 链接伙伴不广播10GBASE-T的EEE功能

---

2	1000BASE_T_EEE	1000BASE-T EEE 1: 链接伙伴广播1000BASE-T的EEE功能 0: 链接伙伴不广播1000BASE-T的EEE功能
1	100BASE_TX_EEE	100BASE-TX EEE 1: 链接伙伴广播100BASE-TX的EEE功能 0: 链接伙伴不广播100BASE-TX的EEE功能
0	保留	必须保持复位值。

## 9. EtherCAT

### 9.1. 简介

GDSCN 是一种 EtherCAT 从站控制器(ESC)。它作为 EtherCAT 现场总线和从站应用之间的接口，负责 EtherCAT 通信。GDSCN 支持广泛的应用。EtherCAT 控制器拥有 8K 字节的过程数据内存(PDRAM)和 8 个现场总线内存管理单元(FMMUs)，每个单元的任务是将逻辑地址映射到物理地址。EtherCAT 从站控制器还包括 8 个 SyncManager，允许 EtherCAT 和本地应用之间进行数据交换。每个 SyncManager 的方向和操作模式由 EtherCAT 主设备配置。有两种工作模式可用：缓冲模式和邮箱模式。在缓冲模式下，微控制器和 EtherCAT 主可以同时写入设备。GDSCN 中的缓冲区始终包含最新数据。如果新数据在旧数据可以读取之前到达，旧数据将丢失。在邮箱模式下，微控制器和 EtherCAT 主通过握手访问缓冲区，确保不会丢失数据。

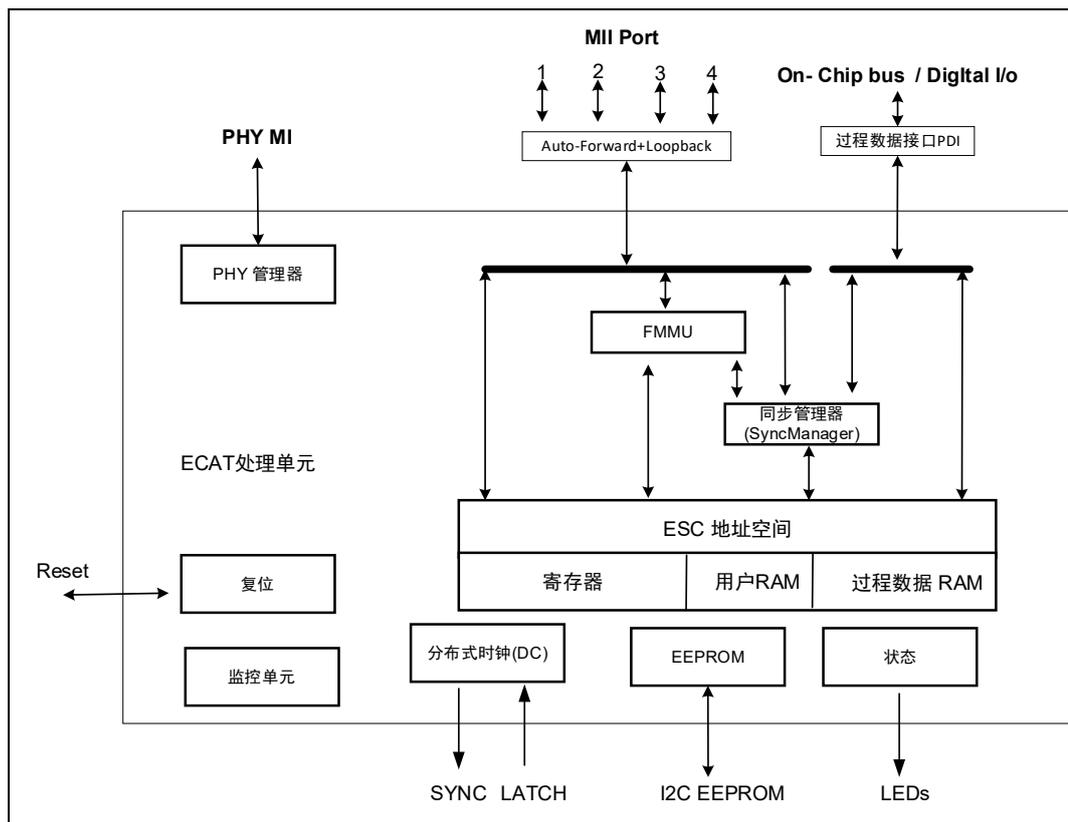
### 9.2. 主要特征

- 端口支持：2个内部PHY接口和1个外部MII；
- 8个现场总线存储管理单元（FMMUs）；
- 8KB过程数据内存（PDRAM）；
- 支持64位分布式时钟，与其他EtherCAT设备同步；
- 8个SyncManager实体；
- DC同步时间小于1微秒。

#### 9.2.1. 结构框图

ESC 的功能模块图如 [图 9-1. EtherCAT 系统框图](#) 所示。

图 9-1. EtherCAT 系统框图



## 9.2.2. EtherCAT 从站控制器功能块

### ■ EtherCAT接口

EtherCAT 接口或端口将 ESC 连接到其他 EtherCAT 从站和主站。MAC 层是 ESC 的组成部分。物理层可能是以太网。对于以太网端口，内部以太网 PHY 连接到 ESC 的 MII 端口。EtherCAT 的传输速度固定为 100Mbit/s，采用全双工通信。链路状态和通信状态会报告给监控设备。GDSCN 使用三个端口，端口 0/1/2。

### ■ EtherCAT处理单元

EtherCAT 处理单元（EPU）接收、分析并处理 EtherCAT 数据流。它在逻辑上位于端口 0 和端口 3 之间。EtherCAT 处理单元的主要目的是启用和协调对 ESC 内部寄存器和存储器空间的访问，这些寄存器和存储器空间既可以从 EtherCAT 主设备访问，也可以通过 PDI 从本地应用程序访问。主设备和从设备应用程序之间的数据交换类似于双端口内存（进程内存），通过特殊功能增强，例如一致性检查（SyncManager）和数据映射（FMMU）。EtherCAT 处理单元包含 EtherCAT 从设备的主要功能块，除了自动转发、环回功能和 PDI。

### ■ 自动转发器

自动转发器接收以太网帧，进行帧检查，并将帧转发到环回功能。自动转发器生成接收帧的时间戳。

### ■ 环回功能

环回功能将以太网帧转发到下一个逻辑端口，如果某个端口没有连接，或者端口不可用，或者

该端口的环回被关闭。端口 0 的环回功能将帧转发到 EtherCAT 处理单元。环回设置可以由 EtherCAT 主控器控制。

#### ■ FMMU

现场总线内存管理单元用于将逻辑地址映射到 ESC 的物理地址。

#### ■ 同步管理器

同步管理器负责在 EtherCAT 主设备和从设备之间进行一致的数据交换和邮箱通信。每个同步管理器的通信方向都可以配置。读或写事务可能会为 EtherCAT 主设备和附加的微控制器生成事件。同步管理器负责区分 ESC 和双端口内存之间的主要区别，因为它们将地址映射到不同的缓冲区，并根据同步管理器的状态阻止访问。这也是 PDI 带宽限制的根本原因。

#### ■ 监控

监控单元包含错误计数器和看门狗。看门狗用于观察通信并在出现错误时返回到安全状态。错误计数器用于错误检测和分析。

#### ■ PHY管理器

PHY 管理单元通过 MII 管理接口与以太网 PHY 通信。这可以由主设备或从设备使用。MII 管理接口由 ESC 本身用于在接收到增强型链路检测机制的错误后选择性地重新启动自动协商，以及用于可选的 MII 链路检测和配置功能。

#### ■ 分布式时钟

分布式时钟 (DC) 允许精确同步地生成输出信号和输入采样，以及生成事件的时间戳。这种同步可能覆盖整个 EtherCAT 网络。

#### ■ EEPROM

需要一个非易失性存储器来存储 EtherCAT 从站信息 (ESI)，通常是一个 I<sup>2</sup>C EEPROM。

#### ■ 状态/指示灯

状态模块提供 ESC 和应用程序状态信息。它控制外部 LED，如应用程序运行 LED/错误 LED 和端口链接/活动 LED。

## 9.3. 功能描述

### 9.3.1. 过程数据接口 (PDI)

进程数据接口 (PDI) 实现了从应用程序与 ESC 之间的连接。定义了几种类型的 PDI，包括串行和并行微控制器接口以及数字 I/O 接口。由于 EtherCAT 和 PDI 对内存、寄存器以及尤其是同步管理器的访问高度依赖，内部 PDI 接口最多可以实现约 12.5Mbyte/s 的吞吐量。

表 9-1. EtherCAT 的 PDI

PDI数量(PDI控制寄存器0x0140)	PDI名称
0	接口失效
4	Digital I/O

128

On-chip bus

## PDI 选择和配置

通常，PDI 的选型和配置是 SII EEPROM 的 ESC 配置区域的一部分。ESC 在上电时选择并配置 PDI。在这种情况下，ESC 配置区域应该反映实际设置，尽管不是由 ESC 本身决定的。释放复位后，PDI 处于激活状态，这使得微控制器能够实现 EEPROM 仿真。在 EEPROM 未加载时，注意数字输出信号和 DC 同步信号，以实现正确的输出行为。

## PDI 寄存器功能通过写入确认

一些 ESC 的功能是通过写入或读取单个字节地址、同步管理器缓冲区变更或 AL 事件请求确认来触发的。随着微控制器数据总线宽度的增加，这可能会导致限制甚至问题。

由于大多数微控制器在使用字节使能信号进行写访问时没有限制，但许多微控制器在读取访问时不使用字节使能信号，它们期望获得整个数据总线的宽度的读取数据。读取单个字节是不可能的。这可能导致问题，特别是意外读取触发某些 ESC 功能的字节地址时。考虑 SyncManager 缓冲区区域从 0x1000-0x1005。一个 32 位微控制器应用程序可能会逐字节读取缓冲区。对 0x1000 的第一次访问将打开缓冲区，并且还会读取 0x1001-0x1003。第二次访问将读取 0x1001，并且还将读取 0x1000/0x1002-0x1003。当地址 0x1004 要被读取时，问题就出现了，因为这也会读取 0x1005。0x1005 的数据被丢弃了，但缓冲区被关闭了。当微控制读取 0x1005 时，它总是得到 0 - 数据似乎被损坏了。类似的问题也出现在 DC 同步信号确认（寄存器 0x098E 和 0x098F）上。32 位微控制器总是会同时确认 SYNC0 和 SYNC1，不可能分别确认。

这个问题可以通过使能 PDI 寄存器功能确认写入来解决。在这种模式下，所有原本由读访问触发的功能现在都由相应的写访问触发——这些写访问使用字节使能，因此可以限制在特定的字节。

此功能通过 IP 核配置启用。在使用此功能之前，必须通过微控制器应用程序检查 PDI 信息寄存器 0x014E[0] 的当前状态。

此功能影响 SyncManager 缓冲区的读取以及从 PDI 端读取某些寄存器。EtherCAT 主设备端没有任何变化。请参阅 [同步管理器\(SyncManager\)](#)，了解同步管理器的行为。以下寄存器受到 PDI 寄存器功能通过写入确认的影响：

**表 9-2. 通过写入确认 PDI 寄存器功能的受影响寄存器**

地址	名称	触发函数
任意	SyncManager内存和地址	读取SyncManager缓冲区，然后写入缓冲区结束地址以确认缓冲区读取。
0x0120:0x0121	AL控制	AL控制发生变化后，读取 0x0120:0x0121，然后写入 0x0120确认读取。
0x0440	看门狗状态进程数据	读取0x0440，然后写入0x0440以清除AL事件请求0x0220[6]
0x0806+X*16	SyncManager激活	读取0x0806+X*16，然后写入 0x0806(SyncManager 0)，只为清

		除所有SyncManager的AL事件请求0x0220[4]
0x098E	SYNC0状态	读取0x098E，然后写入0x098E以确认数据同步状态0x098E[0]
0x098F	SYNC1状态	读取0x098E，然后写入0x098E以确认数据同步状态0x098F[0]
0x09B0:0x09B7	Latch0时间上升沿锁存	读取0x09B0: 0x09B7，然后写入0x09B0，清除DC Latch0状态0x09AE[0]
0x09B8:0x09BF	Latch0时间下降沿锁存	读取0x09B8: 0x09BF，然后写入0x09B8以清除DC Latch0状态0x09AE[1]
0x09C0:0x09C7	Latch1时间上升沿锁存	读取0x09C0: 0x09C7，然后写入0x09C0，清除DC Latch1状态0x09AF[0]
0x09C8:0x09CF	Latch1时间下降沿锁存	读取0x09C8: 0x09CF，然后写入0x09C8以清除DC Latch1状态0x09AF[1]

### 9.3.2. FMMU

现场总线存储器管理单元（FMMU）通过内部地址映射的方式将逻辑地址转换为物理地址。因此，FMMU 允许使用逻辑地址来访问跨越多个从设备的分段数据：一个数据报文在几个任意分布的 EtherCAT 中寻址数据。每个 FMMU 通道将一个连续的逻辑地址空间映射到从设备的连续物理地址空间。FMMU 支持的访问类型可以配置为读取、写入或读写。

#### ■ FMMU设置的限制

ESC 中的 FMMU 受到限制。如果使用位映射（逻辑起始位 $\neq 0$ ，逻辑停止位 $\neq 7$ ，或物理起始位 $\neq 0$ ），则同一方向（读或写）的两个 FMMU 的逻辑地址范围必须至少相隔 3 个未配置任何同类型 FMMU 的逻辑字节。

#### ■ 附加的FMMU特性

- 每个逻辑地址字节最多可以由一个FMMU（读）加上一个FMMU（写）映射，或者由一个FMMU（读/写）映射。如果两个或更多的FMMU（具有相同的方向 - 读或写）被配置用于相同的逻辑字节，则使用编号较低的FMMU（较低的配置地址空间），其他FMMU将被忽略。
- 一个或多个FMMU可能指向相同的物理内存，它们都被使用。碰撞不会发生。
- 使用一个读写FMMU或使用两个FMMU——一个用于读取，另一个用于写入——对于相同的逻辑地址来说是一样的。
- 读写型FMMU不能与同步管理器一起使用，因为独立的读写同步管理器不能配置为使用相同的（或重叠的）物理地址范围。
- 位读取支持在任何地址进行。没有映射到逻辑地址的位在EtherCAT数据报中不会被改变。例如，这允许将来自多个ESC的位映射到同一个逻辑字节中。
- 一个针对未在ESC中配置的逻辑地址空间的帧/数据报，不会改变ESC中的数据，并

且ESC中的数据也不会被放置在帧/数据报中。

### 9.3.3. 同步管理器(SyncManager)

ESC 的内存可以用来在 EtherCAT 主控器和连接到 PDI 的微控制器上本地应用程序之间交换数据，没有任何限制。使用内存进行通信有一些问题，这些问题可以通过 ESC 内的同步管理器来解决：

- 数据一致性无法保证。必须在软件中实现信号量以协调地交换数据。
- 数据安全无法保证。必须在软件中实现安全机制。
- EtherCAT 主站和应用程序都必须轮询内存，以了解对方何时完成访问。

同步管理器能够在 EtherCAT 主设备和本地应用程序之间实现一致且安全的数据处理，并生成中断以通知双方变化。

同步管理器由 EtherCAT 主控制器配置。通信方向是可配置的，通信模式（缓冲模式和邮箱模式）也是可配置的。同步管理器使用位于内存区域的缓冲区来交换数据。对这个缓冲区的访问由同步管理器的硬件控制。

缓冲区必须从起始地址开始访问，否则访问将被拒绝。访问起始地址后，整个缓冲区都可以被访问，包括再次访问起始地址，可以一次性访问整个缓冲区，也可以分多次访问。当访问结束地址时，缓冲区访问结束，缓冲区状态随后改变，并且如果配置了，将生成中断或看门狗触发脉冲。在一个帧内，结束地址不能被访问两次。

同步管理器支持两种通信模式：

- 缓冲模式
  - 缓冲模式允许双方，即EtherCAT主控器和本地应用程序，随时访问通信缓冲区。消费者总是获得生产者写入的最新一致的缓冲区，而生产者可以随时更新缓冲区的内容。如果缓冲区的写入速度比读取速度快，旧数据将被丢弃。
  - 缓冲模式通常用于循环处理数据。
- 邮箱模式
  - 邮箱模式实现了一种数据交换的握手机制，以确保数据不会丢失。每一方，无论是 EtherCAT 主控器还是本地应用程序，只有在另一方完成访问后才能获得对缓冲区的访问。首先，主机向缓冲区写入数据。然后，缓冲区被锁定以防止写入，直到从机读取完毕。之后，主机再次获得写入访问权限，而缓冲区则对从机锁定。
  - 邮箱模式通常用于应用层协议。

同步管理器仅在帧的帧检验序列（FCS）正确时才接受由主机引起的缓冲区变化，因此，缓冲区的变化在帧结束不久后生效。

同步管理器的配置寄存器位于从寄存器地址 0x0800 开始的位置。

### 9.3.4. 分布式时钟（DC）

EtherCAT 从控制器分布式时钟（DC）单元支持以下功能：

- 从机（与主机）之间的时钟同步

- 同步输出信号的生成 (SyncSignals)
- 输入事件的精确时间戳 (LatchSignals)
- 产生同步中断
- 同步数字输出更新
- 同步数字输入采样

本设备支持 64 位分布式时钟，详细介绍如下。

EtherCAT 提供了两个输入引脚 (SYNC 和 LATCH)，用于对外部事件进行时间戳。记录了上升沿和下降沿的时间戳。这些引脚与 SYNC0 和 LATCH0 输出引脚共享，分别用于指示时间事件的发生。SYNC/SYNC0 和 LATCH/LATCH0 引脚的功能由 YNC/LATCH PDI 配置寄存器中的 SYNC0/LATCH0 和 SYNC/LATCH 配置位决定。

当设置为 SYNC0/LATCH0 功能时，输出类型 (推挽式与开漏/开源) 和输出极性由同步/锁存 PDI 配置寄存器中的 SYNC0 输出驱动器/极性位和 LATCH0 输出驱动器/极性位决定。

### 9.3.5. EtherCAT 状态机(ESM)

EtherCAT 状态机 (ESM) 负责协调主机和从机。

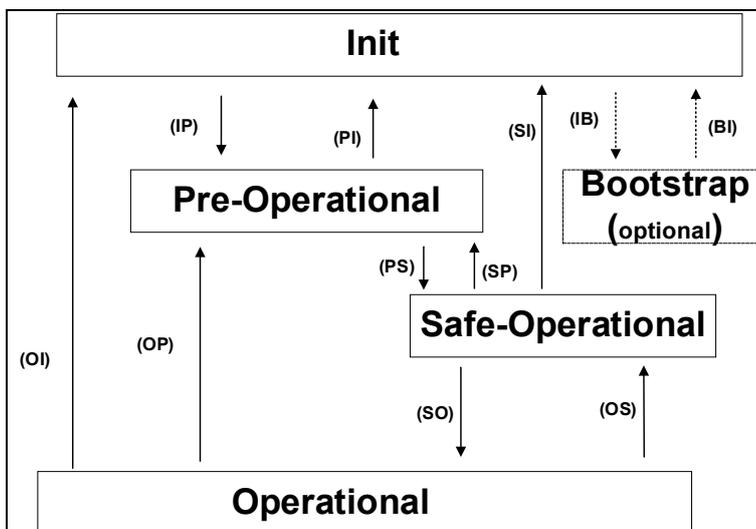
应用程序在启动时和运行期间的状态变化通常由主机的请求来启动。这些状态变化在相关操作执行后由本地应用程序确认。本地应用程序也可能发生未经请求的状态变化。

EtherCAT 从站应支持四种状态，外加一种可选状态：

- 初始化 (复位后的状态)
- 预操作
- 安全操作
- 运营
- 启动 (可选)

状态和允许的状态更改显示在 [图 9-2. EtherCAT 状态机](#)。

图 9-2. EtherCAT 状态机



**注意:** 并非所有状态变化都是可能的，从“初始化”到“操作状态”的转换需要以下序列：初始化 ->

预操作状态 -> 保存操作状态 -> 操作状态。

每个状态定义了所需的服务。在确认状态变更之前，所有为请求状态提供或停止所需的服务都必须得到满足。

### 9.3.6. EEPROM

EtherCAT 从站控制器使用一个强制性的 RAM（通常是一个带有 I<sup>2</sup>C 接口的串行 EEPROM）来存储 EtherCAT 从站信息（ESI）。根据 ESC 的不同，支持从 1 Kbit 到 4 Mbit 的 EEPROM 大小。

EEPROM 结构如 [图 9-3. EEPROM 布局](#) 所示。EEPROM 布局，ESI 使用字地址。

图 9-3. EEPROM 布局

Word	EtherCAT Slave CONTROLLER Configuration Area							
0	EtherCAT Slave CONTROLLER Configuration Area							
8					VendorId	ProductCode	RevisionNo	SerialNo
16					Hardware Delays		Bootstrap Mailbox Config	
24					Mailbox Sync Man Config		Reserved	
64	Additional Information(Subdivided in Categories)...							
	Category Strings							
	Category Generals							
	Category FMMU							
	Category SyncManager							
	Category Tx- / RxPDO for each PDO							

至少存储在地址范围从字 0 到 63（0x00 到 0x3F）的信息是强制性的，以及通用类别（绝对最小的 EEPROM 大小是 2Kbit，具有许多类别的复杂设备应该配备 32Kbit EEPROM 或更大的存储器）。ESC 配置区域由 ESC 用于配置。所有其他部分由主机或本地应用程序使用。

### 9.3.7. 复位

EtherCAT 模块提供了两个寄存器，[ESC 复位 ECAT 寄存器 \(ESC RESET ECAT\)](#) 和 [ESC 复位 PDI 寄存器 \(ESC RESET PDI\)](#)，它们可以分别被 EtherCAT 主站和从站访问触发一个复位请求。

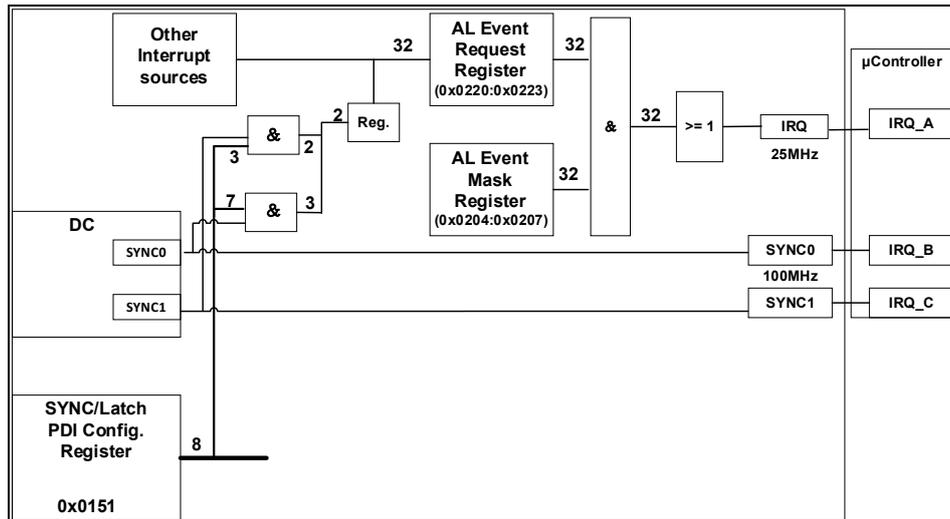
### 9.3.8. 中断

EtherCAT 支持两种类型的中断：针对微控制器的 AL 事件请求，以及针对 EtherCAT 主控的 EtherCAT 事件请求。此外，分布式时钟同步信号也可以作为微控制器的中断使用。

## 事件请求（PDI 中断）

AL 事件请求可以通过 PDI 中断请求信号（IRQ/SPI\_IRQ 等）向微控制器发出信号。对于 IRQ 生成，AL 事件请求寄存器（0x0220:0x0223）与 AL 事件掩码寄存器（0x0204:0x0207）通过逻辑与操作组合，然后所有结果位通过逻辑或操作组合成一个中断信号。IRQ 信号的输出驱动特性可以使用 SYNC/LATCH PDI 配置寄存器（0x0151）进行配置。AL 事件掩码寄存器允许选择对微控制器和应用程序处理相关的中断。

图 9-4. PDI 中断屏蔽和中断信号



DC 同步信号可以用两种方式用于中断生成：

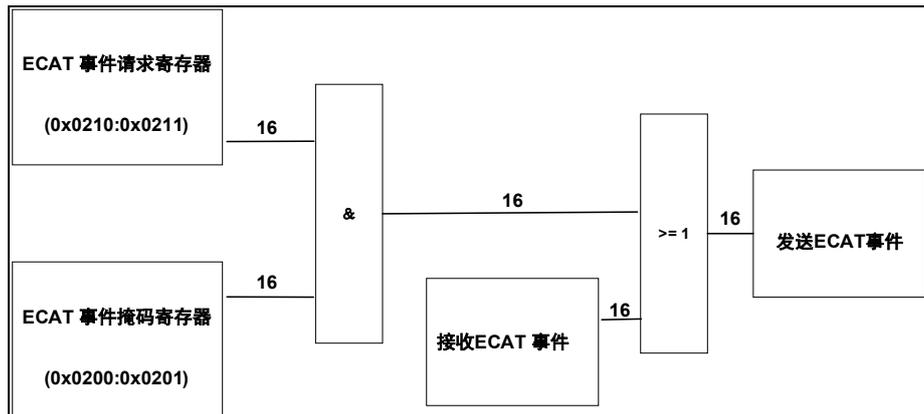
- DC同步信号被映射到AL事件请求寄存器（配置为SYNC/LATCH PDI配置寄存器0x0151.3/7）。在这种情况下，所有来自ESC到微控制器的中断被合并为一个IRQ信号，并且分布式时钟LATCH0/1输入仍然可以使用。IRQ信号的抖动约为40ns。
- DC同步信号直接连接到微控制器的中断输入。微控制器可以更快地响应DC同步信号中断（无需读取AL请求寄存器），但它需要更多的中断输入。同步信号的抖动约为12ns。DC Latch功能的可用性仅限于一个Latch输入，或者根本不可用（如果两个DC同步输出都被使用）。

## ECAT 事件请求（ECAT 中断）

ECAT 事件请求用于通知 EtherCAT 主站从站事件。ECAT 事件利用 EtherCAT 数据报中的 IRQ 字段。ECAT 事件请求寄存器（0x0210:0x0211）与 ECAT 事件掩码寄存器（0x0200:0x0201）通过逻辑与操作结合。由此产生的中断位与传入的 ECAT IRQ 字段通过逻辑或操作结合，并写入传出的 ECAT IRQ 字段。ECAT 事件掩码寄存器允许选择对 EtherCAT 主控制器相关并由主应用程序处理的中断。

注意：主站无法区分哪个从站（甚至可能是多个）是中断的来源。

图 9-5. EtherCAT 中断屏蔽



### 清除意外中断

事件请求寄存器和清除中断的寄存器操作旨在独立访问，即通过单独的以太网帧或单独的 PDI 访问。否则可能会错过中断和/或数据。

## 9.3.9. LED

EtherCAT 从站控制器支持 LED (RUNLED) 以反映链路状态和 AL 状态。ESC 的 LED 输出由 AL 状态寄存器 (0x0130) 控制，并支持以下状态，这些状态会自动转换为闪烁代码。

EtherCAT 内核配置通过 RUN LED 覆盖寄存器直接控制 RUN LED。

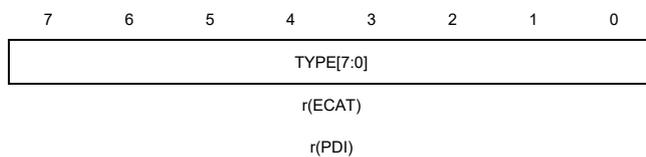
## 9.4. ESC 寄存器定义

### 9.4.1. ESC 类型寄存器 (ESC\_TYPE)

地址偏移: 0x0000

复位值: 0xBC

该寄存器可以按字节 (8位) 访问。



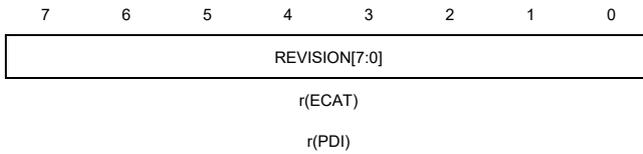
位/位域	名称	描述
7:0	TYPE[7:0]	EtherCAT 控制器类型 0xBC: GDSCN

### 9.4.2. ESC 版本寄存器 (ESC\_REVISION)

地址偏移: 0x0001

复位值：0x0000

该寄存器可以按字节（8位）访问。



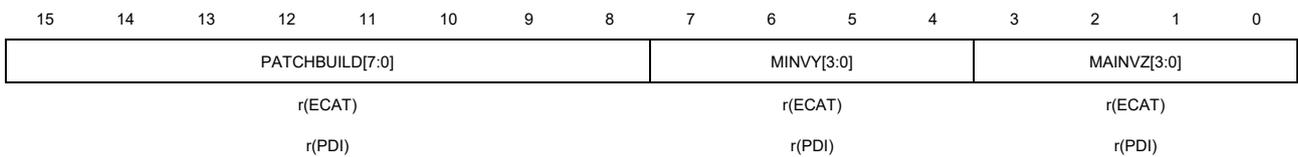
位/位域	名称	描述
7:0	REVISION[7:0]	EtherCAT 控制器版本

### 9.4.3. ESC 编译寄存器（ESC\_BUILD）

地址偏移：0x0002

复位值：0x0000

该寄存器可以按半字（16位）访问。



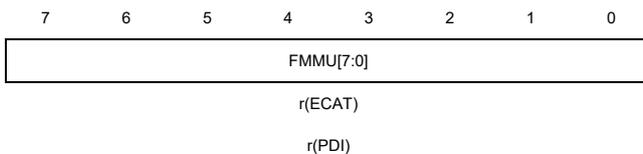
位/位域	名称	描述
15:8	PATCHBUILD[7:0]	补丁级别/开发版本： 0x00：原始发布 0x01-0x0F：原始版本的补丁发布 0x10-0xFF：开发版本
7:4	MINVY[3:0]	次要版本 Y
3:0	MAINVZ[3:0]	维护版本 Z

### 9.4.4. ESC FMMU 数目寄存器（ESC\_FMMUS）

地址偏移：0x0004

复位值：0x08

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:0	FMMU[7:0]	该字段详细说明 EtherCAT 从控制器支持的 FMMU 通道(或实体)的数量。设备提供 8

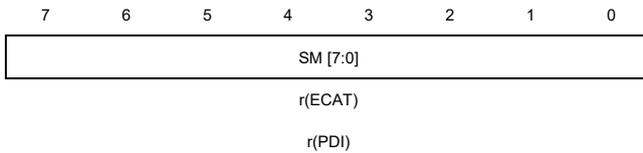
个通道(或实体)。

#### 9.4.5. ESC 同步管理器的数目寄存器 (ESC\_SYNCMANAGERS)

地址偏移: 0x0005

复位值: 0x08

该寄存器可以按字节 (8位) 访问。



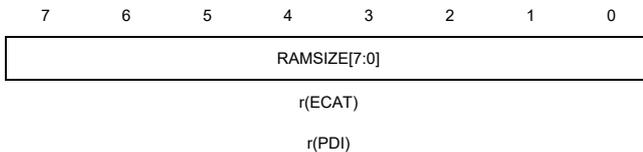
位/位域	名称	描述
7:0	SM[7:0]	支持的 SyncManager 实体数, 设备提供 8 个实体。

#### 9.4.6. ESC RAM 大小寄存器 (ESC\_RAMSIZE)

地址偏移: 0x0006

复位值: 0x08

该寄存器可以按字节 (8位) 访问。



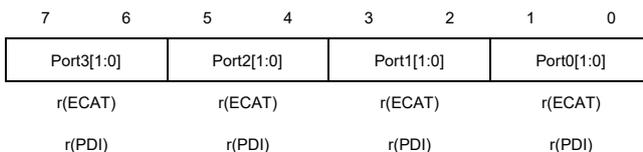
位/位域	名称	描述
7:0	RAMSIZE[7:0]	进程数据内存大小支持 8Kbyte

#### 9.4.7. ESC 端口描述寄存器 (ESC\_PORT\_DESCRIPTION)

地址偏移: 0x0007

复位值: 0x3F

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:6	Port3[1:0]	端口 3 配置 该字段详细说明端口3的配置。

		00: 未实现
		01: 未配置
		10: EBUS
		11: MII/RMII
5:4	Port2[1:0]	端口 2 配置 该字段详细说明端口2的配置。
		00: 未实现
		01: 未配置
		10: EBUS
		11: MII/RMII
3:2	Port1[1:0]	端口 1 配置 该字段详细说明端口1的配置。
		00: 未实现
		01: 未配置
		10: EBUS
		11: MII/RMII
1:0	Port0[1:0]	端口 0 配置 该字段详细说明端口0的配置。
		00: 未实现
		01: 未配置
		10: EBUS
		11: MII/RMII

#### 9.4.8. ESC 功能支持寄存器 (ESC\_FEATURES\_SUPPORTED)

地址偏移: 0x0008

复位值: 0x01CC

该寄存器可以按字节半字 (16位) 访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0		
		r(ECAT)													
		r(PDI)													

位/位域	名称	描述
15:12	保留	必须保持复位值。
11	FS11	固定 FMMU/SyncManager 配置 0: 可变配置 1: 固定配置
10	FS10	EtherCAT 读/写命令支持(BRW,APRW,FPRW) 0: 支持 1: 不支持

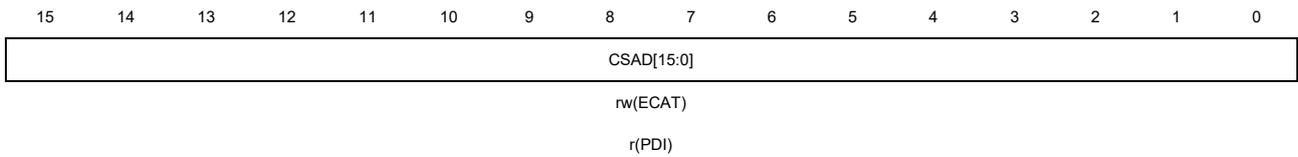
9	FS9	EtherCAT LRW 命令支持 0: 支持 1: 不支持
8	FS8	NOTE: This feature refers to registers 0x981[7:3] and 0x0984  增强型 DC SYNC 激活 0: 不可用 1: 可用  注意: 该功能请参 <a href="#">ESC 激活寄存器 (ESC REGISTER ACTIVE)</a> 和 <a href="#">ESC 激活状态寄存器 (ESC ACTIVE STATUS)</a>
7	FS7	单独处理 FCS 错误 0: 不支持 1: 支持, 具有错误 FCS 的帧以及额外的半字节将在转发的接收计数器中单独计数
6	FS6	增强型链路检测 MII 0: 不可用 1: 可用
5	FS5	增强型链路检测 EBUS 0: 不可用 1: 可用
4	FS4	低抖动 EBUS 0: 不可用, 正常抖动 1: 可用, 最低抖动
3	FS3	分布式时钟 (宽度) 0: 32 位 1: 64 位
2	FS2	分布式时钟 0: 不可用 1: 可用
1	FS1	未使用寄存器接口 0: 允许 1: 不支持
0	FS0	FMMU 操作 0: 针对位操作 1: 针对字节操作

#### 9.4.9. ESC 配置的站点地址寄存器 (ESC\_STATION\_ADDRESS)

地址偏移: 0x0010

复位值: 0x0

该寄存器可以按半字（16位）访问。



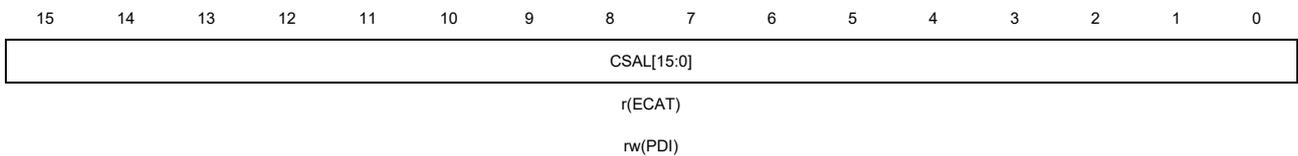
位/位域	名称	描述
15:0	CSAD[15:0]	用于节点寻址的地址(FPRD/FPWR/FPRW/FRMW 命令)。

#### 9.4.10. ESC 配置的站点别名寄存器（ESC\_STATION\_ALIAS）

地址偏移：0x0012

复位值：0x0

该寄存器可以按半字（16位）访问。



位/位域	名称	描述
15:0	CSAL[15:0]	该字段包含用于节点寻址（FPRD/FPWR/FPRW/FRMW命令）的别名地址。该别名的使用由 <a href="#">ESC DL控制寄存器（ESC_DL_CONTROL）</a> 的Bit 24位激活。 注意：仅在上电复位后第一次装载 EEPROM 时接受 EEPROM 值。

#### 9.4.11. 写使能寄存器（WRITE\_ENABLE）

地址偏移：0x0020

复位值：0x0

该寄存器可以按字节（8位）访问。



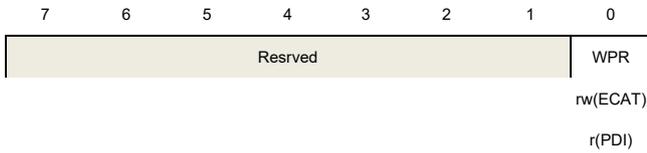
位/位域	名称	描述
7:1	保留	必须保持复位值。
0	WEN	如果使能了写保护，必须在同一以太网帧中写入该寄存器（值无关），之后才允许对该站执行其他写操作。写保护在该帧之后仍有效（如果写寄存器保护寄存器没有变化）

### 9.4.12. ESC 写保护寄存器 (ESC\_WRITE\_PROTECTION)

地址偏移: 0x0021

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



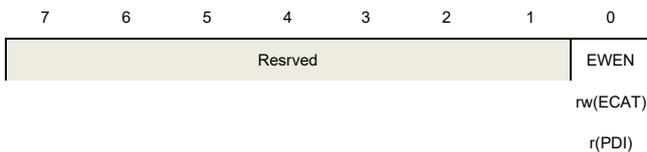
位/位域	名称	描述
7:1	保留	必须保持复位值。
0	WPR	写寄存器保护 0: 禁止保护 1: 使能保护 除了 0020 和 0030h 之外, 寄存器 0000h-0F7Fh 均受写保护。

### 9.4.13. ESC 写寄存器使能寄存器 (ESC\_WRITE\_ENABLE)

地址偏移: 0x0030

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



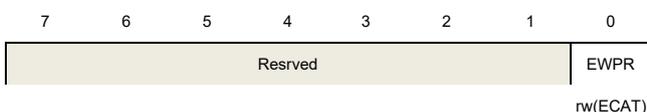
位/位域	名称	描述
7:1	保留	必须保持复位值。
0	EWEN	如果使能了ESC写保护, 必须在同一以太网帧中写入该寄存器 (值无关), 之后才允许对该站执行其他写操作。ESC写保护在该帧后仍有效 (如果ESC写寄存器保护寄存器没有变化)

### 9.4.14. ESC 写寄存器保护寄存器 (ESC\_WRITE\_PROTECTION)

地址偏移: 0x0031

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



r(PDI)

位/位域	名称	描述
7:1	保留	必须保持复位值。
0	EWPR	ESC写寄存器保护 0: 禁止保护 1: 使能保护 除了0030h之外, 所有区域均受写保护。

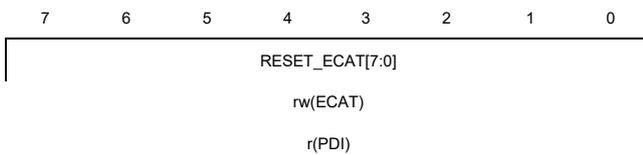
#### 9.4.15. ESC 复位 ECAT 寄存器 (ESC\_RESET\_ECAT)

地址偏移: 0x0040

复位值: 0x0

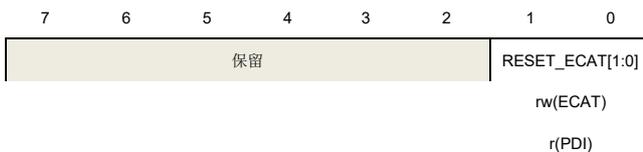
该寄存器可以按字节 (8位) 访问。

写:



位/位域	名称	描述
7:0	RESET_ECAT[7:0]	通过 3 个连续的命令向该寄存器中写入 52h (“R”)、45h (“E”) 和 53h (“S”) 之后, 将触发复位。

读:



位/位域	名称	描述
7:2	保留	必须保持复位值。
1:0	RESET_ECAT[1:0]	复位步骤进度 00: 初始化/复位状态 01: 在写入 52h (“R”) 后, 之前的状态是 00 10: 在写入 45h (“E”) 后, 之前的状态是 01 11: 在写入 45h (“S”) 后, 之前的状态是 10

这个值不能被检查到, 因为ESC在达到这个状态时进入复位, 导致状态00。

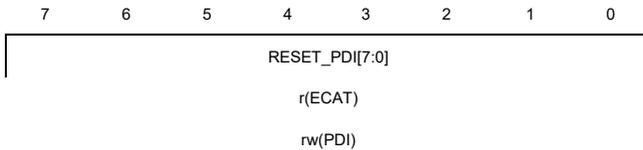
### 9.4.16. ESC 复位 PDI 寄存器 (ESC\_RESET\_PDI)

地址偏移: 0x0041

复位值: 0x0

该寄存器可以按字节 (8位) 访问。

写:



位/位域	名称	描述
7:0	RESET_PDI[7:0]	通过 3 个连续的命令向该寄存器中写入 52h (“R”)、45h (“E”) 和 53h (“S”) 之后，将触发复位。

读:



位/位域	名称	描述
7:2	保留	必须保持复位值。
1:0	RESET_PDI[1:0]	复位步骤进度 00: 初始化/复位状态 01: 在写入 52h (“R”) 后，之前的状态是 00 10: 在写入 45h (“E”) 后，之前的状态是 01 11: 在写入 45h (“S”) 后，之前的状态是 10

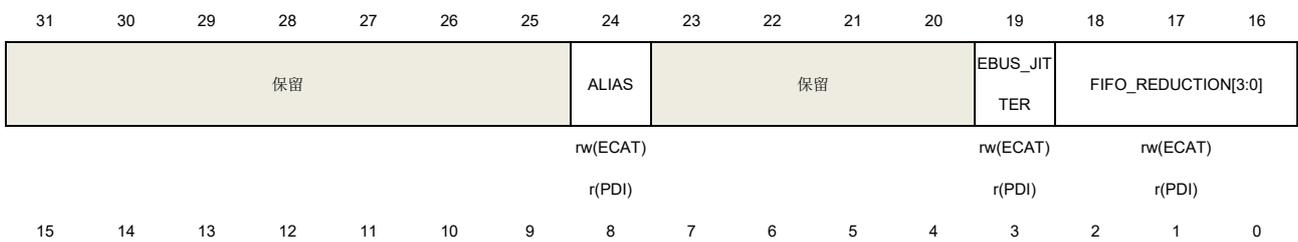
这个值不能被检查到，因为ESC在达到这个状态时进入复位，导致状态00。

### 9.4.17. ESC DL 控制寄存器 (ESC\_DL\_CONTROL)

地址偏移: 0x0100

复位值: 0x7C001

该寄存器可以按半字 (16位) 访问。



LOOP_P3[1:0]	LOOP_P2[1:0]	LOOP_P1[1:0]	LOOP_P0[1:0]	保留	USE	FRAME
rw(ECAT)	rw(ECAT)	rw(ECAT)	rw(ECAT)		rw(ECAT)	rw(ECAT)
r(PDI)	r(PDI)	r(PDI)	r(PDI)		r(PDI)	r(PDI)

位/位域	名称	描述
31:25	保留	必须保持复位值。
24	ALIAS	站别名 0: 忽略站别名 1: 别名可用于所有已配置地址命令类型（FPRD和FPWR等）
23:20	保留	必须保持复位值。
19	EBUS_JITTER	EBUS 低抖动 0: 正常抖动 1: 较低抖动
18:16	FIFO_REDUCTION[3:0]	接收 FIFO 大小/接收延时缩短 (ESC 延迟启动转发, 直至 FIFO 至少半满) Value: EBUS:                    MII: 0:    -50 ns                    -40 ns (-80 ns) 1:    -40 ns                    -40 ns (-80 ns) 2:    -30 ns                    -40 ns 3:    -20 ns                    -40 ns 4:    -10 ns                    无变化 5:    无变化                    无变化 6:    无变化                    无变化 7:    默认值                    默认值 注意: EEPROM值仅在上电或复位后的第一次EEPROM负载时使用
15:14	LOOP_P3[1:0]	环路端口 3 00: 自动 01: 自动关闭 10: 开启 11: 关闭
12:13	LOOP_P2[1:0]	环路端口 2 00: 自动 01: 自动关闭 10: 开启 11: 关闭
11:10	LOOP_P1[1:0]	环路端口 1 00: 自动 01: 自动关闭 10: 开启 11: 关闭

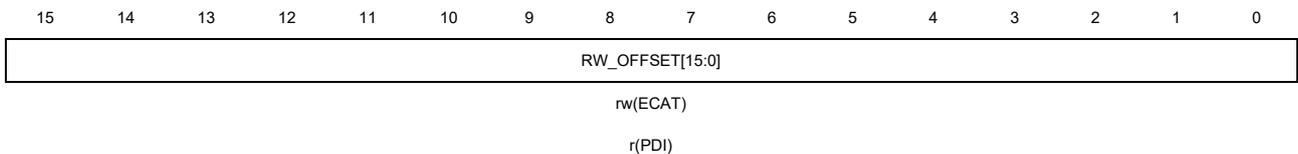
9:8	LOOP_P0[1:0]	<p>环路端口 0</p> <p>00: 自动</p> <p>01: 自动关闭</p> <p>10: 开启</p> <p>11: 关闭</p> <p>注意:环路打开意味着在这个端口上发送/接收是启用的, 环路关闭意味着发送/接收是禁用的, 帧被转发到下一个内部打开的端口。</p> <p><b>Auto:</b> 环路在断开时关闭, 在断开时打开。</p> <p><b>Auto Close:</b> 环路在断开时关闭, 在断开后(或在关闭端口收到有效的以太网帧)再次写入 01 打开。</p> <p><b>Open:</b>无论链路状态如何环路打开。</p> <p><b>Closed:</b>无论链路状态如何环路关闭。</p>
7:2	保留	必须保持复位值。
1	USE	<p>临时使用寄存器 0x0100:0x0103[8:15]设置</p> <p>0: 永久使用</p> <p>1: 临时使用约1s, 然后恢复之前的设置</p>
0	FRAME	<p>转发规则</p> <p>0: 转发非 EtherCAT 帧:处理 EtherCAT 帧, 不处理不修改转发非 EtherCAT 帧。任何帧的源 MAC 地址都不会改变。</p> <p>1: 破坏非EtherCAT帧:处理EtherCAT帧, 破坏非EtherCAT帧。源MAC地址由处理单元为每帧更改(SOURCE_MAC[1]设置为1---本地管理地址)。</p>

#### 9.4.18. ESC 物理读写偏移寄存器 (ESC\_PHYSICAL\_OFFSET)

地址偏移: 0x0108

复位值: 0x0

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15:0	RW_OFFSET[15:0]	这个寄存器用于设备寻址模式(FPRW, APRW, BRW)下的读写命令。内部读地址直接从 EtherCAT 数据报报头的偏移地址字段中获取, 而内部写地址是通过在偏移地址字段中添加物理读/写偏移值来计算的。内部读地址=ADR, 内部写地址=ADR + R/W-偏移

#### 9.4.19. ESC DL 状态寄存器 (ESC\_DL\_STATUS)

地址偏移: 0x0110

复位值：0x0

该寄存器可以按半字（16位）访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL_STAT	保留	DL_STAT	DL_STAT	DL_STAT											
US15	US14	US13	US12	US11	US10	US9	US8	US7	US6	US5	US4		US2	US1	US0
r(ECAT)		r(ECAT)	r(ECAT)	r(ECAT)											
r(PDI)		r(PDI)	r(PDI)	r(PDI)											

位/位域	名称	描述
15	DL_STATUS15	端口 3 上的通信 0: 无稳定通信 1: 已建立通信
14	DL_STATUS14	环路端口 3 0: 开启 1: 关闭
13	DL_STATUS13	端口 2 上的通信 0: 无稳定通信 1: 已建立通信
12	DL_STATUS12	环路端口 2 0: 开启 1: 关闭
11	DL_STATUS11	端口 1 上的通信 0: 无稳定通信 1: 已建立通信
10	DL_STATUS10	环路端口 1 0: 开启 1: 关闭
9	DL_STATUS9	端口 0 上的通信 0: 无稳定通信 1: 已建立通信
8	DL_STATUS8	环路端口 0 0: 开启 1: 关闭
7	DL_STATUS7	端口 3 上的物理链路 0: 无链路 1: 检测到链路
6	DL_STATUS6	端口 2 上的物理链路 0: 无链路 1: 检测到链路

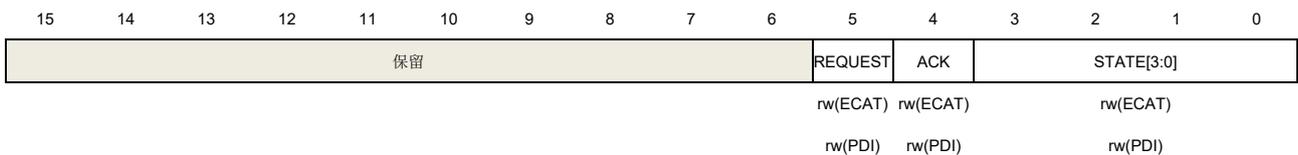
5	DL_STATUS5	端口 1 上的物理链路 0: 无链路 1: 检测到链路
4	DL_STATUS4	端口 0 上的物理链路 0: 无链路 1: 检测到链路
3	保留	必须保持复位值。
2	DL_STATUS2	增强型链路检测 0: 所有端口均禁止 1: 至少一个端口激活 注意: 仅在上电复位后第一次装载EEPROM时接受EEPROM值。
1	DL_STATUS1	PDI 看门狗状态 0: 看门狗超时 1: 看门狗重载
0	DL_STATUS0	PDI 工作/EEPROM 正确装载 0: EEPROM 未装载, PDI 未工作 (未访问过程数据 RAM) 1: EEPROM正确装载, PDI工作 (访问过程数据RAM)

#### 9.4.20. ESC AL 控制寄存器 (ESC\_AL\_CONTROL)

地址偏移: 0x0120

复位值: 0x1

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15:6	保留	必须保持复位值。
5	REQUEST	设备标识 0: 无请求 1: 设备标识请求
4	ACK	错误指示应答 0: AL 状态寄存器中无错误指示应答 1: AL状态寄存器中有错误指示应答
3:0	STATE[3:0]	启动器件状态机的状态转换 1h: 请求启动状态 2h: 请求预工作状态 3h: 请求自举状态

4h: 请求安全工作状态

8h: 请求工作状态

注意:PDI寄存器功能通过写命令确认被禁用:从PDI读取AL控制清除AL事件请求0x0220[0]。无法从PDI写入此寄存器。PDI寄存器功能通过写命令确认已启用:从PDI写AL控制清除AL事件请求0x0220[0]。从PDI写入这个寄存器是可以的;忽略写值(写0)。

**注意:** 如果器件仿真关闭(0x0141[0]=0), 则该寄存器的行为将类似于邮箱。ECAT写入该寄存器后, PDI必须读取该寄存器。否则, ECAT无法再次写入该寄存器。复位后, 该寄存器可通过ECAT写入。(关于邮箱功能, AL控制寄存器的低字节和高字节都触发读/写功能, 例如, 读取0x121足以使该寄存器能够再次被写入。)如果器件仿真开启, AL控制寄存器可始终被写入, 其内容复制到AL状态寄存器。

### 9.4.21. ESC AL 状态寄存器 (ESC\_AL\_STATUS)

地址偏移: 0x0130

复位值: 0x1

该寄存器可以按半字(16位)访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留										IDENTI	ERROR_I ND	AL_STATE[3:0]			
										r(ECAT)	r(ECAT)	r(ECAT)			
										rw(PDI)	rw(PDI)	rw(PDI)			

位/位域	名称	描述
15:6	保留	必须保持复位值。
5	IDENTI	设备标识 0: 设备标识无效 1: 设备标识装载
4	ERROR_IND	错误指示 0: 器件处于请求的状态或者标志由命令清零 1: 本地操作后, 器件未进入请求的状态或更改的状态
3:0	AL_STATE[3:0]	器件状态机的实际状态 1h: 启动状态 2h: 预工作状态 3h: 自举状态 4h: 安全工作状态 8h: 工作状态

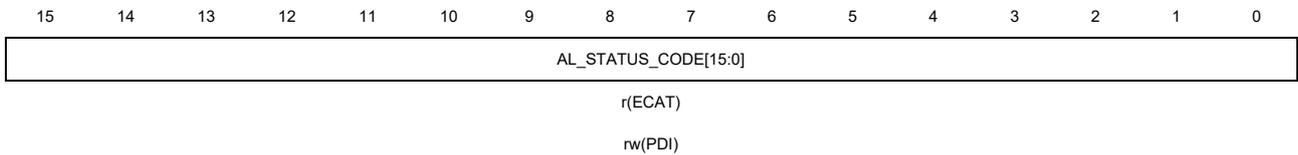
**注意:** 如果设备仿真关闭(0x0141[0]=0), AL状态寄存器只能从PDI写入, 否则AL状态寄存器将反映AL控制寄存器的值。避免从PDI读取AL状态寄存器。

### 9.4.22. ESC AL 状态代码寄存器 (ESC\_AL\_STATUS\_CODE)

地址偏移: 0x0134

复位值: 0x0

该寄存器可以按半字 (16位) 访问。



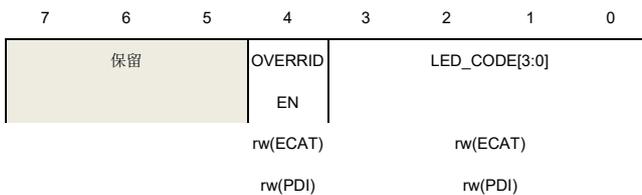
位/位域	名称	描述
15:0	AL_STATUS_CODE[15:0]	状态代码

### 9.4.23. ESC 运行 LED 覆盖寄存器 (ESC\_RUN\_LED)

地址偏移: 0x0138

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述														
7:5	保留	必须保持复位值。														
4	OVERRIDEN	运行改写 0: 禁止改写 1: 使能改写														
3:0	LED_CODE[3:0]	<table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">LED 代码:</td><td style="width: 50%;">AL 状态:</td></tr> <tr> <td>0x0: 熄灭</td><td>启动 1</td></tr> <tr> <td>0x1: 闪烁 1x</td><td></td></tr> <tr> <td>0x2-0xC: 闪烁 2x – 12x</td><td>安全工作 4</td></tr> <tr> <td>0xD: 快速闪烁</td><td>预工作 2</td></tr> <tr> <td>0xE: 非等时闪烁</td><td>自举 3</td></tr> <tr> <td>0xF: 点亮</td><td>工作 8</td></tr> </table>	LED 代码:	AL 状态:	0x0: 熄灭	启动 1	0x1: 闪烁 1x		0x2-0xC: 闪烁 2x – 12x	安全工作 4	0xD: 快速闪烁	预工作 2	0xE: 非等时闪烁	自举 3	0xF: 点亮	工作 8
LED 代码:	AL 状态:															
0x0: 熄灭	启动 1															
0x1: 闪烁 1x																
0x2-0xC: 闪烁 2x – 12x	安全工作 4															
0xD: 快速闪烁	预工作 2															
0xE: 非等时闪烁	自举 3															
0xF: 点亮	工作 8															

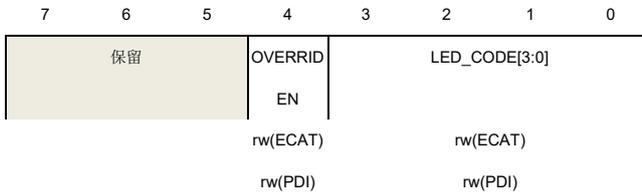
**注意:** 改变AL状态寄存器(0x0130)的有效值将禁用RUN LED覆盖(0x0138[4]=0)。在这个寄存器中读取的值总是反映当前的LED输出。

#### 9.4.24. ESC 错误 LED 覆盖寄存器 (ESC\_ERR\_LED)

地址偏移: 0x0139

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:5	保留	必须保持复位值。
4	OVERRIDEN	运行改写 0: 禁止改写 1: 使能改写
3:0	LED_CODE[3:0]	LED 代码: 0x0: 关闭 0x1-0xC: 闪烁 1x – 12x 0xD: 快速闪烁 0xE: 非等时闪烁 0xF: 点亮

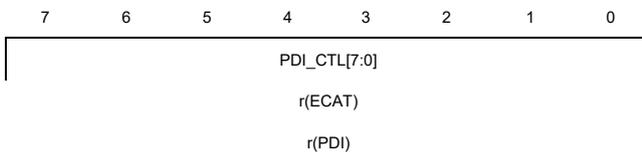
**注意:** 新的错误条件将禁用ERR LED覆盖(0x0139[4]=0)。在这个寄存器中读取的值总是反映当前的LED输出。

#### 9.4.25. ESC PDI 控制寄存器 (ESC\_PDI\_CONTROL)

地址偏移量: 0x0140

复位值: 取决于配置

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:0	PDI_CTL[7:0]	过程数据接口 04h: 数字 I/O 80h: 片上总线 其他: 保留

### 9.4.26. ESC 配置寄存器 (ESC\_CONFIG)

地址偏移: 0x0141

复位值: 取决于配置

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
EH_LINK_P3	EH_LINK_P2	EH_LINK_P1	EH_LINK_P0	CLK_LATCH_EN	CLK_SYNC_EN	EH_LINK_ALL	CTL_AL_STATUS
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)

位/位域	名称	描述
7	EH_LINK_P3	增强型链路端口 3 (默认为 1, EEPROM 加载后字为 0) 0: 禁止 (如果 bit 1 = 0) 1: 使能
6	EH_LINK_P2	增强型链路端口 2 (默认为 1, EEPROM 加载后字为 0) 0: 禁止 (如果 bit 1 = 0) 1: 使能
5	EH_LINK_P1	增强型链路端口 2 (默认为 1, EEPROM 加载后字为 0) 0: 禁止 (如果 bit 1 = 0) 1: 使能
4	EH_LINK_P0	增强型链路端口 0 (默认为 1, EEPROM 加载后字为 0) 0: 禁止 (如果 bit 1 = 0) 1: 使能
3	CLK_LATCH_EN	分布式时钟锁存输入单元 0: 禁止 (节能) 1: 使能
2	CLK_SYNC_EN	分布式时钟 SYNC 输出单元 0: 禁止 (节能) 1: 使能
1	EH_LINK_ALL	0: 禁止 (如果 bit[7:4] = 0) (默认为 1, EEPROM 加载后字为 0) 1: 使能所有端口 (覆写 bit[7:4])
0	CTL_AL_STATUS	器件仿真 (控制 AL 状态寄存器) 0: AL 状态寄存器必须由 PDI 设置 1: AL 状态寄存器设置为写入 AL 控制寄存器的值  <b>注意:</b> 对于数字 I/O 模式, 编程的值应为 1; 对于具有主机控制器的应用, 编程的值应为 0。

**注意:** Bits 1、4、5、6和7的EEPROM值只在上电或复位后的第一次EEPROM加载时传输到这

个寄存器中。

### 9.4.27. ESC PDI 信息寄存器 (ESC\_PDI\_INFM)

地址偏移: 0x014E

复位值: 取决于配置

该寄存器可以按半字 (16位) 访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留												CONFIGURE_EN	ACTIVE_EN	LOAD_EN	PDI_WREN
												r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
												r(PDI)	r(PDI)	r(PDI)	r(PDI)

位/位域	名称	描述
15:4	保留	必须保持复位值。
3	CONFIGURE_EN	PDI 配置无效(默认为 0): 0: PDI 配置 ok 1: PDI配置无效
2	ACTIVE_EN	PDI 激活(默认为 0): 0: PDI 未激活 1: PDI激活
1	LOAD_EN	ESC 配置区从 EEPROM 加载(默认 0): 0: 不加载 1: 加载
0	PDI_WREN	PDI 功能写确认: 0: 关闭 1: 使能 注意: 复位值取决于配置

### 9.4.28. ESC PDI 配置寄存器 (ESC\_PDI\_CONFIG)

地址偏移: 0x0150

复位值: 取决于配置

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
OUT_DATA[1:0]	IN_DATA[1:0]	WATCHDOG	MODE	OUTVALI	POLARIT		
		OG		D_MOD	Y		
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)

位/位域	名称	描述
------	----	----

7:6	OUT_DATA[1:0]	输出数据采样选择 00: 帧结束 01: 保留 10: DC SYNC0 事件 11: DC SYNC1 事件 如果0x0150[1]=1, 则在进程数据看门狗触发事件时更新输出数据(忽略0x0150[7:6])
5:4	IN_DATA[1:0]	输入数据采样选择 00: 帧结束 01: LATCH_IN 的上升沿 10: DC SYNC0 事件 11: DC SYNC1事件
3	WATCHDOG	看门狗行为(默认0): 0: 输出在看门狗超时后立即复位 1: 输出在看门狗超时后的下一个输出事件时复位
2	MODE	单向/双向模式(默认 0): 0: 单向模式: 单独配置引脚的输入/输出方向 1: 双向模式: 所有I/O引脚均是双向的, 忽略方向配置
1	OUTVALID_MOD	OUTVALID 模式(默认 0): 0: 输出事件信号传输 1: OUTVALID上的过程数据看门狗触发信号(WD_TRIG)传输。如果看门狗被触发, 则会更新输出数据。改写0x0150[7:6]。
0	POLARITY	OUTVALID 极性(默认 0): 0: 高电平有效 1: 低电平有效

**注意:** 所有bit都可以通过EEPROM配置。

#### 9.4.29. ESC 同步/锁存器配置寄存器 (ESC\_SL\_CONFIG)

地址偏移量: 0x0151

复位值: 取决于配置

该寄存器可以按字节(8位)访问。

7	6	5	4	3	2	1	0
SYNC1_M	SL1_CON	SYNC1_POLARITY[0:	SYNC0_M	SL0_CON	SYNC1_POLARITY[1:		
AP	FIG	1]	AP	FIG	0]		
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	
r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	

位/位域	名称	描述
7	SYNC1_MAP	SYNC1 映射到 AL 事件请求寄存器 0x0220[3]: 0: 禁止

		1: 使能
6	SL1_CONFIG	SYNC1/LATCH1 配置(默认 1) 0: LATCH1 输入 1: SYNC1 输出 注意: ESC具有SYNC[1:0]输出和LATCH[1:0]输入, 独立于此配置
5:4	SYNC1_POLARITY[0:1]	SYNC1 输出驱动器/极性(默认 10) 00: 推挽式 (低电平有效) 01: 漏极开路 (低电平有效) 10: 推挽式 (高电平有效) 11: 源极开路 (高电平有效)
3	SYNC0_MAP	SYNC0 映射到 AL 事件请求寄存器 0x0220[2]: 0: 禁止 1: 使能
2	SL0_CONFIG	SYNC0/LATCH0 配置(默认 1) 0: LATCH0 输入 1: SYNC0 输出 注意: ESC具有SYNC[1:0]输出和LATCH[1:0]输入, 独立于此配置
1:0	SYNC0_POLARITY[1:0]	SYNC0 输出驱动器/极性(默认 10) 00: 推挽式 (低电平有效) 01: 漏极开路 (低电平有效) 10: 推挽式 (高电平有效) 11: 源极开路 (高电平有效)

### 9.4.30. ESC PDI 扩展配置寄存器 (ESC\_DEXT\_CFG)

地址偏移: 0x0152

复位值: 取决于配置

该寄存器可以按字节 (8位) 访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR15	DIR8	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0						
r(ECAT)															
r(PDI)															

位/位域	名称	描述
15	DIR15	I/O 方向[31:30]成对配置为输入或输出: 参考DIR0描述
14	DIR14	I/O 方向[29:28]成对配置为输入或输出: 参考DIR0描述
13	DIR13	I/O 方向[27:26]成对配置为输入或输出:

		参考DIR0描述
12	DIR12	I/O 方向[25:24]成对配置为输入或输出： 参考DIR0描述
11	DIR11	I/O 方向[23:22]成对配置为输入或输出： 参考DIR0描述
10	DIR10	I/O 方向[21:20]成对配置为输入或输出： 参考DIR0描述
9	DIR9	I/O 方向[19:18]成对配置为输入或输出： 参考DIR0描述
8	DIR8	I/O 方向[17:16]成对配置为输入或输出： 参考DIR0描述
7	DIR7	I/O 方向[15:14]成对配置为输入或输出： 参考DIR0描述
6	DIR6	I/O 方向[13:12]成对配置为输入或输出： 参考DIR0描述
5	DIR5	I/O 方向[11:10]成对配置为输入或输出： 参考DIR0描述
4	DIR4	I/O 方向[9:8]成对配置为输入或输出： 参考DIR0描述
3	DIR3	I/O 方向[7:6]成对配置为输入或输出： 参考DIR0描述
2	DIR2	I/O 方向[5:4]成对配置为输入或输出： 参考DIR0描述
1	DIR1	I/O 方向[3:2]成对配置为输入或输出： 参考DIR0描述
0	DIR0	I/O 方向[1:0]成对配置为输入或输出： 0: 输入 1: 输出 备注:在双向模式下保留，设置为0。预留不可用的I/O配置位，将EEPROM值设置为0。

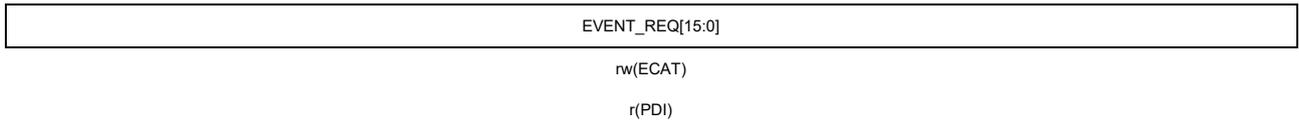
### 9.4.31. ESC 事件屏蔽寄存器 (ESC\_EVENT\_MASK)

地址偏移: 0x0200

复位值: 0x0

该寄存器可以按半字 (16位) 访问。

15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0



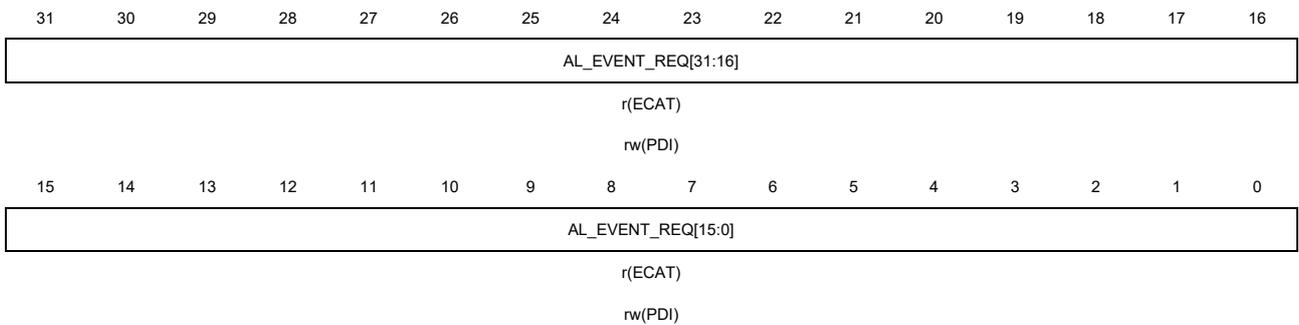
位/位域	名称	描述
15:0	EVENT_REQ[15:0]	ECAT 事件请求寄存器事件的 ECAT 事件屏蔽，用于映射到 EtherCAT 帧的 ECAT 事件字段。 0: 不映射相应的 ECAT 事件请求寄存器位 1: 映射相应的 ECAT 事件请求寄存器位

### 9.4.32. ESC PDI AL 事件寄存器 (ESC\_PDI\_AL\_EVENT)

地址偏移: 0x0204

复位值: 0x00FFFF0F

该寄存器可以按字节 (8位)、半字 (16位) 和字 (32位) 访问。



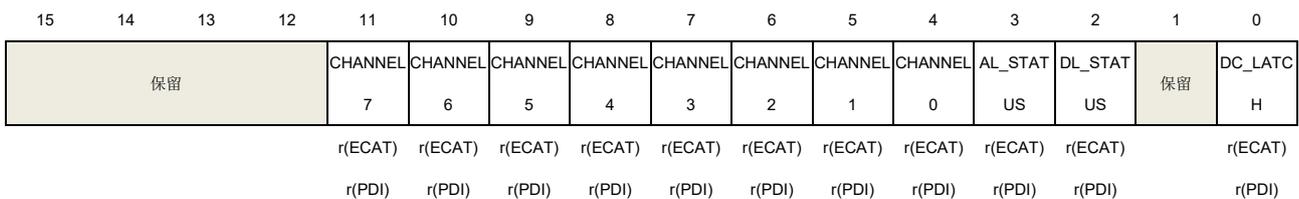
位/位域	名称	描述
31:0	AL_EVENT_REQ[31:0]	AL 事件请求寄存器事件的 AL 事件屏蔽，用于映射到 PDI IRQ 信号。 0: 不映射相应的 AL 事件请求寄存器位 1: 映射相应的 AL 事件请求寄存器位

### 9.4.33. ESC 事件请求寄存器 (ESC\_EVENT\_RQST)

地址偏移: 0x0210

复位值: 0x0

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
------	----	----

15:12	保留	必须保持复位值。
11	CHANNEL7	SyncManager 状态镜像 0: 无同步通道 7 事件 1: 同步通道7事件待处理
10	CHANNEL6	SyncManager 状态镜像 0: 无同步通道 6 事件 1: 同步通道6事件待处理
9	CHANNEL5	SyncManager 状态镜像 0: 无同步通道 5 事件 1: 同步通道5事件待处理
8	CHANNEL4	SyncManager 状态镜像 0: 无同步通道 4 事件 1: 同步通道4事件待处理
7	CHANNEL3	SyncManager 状态镜像 0: 无同步通道 3 事件 1: 同步通道3事件待处理
6	CHANNEL2	SyncManager 状态镜像 0: 无同步通道 2 事件 1: 同步通道2事件待处理
5	CHANNEL1	SyncManager 状态镜像 0: 无同步通道 1 事件 1: 同步通道1事件待处理
4	CHANNEL0	SyncManager 状态镜像 0: 无同步通道 0 事件 1: 同步通道0事件待处理
3	AL_STATUS	AL 状态事件: 0: AL 状态无变化 1: AL 状态变化 (通过从ECAT读出AL状态0x0130:0x0131来清除位)
2	DL_STATUS	DL 状态事件: 0: DL 状态无变化 1: DL 状态变化 (通过从ECAT读出DL状态0x010:0x0111来清除位)
1	保留	必须保持复位值。
0	DC_LATCH	DC 锁存事件: 0: DC 锁存输入无变化 1: 至少有一个 DC 锁存输入变化 (该位的清零方式是通过ECAT读取ECAT控制的锁存单元的DC锁存事件的时间, 以

使0x09AE:0x09AF指示无事件)

#### 9.4.34. ESC AL 事件请求寄存器 (ESC\_AL\_EVENT\_RQST)

地址偏移: 0x0220

复位值: 0x0

该寄存器可以按字(32位)访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								SYNC15_I	SYNC14_I	SYNC13_I	SYNC12_I	SYNC11_I	SYNC10_I	SYNC9_IN	SYNC8_IN
								NT	NT	NT	NT	NT	NT	T	T
								r(ECAT)							
								r(PDI)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC7_I	SYNC6_I	SYNC5_IN	SYNC4_I	SYNC3_I	SYNC2_IN	SYNC1_I	SYNC0_I	保留	PDATA	EEPROM	SYNC_CH	DC_SYN	DC_SYNC	DC_LAT	AL_CONT
NT	NT	T	NT	NT	T	NT	NT				ANGE	C1	0	CH	ROL
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)		r(ECAT)						
r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)		r(PDI)						

位/位域	名称	描述
31:24	保留	必须保持复位值。
23	SYNC15_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 15 中断 1: SyncManager 15 中断待处理
22	SYNC14_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 14 中断 1: SyncManager 14 中断待处理
21	SYNC13_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 13 中断 1: SyncManager 13 中断待处理
20	SYNC12_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 12 中断 1: SyncManager 12 中断待处理
19	SYNC11_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 11 中断 1: SyncManager 11 中断待处理
18	SYNC10_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 10 中断 1: SyncManager 10 中断待处理
17	SYNC9_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 9 中断

		1: SyncManager 9中断待处理
16	SYNC8_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 8 中断 1: SyncManager 8中断待处理
15	SYNC7_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 7 中断 1: SyncManager 7中断待处理
14	SYNC6_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 6 中断 1: SyncManager 6中断待处理
13	SYNC5_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 5 中断 1: SyncManager 5中断待处理
12	SYNC4_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 4 中断 1: SyncManager 4中断待处理
11	SYNC3_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 3 中断 1: SyncManager 3中断待处理
10	SYNC2_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 2 中断 1: SyncManager 2中断待处理
9	SYNC1_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 1 中断 1: SyncManager 1中断待处理
8	SYNC0_INT	SyncManager 中断 (SyncManager 寄存器偏移 0x5, bit 0 或 bit 1) 0: 无 SyncManager 0 中断 1: SyncManager 0中断待处理
7	保留	必须保持复位值。
6	PDATA	看门狗过程数据: 0: 未超时 1: 已超时 (该位的清零方式是通过PDI读取看门狗状态过程数据0x0440。)
5	EEPROM	EEPROM 仿真 0: 无命令待处理 1: EEPROM 命令待处理 (该位的清零方式是通过PDI应答EEPROM控制/状态寄存器0x0502:0x0503[10:8]中的命令。)

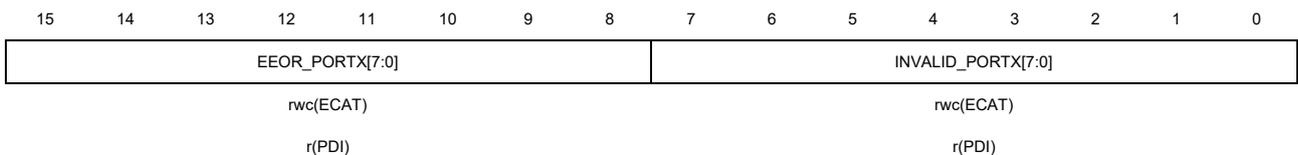
4	SYNC_CHANGE	SyncManager 激活寄存器变化 (SyncManager 激活寄存器) 0: 任何 SyncManager 中均无变化 1: 至少一个 SyncManager 发生变化 (该位的清零方式是通过PDI读取相应的SyncManager激活寄存器0x0806。)
3	DC_SYNC1	DC SYNC1 的状态 (如果 0x0151 寄存器的 bit 7 = 1) (该位的清零方式是通过PDI读取SYNC1状态0x098F, 仅在确认模式下使用。)
2	DC_SYNC0	DC SYNC0 的状态 (如果 0x0151 寄存器的 bit 3 = 1) (该位的清零方式是通过PDI读取SYNC0状态0x098E, 仅在确认模式下使用。)
1	DC_LATCH	DC 锁存事件 0: DC 锁存输入无变化 1: 至少有一个 DC 锁存输入变化 (该位的清零方式是通过PDI读取PDI控制的锁存单元的DC锁存事件的时间, 以使 0x09AE:0x09AF指示无事件。)
0	AL_CONTROL	AL 控制事件 0: AL 控制寄存器无变化 1: 已写入 AL 控制寄存器 (AL 控制事件仅在 PDI 仿真关闭时生成(ESC 配置 0 寄存器 0x0141[0]=0)) (该位的清零方式是通过PDI读取AL控制寄存器0x0120:0x0121)

#### 9.4.35. PortX 接收错误计数器寄存器 (RX\_PORTX\_ERROR) (X = 0,1,2,3)

地址偏移:  $0x0300 + X * 2$

复位值: 0x0

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15:8	EEOR_PORTX[7:0]	端口 X 接收错误计数器 当达到FFh时, 计数停止。端口X接收错误计数
7:0	INVALID_PORTX[7:0]	端口 X 无效帧计数器 当达到FFh时, 计数停止。端口X无效帧计数

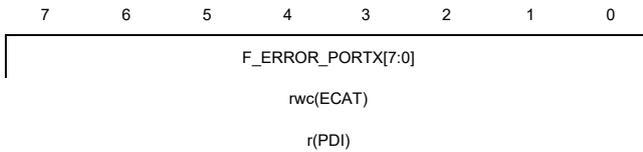
**注意:** 如果写入任意一个的RX错误计数器0x0300-0x030B(最好是0x0300), 则清除错误计数器 0x0300- 0x030B, 忽略写值(写0), 只有当端口环路打开时才会计算错误。

#### 9.4.36. Port X 转发接收错误计数器寄存器 (FRX\_PORTX\_ERROR) (X = 0,1,2,3)

地址偏移:  $0x0308 + X$

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:0	F_ERROR_PORTX[7:0]	当达到 FFh 时，计数停止。端口 X 转发的接收错误计数器

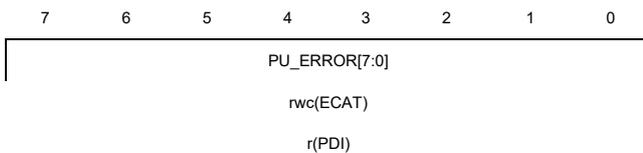
**注意：**如果写入任意一个的RX错误计数器0x0300-0x030B(最好是0x0300)，则清除错误计数器0x0300- 0x030B，忽略写值(写0)，只有当端口环路打开时才会计算错误。

#### 9.4.37. ESC 处理单元错误计数器寄存器（ESC\_PU\_ERROR）

地址偏移：0x030C

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:0	PU_ERROR[7:0]	ECAT处理单元错误计数器(当到达0xFF时停止计数)。计数通过处理单元的帧的错误。

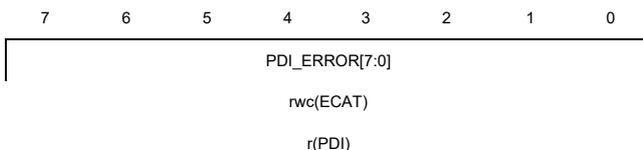
**注意：**如果写入错误计数器0x030C，则错误计数器0x030C会被清除。写入的值将被忽略（写入0）。

#### 9.4.38. ESC PDI 错误计数器寄存器（ESC\_PDI\_ERROR）

地址偏移：0x030D

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:0	PDI_ERROR[7:0]	PDI 错误计数器(当到达 0xFF 时停止计数)。计数 PDI 访问是否有接口错误。

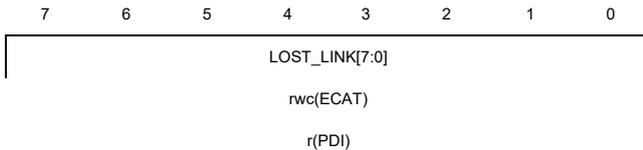
**注意：**如果写入错误计数器0x030D，则清除错误计数器0x030D和错误代码0x030E:0x030F。写入值将被忽略（写入0）。

#### 9.4.39. ESC PortX 丢失链路计数器寄存器 (ESC\_PORTX\_LOST\_LINK) (X = 0,1,2,3)

地址偏移：0x0310 + X

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:0	LOST_LINK[7:0]	端口 X 丢失链路计数器（当达到 FFh 时，计数停止。）仅当端口环路处于自动模式或自动关闭模式时，该计数器才会计数。

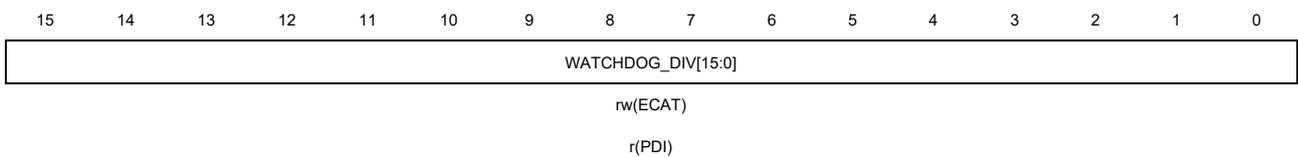
**注意：**如果写入其中一个已实现的丢失链接计数器0x0310-0x0313（最好是0x0310），则0x0310-0x0313的丢失链接计数器将被清除。写入值将被忽略（写入0）。

#### 9.4.40. ESC 看门狗分频器寄存器 (ESC\_WTG\_DIVIDER)

地址偏移：0x0400

复位值：0x09C2

该寄存器可以按半字（16位）访问。



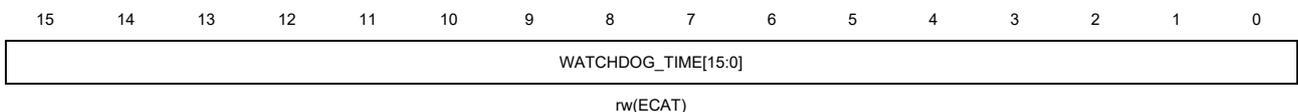
位/位域	名称	描述
15:0	WATCHDOG_DIV[15:0]	看门狗分频器：25 MHz 计时周期数（减2），表示基本看门狗递增次数。（默认值为 100 μs = 2498）

#### 9.4.41. ESC 看门狗定时器 PDI 寄存器 (ESC\_WTG\_TIME)

地址偏移：0x0410

复位值：0x03E8

该寄存器可以按半字（16位）访问。



r(PDI)

位/位域	名称	描述
15:0	WATCHDOG_TIME[15:0]	看门狗时间 PDI: 基本看门狗递增次数。(100 $\mu$ s 看门狗分频器默认值对应于 100 ms 看门狗。)

**注意:** 如果看门狗时间设置为0x0000, 则看门狗将被禁用。每次PDI访问后, 看门狗都会重新开始计数。

#### 9.4.42. ESC 看门狗定时器过程数据寄存器 (ESC\_WTG\_TPD)

地址偏移: 0x0420

复位值: 0x03E8

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15:0	WATCHDOG_TPD[15:0]	看门狗时间过程数据: 基本看门狗递增次数。(100 $\mu$ s 看门狗分频器默认值对应于 100 ms 看门狗。)

#### 9.4.43. ESC 看门狗状态过程数据寄存器 (ESC\_WTG\_STATUS)

地址偏移: 0x0440

复位值: 0x0

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15:1	保留	必须保持复位值。
0	WATCHDOG_STA	看门狗过程数据的状态 (由 SyncManager 触发) 0: 看门狗过程数据超时 1: 看门狗过程数据激活或禁止 <b>注意:</b> PDI 寄存器功能通过写命令确认被禁用:从 PDI 读取该寄存器清除 AL 事件请求 0x0220[6]。无法从 PDI 写入此寄存器。PDI 寄存器功能通过写命令确认已启用:从 PDI 写入该寄存器清除 AL 事件请求 0x0220[6]。从 PDI 写入这个寄存器是可能的;

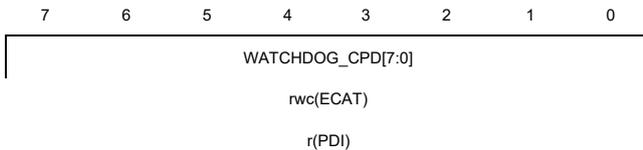
忽略写值(写 0)。

#### 9.4.44. ESC 看门狗计数器过程数据寄存器 (ESC\_WTG\_CTR)

地址偏移: 0x0442

复位值: 0x0

该寄存器可以按字节(8位)访问。



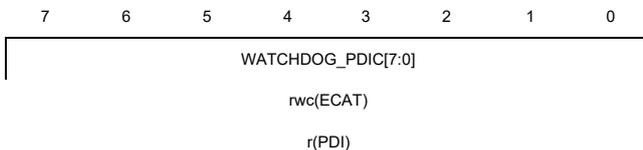
位/位域	名称	描述
7:0	WATCHDOG_CPD[7:0]	看门狗计数器过程数据(当达到 FFh 时, 计数停止)。计数以确定过程数据看门狗是否超时。

#### 9.4.45. ESC 看门狗计数器 PDI 寄存器 (ESC\_WTG\_CTR\_PDI)

地址偏移: 0x0443

复位值: 0x0

该寄存器可以按字节(8位)访问。



位/位域	名称	描述
7:0	WATCHDOG_PDIC[7:0]	看门狗 PDI 计数器(当达到 FFh 时, 计数停止)。计数以确定 PDI 看门狗是否超时。

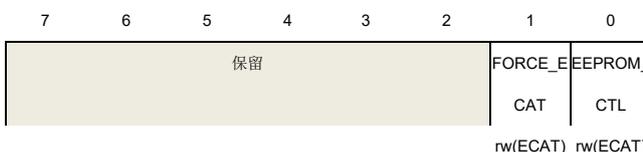
**注意:** 如果写入了任何一个看门狗计数器0x0442-0x0443, 那么0x0442-0x0443看门狗计数器都会被清零。写入值将被忽略(写入0)。

#### 9.4.46. ESC EEPROM 配置寄存器 (ESC\_EEPROM\_CONFIG)

地址偏移: 0x0500

复位值: 0x0

该寄存器可以按字节(8位)访问。



r(PDI) r(PDI)

位/位域	名称	描述
7:2	保留	必须保持复位值。
1	FORCE_ECAT	强制 ECAT 访问 0: 不改变 0x0501 bit 0 1: 复位 0x0501bit 0
0	EEPROM_CTL	PDI EEPROM 控制 0: 无 1: 有 (PDI 具有 EEPROM 控制)

#### 9.4.47. ESC EEPROM PDI 访问寄存器 (ESC\_EEPROM\_ACCESS)

地址偏移: 0x0501

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:1	保留	必须保持复位值。
0	ACCESS_EEPROM	EEPROM 访问 0: PDI 释放 EEPROM 访问 1: PDI采用EEPROM访问(PDI具有EEPROM控制)

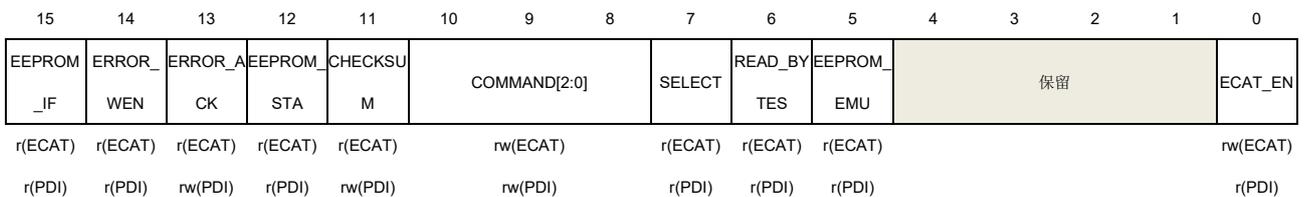
**注意:** 仅当(0x0500[0]=1 或 0x0501[0]=1)且0x0500[1]=0时, 才允许写入访问。

#### 9.4.48. ESC EEPROM 控制 / 状态寄存器 (ESC\_EEPROM\_CONTROL)

地址偏移: 0x0502

复位值: 取决于配置

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15	EEPROM_IF	繁忙（默认 0） 0: EEPROM 接口空闲 1: EEPROM 接口繁忙
14	ERROR_WEN	错误写使能（默认 0） 0: 无错误 1: 未进行写使能的写命令 注意：通过向命令寄存器bit[10:8]位写入“000”（或任何有效命令）将错误位清零。
13	ERROR_ACK	错误应答/命令（默认 0） 0: 无错误 1: 丢失 EEPROM 应答或无效命令 仅 EEPROM 仿真：如果发生临时故障，则 PDI 写为 1。 注意：通过向命令寄存器bit[10:8]位写入“000”（或任何有效命令）将错误位清零。
12	EEPROM_STA	EEPROM 装载状态（默认 0） 0: EEPROM 已装载，器件信息可用 1: EEPROM未装载，器件信息不可用（正在装载EEPROM或者EEPROM装载失败）
11	CHECKSUM	ESC 配置区域中的校验和错误 0: 校验和正确 1: 校验和错误 仅针对ESC的EEPROM仿真:如果重载命令发生CRC失败，PDI将写1。
10:8	COMMAND[2:0]	命令寄存器（默认 0） 写：启动命令 读：当前执行的命令 000: 无命令/EEPROM 空闲（清零错误位） 001: 读 010: 写 100: 重载 其他：保留/无效命令（不发出） 仅 EEPROM 仿真:执行后，PDI 写入命令值以指示操作准备就绪。 注意：执行命令后（EEPROM 繁忙结束），命令寄存器位将自清零。向命令寄存器位写入“000”还会将错误位bit 14:13清零。如果错误应答/命令待处理（bit13），则命令寄存器位bit10:8将被忽略。
7	SELECT	选择的 EEPROM 算法 0: 1 个地址字节（1 Kbit - 16 Kbit EEPROM） 1: 2个地址字节（32 Kbit - 4 Mbit EEPROM）
6	READ_BYTES	支持的 EEPROM 字节数（默认 0） 0: 4 字节 1: 8字节
5	EEPROM_EMU	EEPROM 仿真 0: 正常工作（使用 I2C 接口）

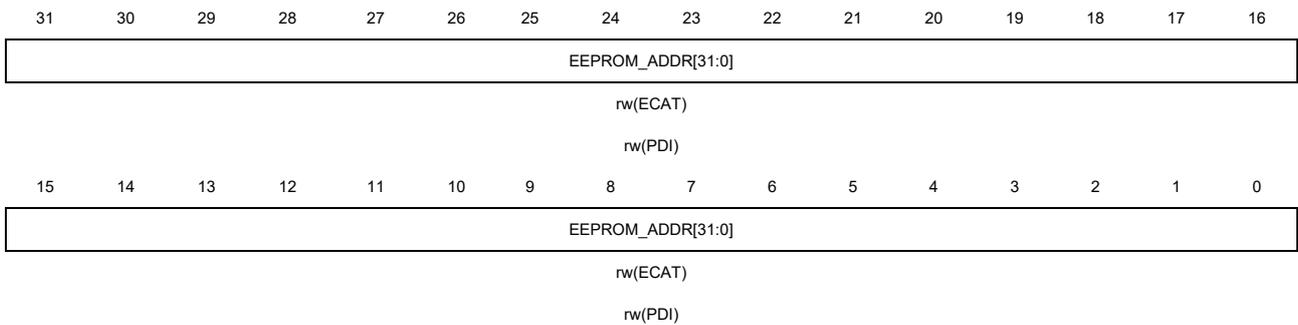
		1: PDI仿真EEPROM (未使用I2C)
4:1	保留	必须保持复位值。
0	ECAT_EN	ECAT 写使能 (默认 0) 0: 禁止写请求 1: 使能写请求 如果 PDI 有 EEPROM 控制, 这个位总是 1。 注意: 执行命令后 (EEPROM 繁忙结束), 命令寄存器位将自清零。向命令寄存器位写入“000”还会将错误位bit 14:13清零。如果错误应答/命令待处理 (bit13), 则命令寄存器位bit10:8将被忽略。

#### 9.4.49. ESC EEPROM 地址寄存器 (ESC\_EEPROM\_ADDR)

地址偏移: 0x0504

复位值: 0x0

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31:0	EEPROM_ADDR[31:0]	EEPROM 地址 Bit 0: 第一个字 (16 位) Bit 1: 第二个字 ..... 实际使用的 EEPROM 地址位: [9:0]: EEPROM 大小最大为 16 Kbit [17:0]: EEPROM 大小为 32 Kbit - 4 Mbit [31:0]: EEPROM仿真

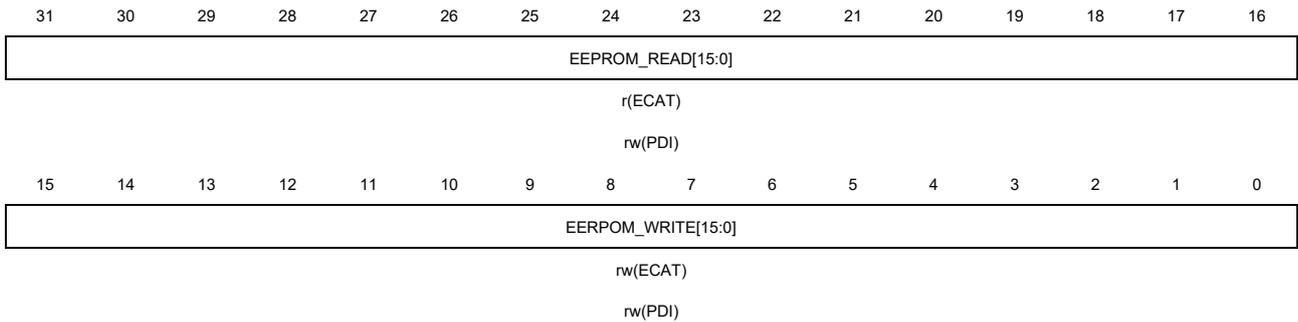
**注意:** 写入访问取决于EEPROM接口 (ECAT/PDI) 的分配。如果EEPROM接口忙碌 (0x0502[15]=1), 则写入访问将被阻止。

#### 9.4.50. ESC EEPROM 数据寄存器 (ESC\_EEPROM\_DATA)

地址偏移: 0x0508

复位值: 0x0

该寄存器可以按字 (32位) 访问。



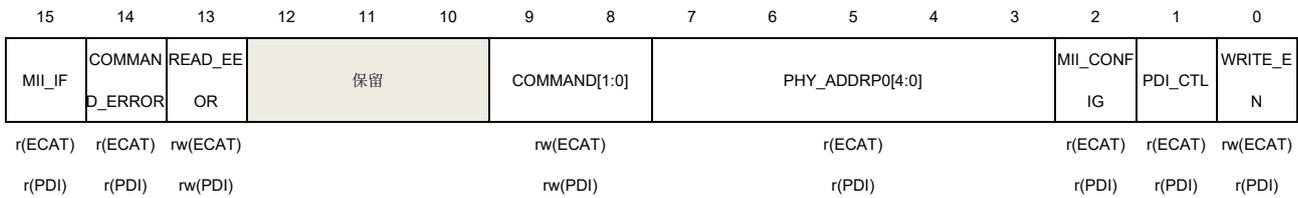
位/位域	名称	描述
31:16	EEPROM_READ[15:0]	EEPROM读数据（要从EEPROM读取的数据的高字节）
15:0	EERPOM_WRITE[15:0]	EEPROM 读/写数据 要从EEPROM读取的数据的低字节或者要写入EEPROM的数据

#### 9.4.51. ESC MII 管理控制/状态寄存器（ESC\_MII\_CTL）

地址偏移：0x0510

复位值：0x0

该寄存器可以按半字（16位）访问。



位/位域	名称	描述
15	MII_IF	繁忙 0: MI 控制状态机空闲 1: MI控制状态机工作
14	COMMAND_ERROR	命令错误 0: 最后一条命令成功 1: 无效命令或未进行写使能的写命令 通过无效命令或通过向命令寄存器bits[9:8]写入“00”清零。
13	READ_EEOR	读错误 0: 无读错误 1: 出现读错误（PHY 或寄存器不可用） 通过写入该寄存器清零。
12:10	保留	必须保持复位值
8:9	COMMAND[1:0]	命令寄存器 写: 启动命令 读: 当前执行的命令

		00: 无命令/MI 空闲 (清零错误位)
		01: 读
		10: 写
		11: 保留 (不发出)
		注意: 写使能bit 0在下一帧的SOF处自清除, 命令bits[10:8]在命令执行后自清除(繁忙结束)。将“00”写入命令寄存器也将清除错误bits[14:13]。命令执行完成后, 命令位将被清除。
7:3	PHY_ADDRP0[4:0]	端口 0 的 PHY 地址 (如果PHY地址是连续的, 则等于PHY地址偏移量)
2	MII_CONFIG	MII 链路检测和配置: 0: 禁用所有端口 1: 至少启用一个MII端口, 请参考PHY端口状态 (0x0518 ff.) 了解详细信息。
1	PDI_CTL	管理接口可以通过PDI (寄存器0x0516-0x0517) 进行控制: 0: 仅ECAT控制 1: PDI控制可能
0	WRITE_EN	写入使能: 0: 写入禁用 1: 写入启用  如果 PDI 控制了 MI (管理接口), 这个位总是 1。 <b>注意:</b> 写入使能位0在下一帧的SOF (帧开始) 时自动清零, 命令位[10:8]在命令执行后 (忙碌结束) 自动清零。向命令寄存器写入“00”也会清除错误位[14:13]。命令位在命令执行后被清除。

#### 9.4.52. ESC PHY 地址寄存器 (ESC\_PHY\_ADDR)

地址偏移: 0x0512

复位值: 0x0

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
PHY_ADDR	保留		TARGET_PHY[4:0]				
R							
rw(ECAT)			rw(ECAT)				
rw(PDI)			rw(PDI)				

位/位域	名称	描述
7	PHY_ADDR	目标PHY地址转换: 0: 启用 1: 禁用 详情请参考0x0512[4:0]和0x0510[7:3]。
6:5	保留	必须保持复位值。

- 4:0 TARGET\_PHY[4:0] 目标PHY地址转换  
 当0x0512[7]=0时：  
 0-3：目标PHY地址0-3用于访问端口0-3的PHY，当PHY地址正确配置时  
 4-31：当访问PHY时，端口0的配置PHY地址（PHY地址偏移）将被加到目标PHY地址值4-31上  
 当0x0512[7]=1时：  
 0-31：在不进行转换的情况下使用目标PHY地址访问PHY。

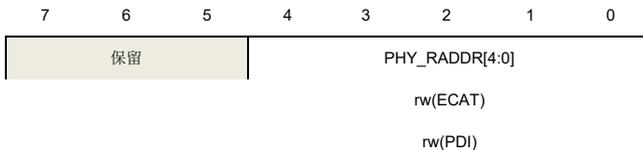
**注意：**写入访问取决于MII（ECAT/PDI）的分配。如果管理接口忙（0x0510[15]=1），则写入访问被阻止。

### 9.4.53. ESC PHY 寄存器地址寄存器（ESC\_PHY\_RADDR）

地址偏移：0x0513

复位值：0x0

该寄存器可以按字节（8位）访问。



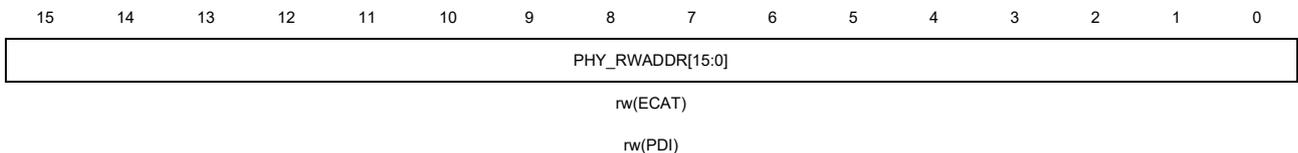
位/位域	名称	描述
7:5	保留	必须保持复位值。
4:0	PHY_RADDR[4:0]	要读取/写入的PHY寄存器的地址

### 9.4.54. ESC PHY 数据寄存器（ESC\_PHY\_DATA）

地址偏移：0x0514

复位值：0x0

该寄存器可以按半字（16位）访问。



位/位域	名称	描述
15:0	PHY_RWADDR[15:0]	PHY读/写数据

### 9.4.55. MII 管理 ECAT 访问状态寄存器（MII\_ECAT\_STATE）

地址偏移：0x0516

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:1	保留	必须保持复位值。
0	MII_ACCESS	访问MII管理： 0: ECAT使能PDI接管MII管理接口 1: ECAT声明对MII管理接口的独占访问权

**注意：**只有当0x0517[0]=0时，才允许写入访问。

#### 9.4.56. MII 管理 PDI 访问状态寄存器（MII\_PDI\_STATE）

地址偏移：0x0517

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:2	保留	必须保持复位值。
1	PDI_ACCESS	强制PDI访问状态： 0: 不改变0x0517[0]位 1: 将0x0517[0]位复位为0
0	MII_ACCESS	访问MII管理： 0: ECAT可以访问MII管理 1: PDI可以访问MII管理

#### 9.4.57. PHY Port X 状态寄存器（PHY\_PORTX\_STA）（X = 0,1,2,3）

地址偏移：0x0518 + X

复位值：0x0

该寄存器可以按字节（8位）访问。

7	6	5	4	3	2	1	0
保留	PHY_UPD	PARTNER	READ_ER	STATUS_	LINK_STA	PHYLINK	
	ATE	_ERROR	ROR	ERROR	TUS		
	rwc(ECAT)	r(ECAT)	r(ECAT)	rwc(ECAT)	r(ECAT)	r(ECAT)	
	rwc(PDI)	r(PDI)	r(PDI)	rwc(PDI)	r(PDI)	r(PDI)	

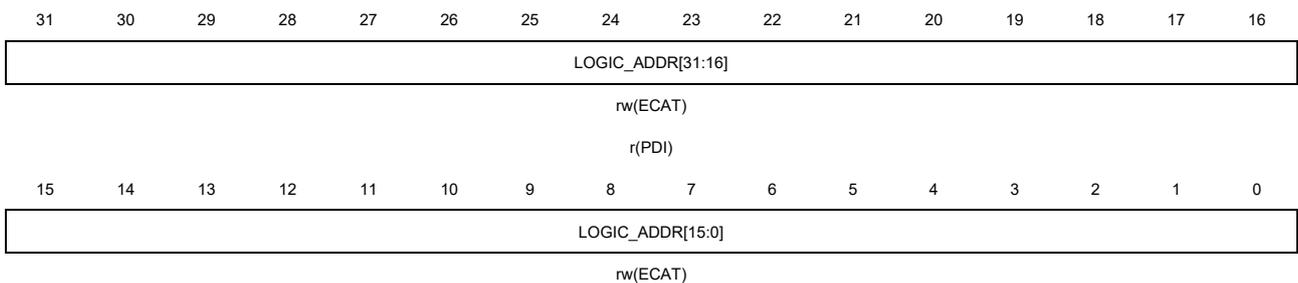
位/位域	名称	描述
7:6	保留	必须保持复位值。
5	PHY_UPDATE	PHY配置更新： 0: 无更新 1: PHY配置已更新 通过向任何一个PHY端口X状态寄存器写入任意值来清除。
4	PARTNER_ERROR	链路伙伴错误： 0: 未检测到错误 1: 链路伙伴错误
3	READ_ERROR	读取错误： 0: 未发生读取错误 1: 发生了读取错误 通过向任何一个PHY端口X状态寄存器写入任意值来清除。
2	STATUS_ERROR	链路状态错误： 0: 无错误 1: 链路错误，链路被禁止
1	LINK_STATUS	链路状态（100 Mbit/s，全双工，自动协商）： 0: 无链路 1: 检测到链路
0	PHYLINK	物理链路状态： 0: 无物理链路 1: 检测到物理链路

#### 9.4.58. FMMUX 逻辑起始地址寄存器 (FMMUX\_LOGIC\_ADDR) (X = 0...F)

地址偏移量:  $0x0600 + X * 0x10$

复位值: 0x0

该寄存器可以按字（32位）访问。



r(PDI)

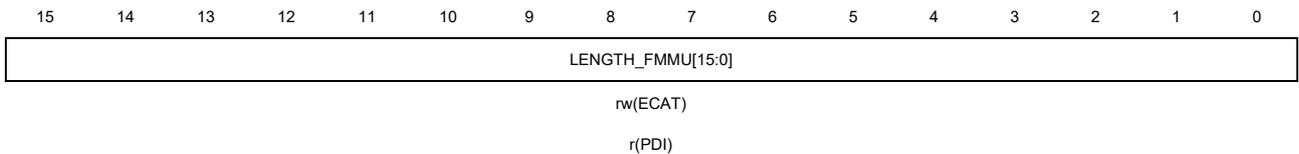
位/位域	名称	描述
31:0	LOGIC_ADDR[31:0]	EtherCAT地址空间内的逻辑起始地址。

#### 9.4.59. FMMUX 长度寄存器 (FMMUX\_LENGTH) (X = 0...F)

 地址偏移量:  $0x0604 + X * 0x10$ 

复位值: 0x0

该寄存器可以按半字 (16位) 访问。



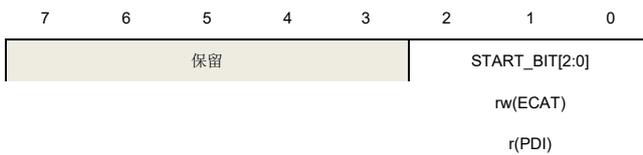
位/位域	名称	描述
15:0	LENGTH_FMMU[15:0]	从第一个逻辑FMMU字节到最后一个FMMU字节加1的偏移量 (例如, 如果使用了两个字节, 则此参数应包含2)

#### 9.4.60. FMMUX 逻辑起始位寄存器 (FMMUX\_STRA\_BIT) (X = 0...F)

 地址偏移:  $0x0606 + X * 0x10$ 

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:3	保留	必须保持复位值。
2:0	START_BIT[2:0]	需要映射的逻辑起始位 (位从最低有效位0开始计数到最高有效位7)

#### 9.4.61. FMMUX 逻辑停止位寄存器 (FMMUX\_STOP\_BIT) (X = 0...F)

 地址偏移:  $0x0607 + X * 0x10$ 

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



rw(ECAT)

r(PDI)

位/位域	名称	描述
7:3	保留	必须保持复位值。
2:0	STOP_BIT[2:0]	需要映射的最后一个逻辑位（位从最低有效位0开始计数到最高有效位7）

#### 9.4.62. FMMUX 物理起始地址寄存器（FMMUX\_ADRR）（X = 0...F）

地址偏移：0x0608 + X \* 0x10

复位值：0x0

该寄存器可以按半字（16位）访问。



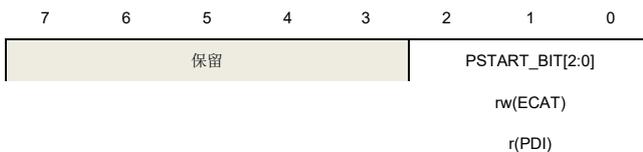
位/位域	名称	描述
15:0	START_ADDR[15:0]	物理起始地址（映射到逻辑起始地址）

#### 9.4.63. FMMUX 物理起始位寄存器（FMMUX\_PSBIT）（X = 0...F）

地址偏移：0x060A + X \* 0x10

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:3	保留	必须保持复位值。
2:0	PSTART_BIT[2:0]	物理起始位作为逻辑起始位映射的目标（位从最低有效位0开始计数到最高有效位7）

#### 9.4.64. FMMUX 类型寄存器（FMMUX\_TYPE）（X = 0...F）

地址偏移量：0x060B + X \* 0x10

复位值：0x0

该寄存器可以按字节（8位）访问。



保留	MAP_WRI	MAP_REA
	TE	D
	rw(ECAT) rw(ECAT)	
r(PDI)		r(PDI)

位/位域	名称	描述
7:2	保留	必须保持复位值。
1	MAP_WRITE	0: 忽略写访问的映射 1: 写访问时使用映射
0	MAP_READ	0: 忽略读访问的映射 1: 读访问时使用映射

#### 9.4.65. FMMUX 激活寄存器 (FMMUX\_ACTIVE) (X = 0...F)

地址偏移:  $0x060C + X * 0x10$

复位值: 0x0

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
保留							ACTIVE_F
							MMU
							rw(ECAT)
							r(PDI)

位/位域	名称	描述
7:1	保留	必须保持复位值。
0	ACTIVE_FMMU	0: FMMU未激活 1: FMMU已激活。FMMU根据配置的映射检查逻辑地址的块是否需要映射

#### 9.4.66. SyncManager X 物理起始地址寄存器 (SMX\_ADDR) (X = 0...F)

地址偏移量:  $0x0800 + X * 8$

复位值: 0x0

该寄存器可以按半字 (16位) 访问。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMSTART_ADDR[15:0]															
nw(ECAT)															
r(PDI)															

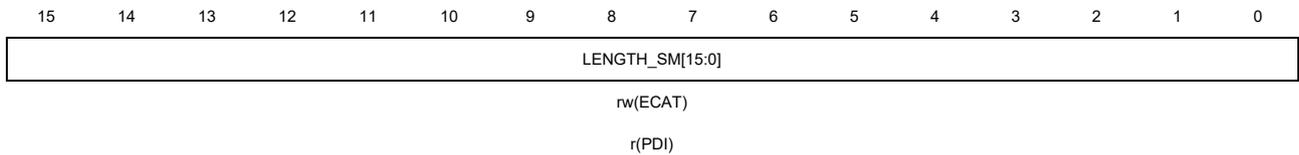
位/位域	名称	描述
15:0	SMSTART_ADDR[15:0]	将由SyncManager处理的第一个字节

### 9.4.67. SyncManager X 长度寄存器 (SMX\_LENGTH) (X = 0...F)

地址偏移:  $0x0802 + X * 8$

复位值: 0x0

该寄存器可以按半字 (16位) 访问。



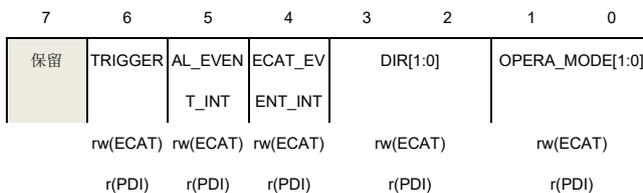
位/位域	名称	描述
15:0	LENGTH_SM[15:0]	分配给SyncManager的字节数 (应大于1, 否则SyncManager不会被激活。如果设置为1, 只有在配置了看门狗触发器时才会生成)

### 9.4.68. SyncManager X 控制寄存器 (SMX\_CTL) (X = 0...F)

地址偏移量:  $0x0804 + X * 8$

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7	保留	必须保持复位值。
6	TRIGGER	看门狗触发器启用: 0: 禁用 1: 启用
5	AL_EVENT_INT	中断在AL事件请求寄存器中: 0: 禁用 1: 启用
4	ECAT_EVENT_INT	中断在以ECAT事件请求寄存器中: 0: 禁用 1: 启用
3:2	DIR[1:0]	方向: 00: 读: ECAT读访问, PDI写访问。 01: 写: ECAT写访问, PDI读访问。 10: 保留

		11: 保留
1:0	OPERA_MODE[1:0]	操作模式: 00: 缓冲 (3缓冲模式) 01: 保留 10: 邮箱 (单缓冲模式) 11: 保留

#### 9.4.69. SyncManager X 状态寄存器 (SMX\_STA) (X = 0...F)

地址偏移量:  $0x0805 + X * 8$

复位值: 0x30

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
WRITE_BUFFER	READ_BUFFER	BUFFER_STATUS[1:0]		MAILBOX_STATUS	保留	READ_INT	WRITE_INT
r(ECAT)	r(ECAT)	r(ECAT)		r(ECAT)		r(ECAT)	r(ECAT)
r(PDI)	r(PDI)	r(PDI)		r(PDI)		r(PDI)	r(PDI)

位/位域	名称	描述
7	WRITE_BUFFER	正在使用的写缓冲区 (已打开)
6	READ_BUFFER	正在使用的读缓冲区 (已打开)
5:4	BUFFER_STATUS[1:0]	缓冲模式: 缓冲状态 (最后写入的缓冲区): 00: 第一个缓冲区 01: 第二个缓冲区 10: 第三个缓冲区 11: (没有缓冲区被写入) 邮箱模式: 保留
3	MAILBOX_STATUS	邮箱模式: 邮箱状态: 0: 邮箱为空 1: 邮箱已满 缓冲模式: 保留
2	保留	必须保持复位值。
1	READ_INT	中断读取: 1: 在缓冲区完全且成功读取后中断 0: 在缓冲区写入第一个字节后清除中断 <b>注意:</b> 如果已在SM控制寄存器中启用, 此中断将被信号到写入端。
0	WRITE_INT	中断写入: 1: 在缓冲区完全且成功写入后中断 0: 在缓冲区读取第一个字节后清除中断

**注意：**如果已在SM控制寄存器中启用，此中断将被信号到读取端。

### 9.4.70. SyncManager X 激活寄存器 (SMX\_ACTIVE) (X = 0...F)

地址偏移:  $0x0806 + X * 8$

复位值: 0x0

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
LATCH_P DI	LATCH_E CAT	保留				REQUEST	SM_EN
rw(ECAT) rw(ECAT)						rw(ECAT)	r(ECAT)
r(PDI) r(PDI)						r(PDI)	rw(PDI)

位/位域	名称	描述
7	LATCH_PDI	Latch事件PDI: 0: 否 1: 当PDI发出缓冲区交换或当PDI访问缓冲区起始地址时生成Latch事件
6	LATCH_ECAT	Latch事件ECAT: 0: 否 1: 当以太网CAT主控器发出缓冲区交换时生成Latch事件
5:2	保留	必须保持复位值。
1	REQUEST	重复请求: 重复请求的切换意味着需要进行邮箱重试 (主要与ECAT读取邮箱一起使用)
0	SM_EN	同步管理器启用/禁用: 0: 禁用: 在没有同步管理器控制的情况下访问内存 1: 启用: 同步管理器处于活动状态, 并控制配置中设置的内存区域 <b>注意:</b> PDI寄存器功能通过写命令确认被禁用: 从PDI读取此寄存器将清除所有已更改激活状态的同步管理器的AL事件请求0x0220[4]。从PDI写入此寄存器是不可能的。

### 9.4.71. PDI 控制寄存器 (SMX\_PDICTL) (X = 0...F)

地址偏移:  $0x0807 + X * 8$

复位值: 0x0

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
保留						ACK	DEACTIV E_SM
						r(ECAT)	r(ECAT)
						rw(PDI)	rw(PDI)

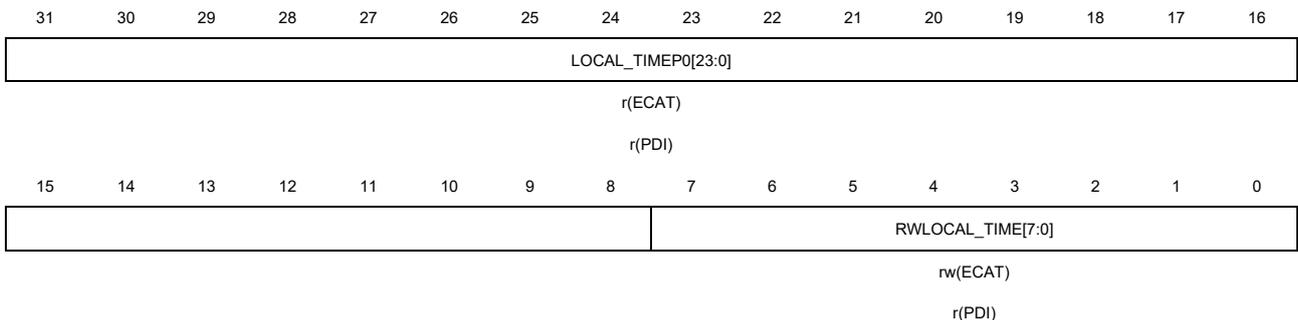
位/位域	名称	描述
7:2	保留	必须保持复位值。
1	ACK	重复确认： 如果此值设置为与重复请求所设置的值相同，PDI将确认执行之前的重复请求。
0	DEACTIVE_SM	停用同步管理器： 读取： 0: 正常操作，同步管理器已激活。 1: 同步管理器已停用并重置。同步管理器锁定对内存区域的访问。 写入： 0: 激活同步管理器 1: 请求停用同步管理器 <b>注意：</b> 写入1将延迟直到当前正在处理的帧结束。

#### 9.4.72. ESC 接收时间端口 0 寄存器 (ESC\_RECVE\_TIMEP0)

地址偏移: 0x0900

复位值: 未定义

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
31:8	LOCAL_TIMEP0[23:0]	包含对寄存器0x0900的写访问的最后一个接收帧开始时的本地时间
7:0	RWLOCAL_TIME[7:0]	写入: 使用BWR或FPWR对寄存器0x0900的写访问会在每个端口的接收帧开始时 (前导码的第一个比特) 锁定本地时间。 读取: 包含对此寄存器的写访问的最后一个接收帧开始时的本地时间。 <b>注意:</b> FPWR需要像任何FPWR命令一样进行地址匹配才能访问此寄存器。所有具有地址匹配的写命令将增加工作计数器 (例如, APWR), 但它们不会触发接收时间锁定。

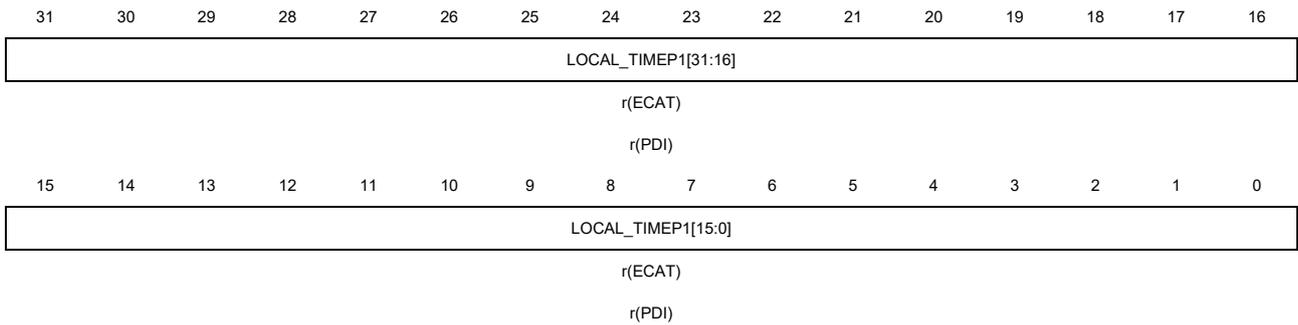
**注意:** 在写入此寄存器的同一帧中无法读取时间戳。

#### 9.4.73. ESC 接收时间端口 1 寄存器 (ESC\_RECVE\_TIMEP1)

地址偏移: 0x0904

复位值: 未定义

该寄存器可以按字（32位）访问。



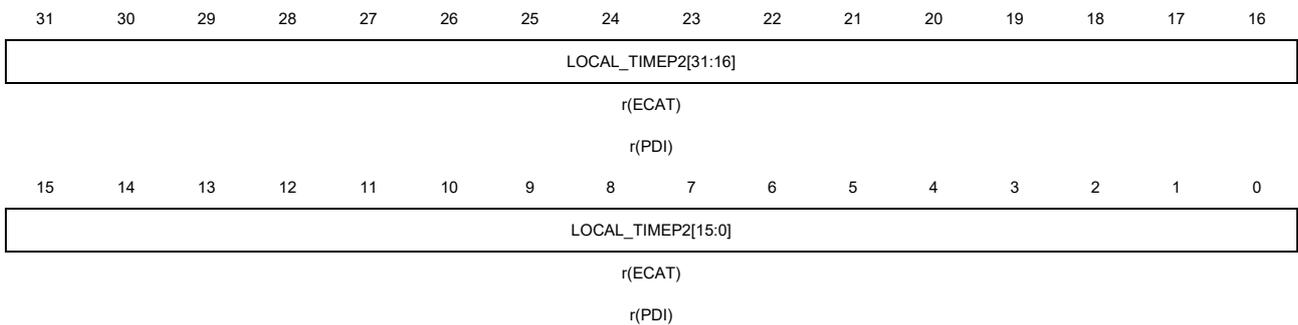
位/位域	名称	描述
31:0	LOCAL_TIMEP1[31:0]	在端口1接收到的包含对寄存器0x0900的BWR或FPWR的帧的开始（前导码的第一个比特）的本地时间。

#### 9.4.74. ESC 接收时间端口 2 寄存器（ESC\_RECVE\_TIMEP2）

地址偏移：0x0908

复位值：未定义

该寄存器可以按字（32位）访问。



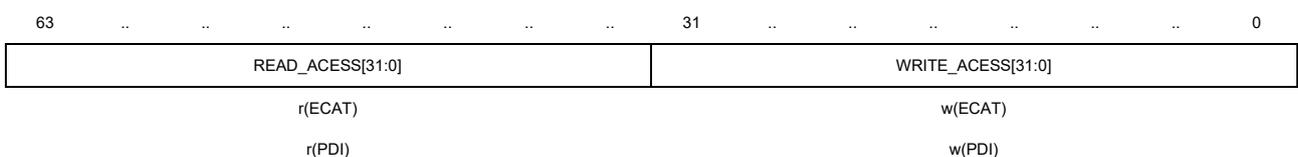
位/位域	名称	描述
31:0	LOCAL_TIMEP2[31:0]	在端口2接收到的包含对寄存器0x0900的BWR或FPWR的帧的开始（前导码的第一个比特）的本地时间。

#### 9.4.75. ESC 系统时间寄存器（ESC\_SYS\_TIME）

地址偏移：0x0910

复位值：0x0

该寄存器可以按字（32位）访问。



位/位域	名称	描述
63:0	READ_ACCESS[63:0]	<p>ECAT读取访问： 当帧通过参考时钟时的系统时间的本地副本（即，包括系统时间延迟）。时间在帧的开始时锁定（以太网SOF分隔符）</p> <p>PDI读取访问： 系统时间的本地副本。在读取第一个字节（0x0910）时锁定时间。</p>
31:0	WRITE_ACCESS[31:0]	<p>写入访问： 写入的值将与系统时间的本地副本进行比较。结果是时间控制循环的输入。</p> <p><b>注意：</b>如果至少写入了第一个字节（0x0910），写入的值将在帧的末尾与锁定的（SOF）系统时间的本地副本进行比较。</p>

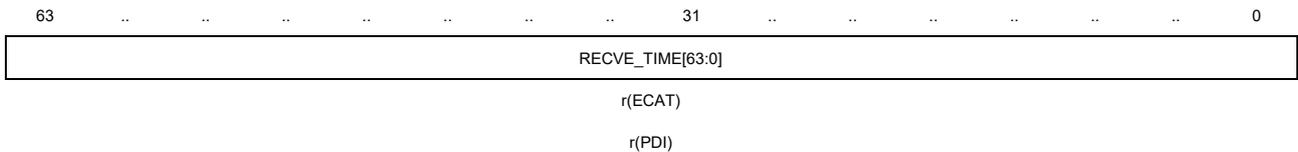
**注意：**此寄存器的写访问权限取决于ESC配置（系统时间PDI控制关闭=ECAT/开启=PDI；ECAT控制是通用的）。

#### 9.4.76. ESC 接收时间 ECAT 处理单元寄存器（ESC\_RCVMTIME）

地址偏移：0x0918

复位值：未定义

该寄存器可以按字（32位）访问。



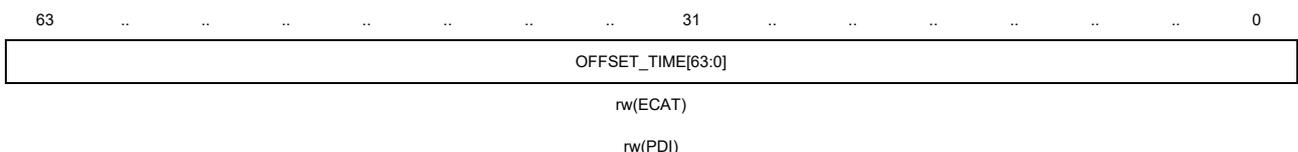
位/位域	名称	描述
63:0	RECVE_TIME[63:0]	<p>在ECAT处理单元接收到的包含对寄存器0x0900的写访问的帧的开始（前导码的第一个比特）的本地时间</p> <p><b>注意：</b>如果端口0是开放的，此寄存器以64位值的形式反映端口0的接收时间。任何有效的ECAT写访问到寄存器0x0900都会触发锁定，而不仅仅是像寄存器0x0900那样的BWR/FPWR命令。</p>

#### 9.4.77. ESC 系统时间偏移寄存器（ESC\_OFFSET\_TIME）

地址偏移：0x0920

复位值：0x0

该寄存器可以按字（32位）访问。



位/位域	名称	描述
------	----	----

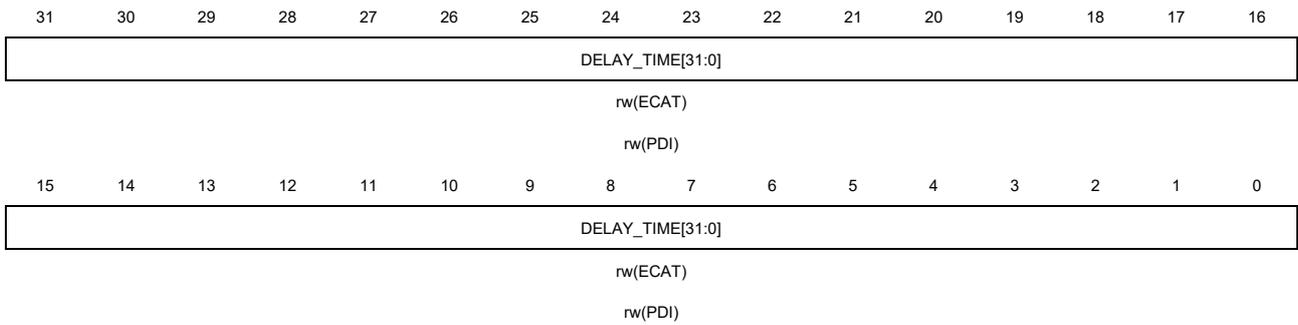
63:0      OFFSET\_TIME[63:0] 本地时间与系统时间之间的差异。偏移量被加到本地时间上。

#### 9.4.78. ESC 系统时间延迟寄存器 (ESC\_DELAY\_TIME)

地址偏移: 0x0928

复位值: 0x0

该寄存器可以按字(32位)访问。



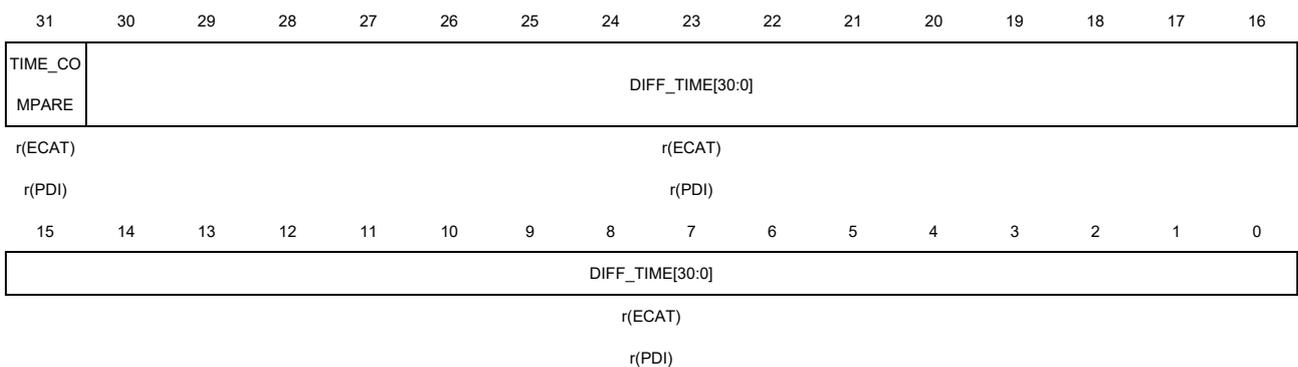
位/位域	名称	描述
31:0	DELAY_TIME[31:0]	参考时钟与ESC之间的延迟

#### 9.4.79. ESC 系统时间差值寄存器 (ESC\_DIFF\_TIME)

地址偏移: 0x092C

复位值: 0x0

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
31	TIME_COMPARE	0: 本地系统时间副本小于接收到的系统时间 1: 本地系统时间副本大于或等于接收到的系统时间
30:0	DIFF_TIME[30:0]	系统时间的本地副本与接收到的系统时间值之间的平均差异 差异 = 接收到的系统时间 - 系统时间的本地副本

**注意:** 当读取位[7:0]时, 位[31:8]会在ECAT/PDI内部被锁定 (独立地), 这保证了读取到的值是

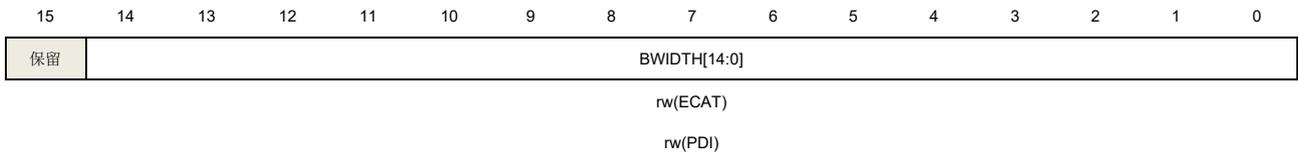
一致的。

#### 9.4.80. ESC 速度计数器起始值寄存器 (ESC\_COUNT\_START)

地址偏移: 0x0930

复位值: 0x1000

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15	保留	必须保持复位值。
14:0	BWIDTH[14:0]	调整系统时间本地副本的带宽 (较大值 -> 较小带宽和更平滑的调整) 写访问会重置系统时间差异 (0x092C:0x092F) 和速度计数器差异 (0x0932:0x0933)。有效值: 0x0080至0x3FFF

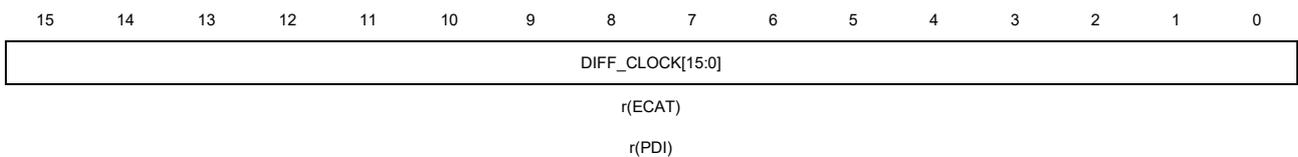
**注意:** 此寄存器的写访问权限取决于ESC配置 (系统时间PDI控制关闭=ECAT / 开启=PDI; ECAT控制是通用的)。

#### 9.4.81. ESC 速度计数器差值寄存器 (ESC\_COUNT\_DIFF)

地址偏移: 0x0932

复位值: 0x0000

该寄存器可以按半字 (16位) 访问。



位/位域	名称	描述
15:0	DIFF_CLOCK[15:0]	本地时钟周期与参考时钟周期之间的偏差表示 (表示方式: 二进制补码) 范围: ± (速度计数器起始值 - 0x7F)

**注意:** 系统时间差稳定在低值后, 时钟偏差可以按以下方式计算:

偏差 = 速度计数器差值 / 5(速度计数器起始 + 速度计数器差值 + 2)(速度计数器起始 - 速度计数器差值 + 2)

#### 9.4.82. ESC 系统时间差值滤波器深度寄存器 (ESC\_TIME\_DIFF)

地址偏移: 0x0934

复位值：0x04

该寄存器可以按字节（8位）访问。

7	6	5	4	3	2	1	0
保留				FILTER_DAPTH[3:0]			
rw(ECAT)							
rw(PDI)							

位/位域	名称	描述
7:4	保留	必须保持复位值。
3:0	FILTER_DAPTH[3:0]	用于平均接收到的系统时间偏差的滤波深度

**注意：**此寄存器的写访问权限取决于ESC配置（系统时间PDI控制关闭=ECAT/开启=PDI；ECAT控制是通用的）。

### 9.4.83. ESC 速度计数器滤波器深度寄存器（ESC\_SPEED\_COUNT）

地址偏移：0x0935

复位值：0x12

该寄存器可以按字节（8位）访问。

7	6	5	4	3	2	1	0
保留				FILTER_DAPTH[3:0]			
rw(ECAT)							
rw(PDI)							

位/位域	名称	描述
7:4	保留	必须保持复位值。
3:0	FILTER_DAPTH[3:0]	用于平均时钟周期偏差的滤波深度。

**注意：**写入此寄存器的权限取决于ESC配置（系统时间PDI控制关闭=ECAT/开启=PDI；ECAT控制是通用的）。

### 9.4.84. ESC 循环单元控制寄存器（ESC\_UNIT\_CTL）

地址偏移：0x0980

复位值：0x0

该寄存器可以按字节（8位）访问。

7	6	5	4	3	2	1	0
保留		LATCH_U	LATCH_U	保留			UNIT_CTL
		NIT1	NIT0				
		rw(ECAT)	rw(ECAT)				rw(ECAT)
		r(PDI)	r(PDI)				r(PDI)

位/位域	名称	描述
7:6	保留	必须保持复位值。
5	LATCH_UNIT1	锁存单元1: 0: ECAT控制 1: PDI控制 <b>注意:</b> 根据此设置, 锁存中断被路由到ECAT /PDI
4	LATCH_UNIT0	锁存单元0: 0: ECAT控制 1: PDI控制 注意: 根据此设置, 锁存中断被路由到ECAT /PDI。如果系统时间由PDI控制, 则始终为1 (PDI控制)。
3:1	保留	必须保持复位值。
0	UNIT_CTL	循环单元和SYNC0输出单元控制: 0: ECAT控制 1: PDI控制

#### 9.4.85. ESC 激活寄存器 (ESC\_REGISTER\_ACTIVE)

地址偏移: 0x0981

复位值: 0x0

该寄存器可以按字节 (8位) 访问。

7	6	5	4	3	2	1	0
SYNCSIG	CONFIG	START_TI	EXTEN	AUTO_AC	SYNC1	SYNC0	SYNC_OU
NAL		ME		TIVE			T
rw(ECAT)							
rw(PDI)							

位/位域	名称	描述
7	SYNCSIGNAL	同步信号调试脉冲 (Vasily位): 0: 未激活 1: 根据0x0981[2:1]立即仅在SYNC0-1上生成一个脉冲, 用于调试 此位是自清除的, 总是读取为0。 所有脉冲同时生成, 循环时间被忽略。使用配置的脉冲长度。
6	CONFIG	近期配置 (大约): 0: 1/2 DC宽度的一半特征(2 <sup>31</sup> ns 或2 <sup>63</sup> ns) 1: 大约2.1秒(2 <sup>31</sup> ns)
5	START_TIME	开始时间合理性检查: 0: 禁用。如果达到开始时间, 则生成同步信号。 1: 如果开始时间不在近期未来 (见0x0981[6]) 内, 则立即生成同步信号。
4	EXTEN	开始时间循环操作的扩展 (0x0990:0x0993):

		0: 不扩展 1: 将32位写入的开始时间扩展到64位
3	AUTO_ACTIVE	通过写入开始时间循环操作自动激活（0x0990:0x0997）： 0: 禁用 1: 启用自动激活。在写入开始时间后，0x0981[0]会自动设置。
2	SYNC1	SYNC1生成： 0: 停用 1: 生成SYNC1脉冲
1	SYNC0	SYNC0 生成： 0: 停用 1: 生成SYNC0 脉冲
0	SYNC_OUT	同步输出单元激活： 0: 停用 1: 激活

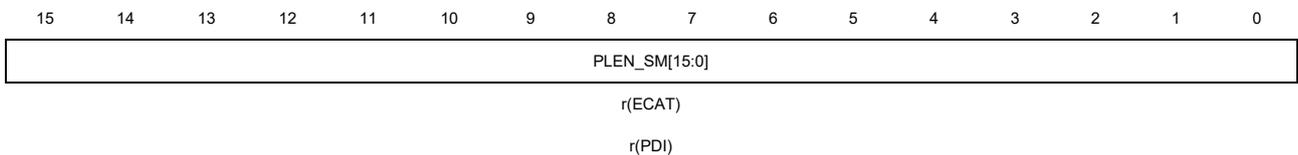
**注意：**写入此寄存器取决于0x0980[0]的设置。

#### 9.4.86. ESC 同步信号寄存器的脉冲长度（ESC\_PLEN\_SM）

地址偏移：0x0982

复位值：取决于配置

该寄存器可以按半字（16位）访问。



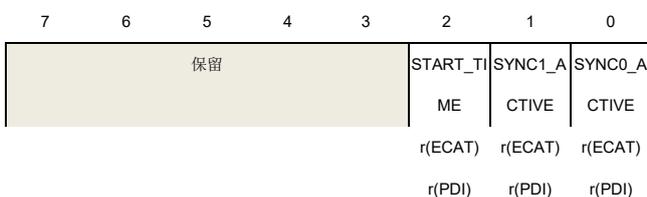
位/位域	名称	描述
15:0	PLEN_SM[15:0]	同步信号的脉冲长度（以10ns为单位） 0: 确认模式：同步信号将在读取SYNC[1:0]状态寄存器时清除

#### 9.4.87. ESC 激活状态寄存器（ESC\_ACTIVE\_STATUS）

地址偏移量：0x0984

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:3	保留	必须保持复位值。
2	START_TIME	同步输出单元激活时的循环操作开始时间（0x0990:0x0997）合理性检查结果： 0: 开始时间在近期内 1: 开始时间不在近期内（0x0981[6]）
1	SYNC1_ACTIVE	SYNC1激活状态： 0: 第一个SYNC1脉冲未待处理 1: 第一个SYNC1脉冲待处理
0	SYNC0_ACTIVE	SYNC0激活状态： 0: 第一个SYNC0脉冲未待处理 1: 第一个SYNC0脉冲待处理

#### 9.4.88. ESC SYNC0 状态寄存器（ESC\_SYNC0\_STATUS）

地址偏移：0x098E

复位值：0x0

该寄存器可以按字节（8位）访问。



位/位域	名称	描述
7:1	保留	必须保持复位值。
0	SYNC0_STATE	确认模式下的SYNC0状态。 在确认模式下，通过从PDI读取此寄存器来清除SYNC0，仅在确认模式下使用 <b>注意：</b> PDI寄存器功能通过写命令确认被禁用：从PDI读取此寄存器将清除AL事件请求0x0220[2]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用：从PDI写入此寄存器将清除AL事件请求0x0220[2]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。

#### 9.4.89. ESC SYNC1 状态寄存器（ESC\_SYNC1\_STATUS）

地址偏移：0x098F

复位值：0x0

该寄存器可以按字节（8位）访问。





63:0 NEXT\_SYNC1[63:0] 下一个SYNC1脉冲的系统时间以ns为单位。

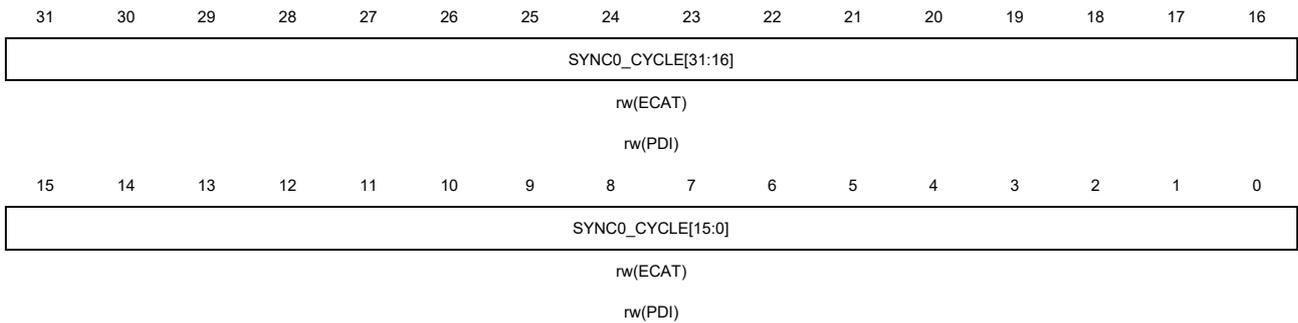
**注意：**当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地锁存，这保证了读取到的值是一致的。

#### 9.4.92. ESC SYNC0 周期时间寄存器 (ESC\_SYNC0\_CYCLE)

地址偏移：0x09A0

复位值：0x0

该寄存器可以按字（32位）访问。



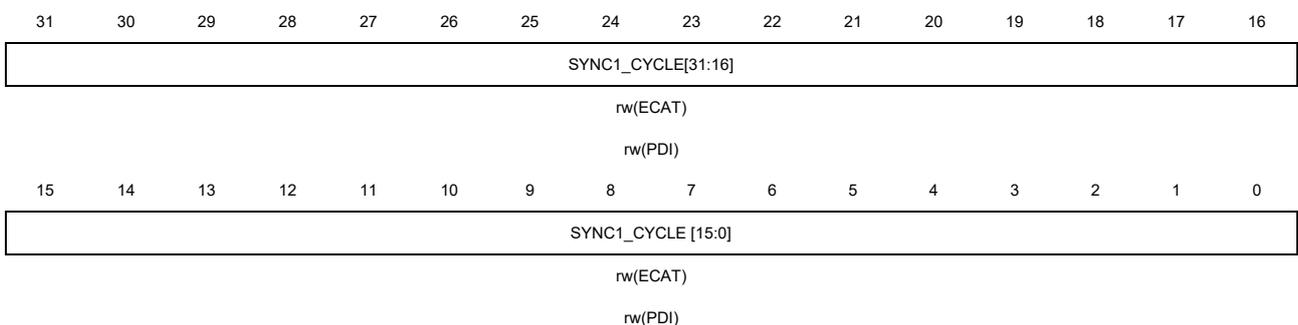
位/位域	名称	描述
31:0	SYNC0_CYCLE[31:0]	两个连续SYNC0脉冲之间的时间以ns为单位。 0：单次触发模式，只产生一个SYNC0脉冲。 <b>注意：</b> 写入此寄存器取决于0x0980[0]的设置。循环操作的最小值：60 [ns]。

#### 9.4.93. ESC SYNC1 周期时间寄存器 (ESC\_SYNC1\_CYCLE)

地址偏移：0x09A4

复位值：0x0

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:0	SYNC1_CYCLE[31:0]	SYNC0脉冲和SYNC1脉冲之间的时间以ns为单位。 <b>注意：</b> 写入此寄存器取决于0x0980[0]的设置。

### 9.4.94. ESC LATCH0 控制寄存器 (ESC\_LATCH0\_CTL)

地址偏移: 0x09A8

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:2	保留	必须保持复位值。
1	LATCH0_NEGATIVE	Latch0下降沿: 0: 连续Latch激活 1: 单事件 (只有第一个事件激活)
0	LATCH0_POSITIVE	Latch0上升沿: 0: 连续Latch激活 1: 单事件 (只有第一个事件激活)

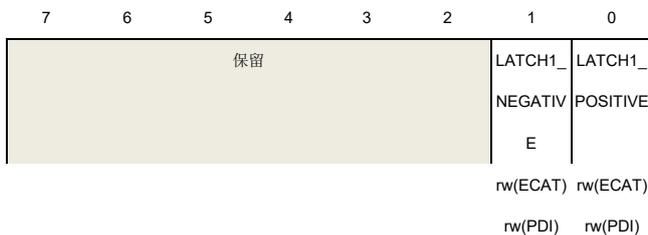
写入访问取决于0x0980[4]的设置。

### 9.4.95. ESC LATCH1 控制寄存器 (ESC\_LATCH1\_CTL)

地址偏移量: 0x09A9

复位值: 0x0

该寄存器可以按字节 (8位) 访问。



位/位域	名称	描述
7:2	保留	必须保持复位值。
1	LATCH1_NEGATIVE	Latch1下降沿: 0: 连续Latch激活 1: 单事件 (只有第一个事件激活)
0	LATCH1_POSITIVE	Latch1上升沿:

- 0: 连续Latch激活
- 1: 单事件（只有第一个事件激活）

**注意：**写访问权限取决于0x0980[5]的设置。

#### 9.4.96. ESC LATCH0 状态寄存器（ESC\_LATCH0\_STATUS）

地址偏移：0x09AE

复位值：0x0

该寄存器可以按字节（8位）进行访问。

7	6	5	4	3	2	1	0
保留				LOPIN_ST	ELATCH0	ELATCH0	
				ATE	_NEGATI	_POSITIV	
					VE	E	
				r(ECAT)	r(ECAT)	r(ECAT)	
				r(PDI)	r(PDI)	r(PDI)	

位/位域	名称	描述
7:3	保留	必须保持复位值。
2	LOPIN_STATE	Latch0引脚状态
1	ELATCH0_NEGATIVE	事件 Latch0下降沿。 0: 未检测到下降沿或连续模式 1: 仅在单事件模式下检测到下降沿。 通过读取 Latch0 时间下降沿清除标志。
0	ELATCH0_POSITIVE	事件 Latch0上升沿。 0: 未检测到上升沿或连续模式 1: 仅在单事件模式下检测到上升沿。 通过读取 Latch0 时间上升沿清除标志

#### 9.4.97. ESC LATCH1 状态寄存器（ESC\_LATCH1\_STATUS）

地址偏移：0x09AF

复位值：0x0

该寄存器可以按字节（8位）访问。

7	6	5	4	3	2	1	0
保留				L1PIN_ST	ELATCH1	ELATCH1	
				ATE	_NEGATI	_POSITIV	
					VE	E	
				r(ECAT)	r(ECAT)	r(ECAT)	
				r(PDI)	r(PDI)	r(PDI)	

位/位域	名称	描述
------	----	----

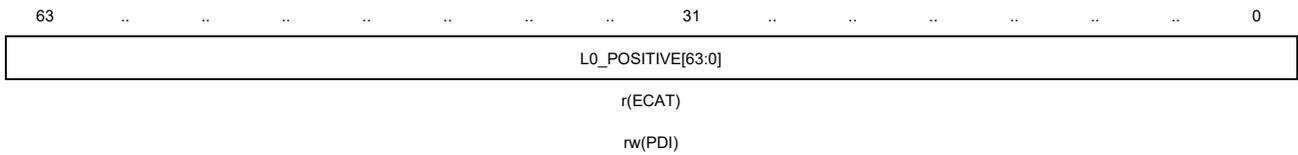
7:3	保留	必须保持复位值。
2	L1PIN_STATE	Latch1引脚状态
1	ELATCH1_NEGATIVE	事件 Latch1下降沿。 0: 未检测到下降沿或连续模式 1: 仅在单事件模式下检测到下降沿。 通过读取 Latch1 时间下降沿清除标志。
0	ELATCH1_POSITIVE	事件 Latch1上升沿。 0: 未检测到上升沿或连续模式 1: 仅在单事件模式下检测到上升沿。 通过读取 Latch1 时间上升沿清除标志

### 9.4.98. ESC LATCH0 时间上升沿寄存器 (ESC\_LATCH0\_POSITIVE)

地址偏移: 0x09B0

复位值: 0x0

该寄存器可以按字 (32位) 访问。



位/位域	名称	描述
63:0	L0_POSITIVE[63:0]	在Latch0信号的上升沿的系统时间。 <b>注意:</b> PDI寄存器功能通过写命令确认被禁用: 如果0x0980[4]=1, 从PDI读取此寄存器将清除Latch0状态0x09AE[0]。从PDI写入此寄存器是不可能的。 PDI寄存器功能通过写命令确认被启用: 如果0x0980[4]=1, 从PDI写入此寄存器将清除Latch0状态0x09AE[0]。从PDI写入此寄存器是可能的; 写入的值被忽略(写入0)。

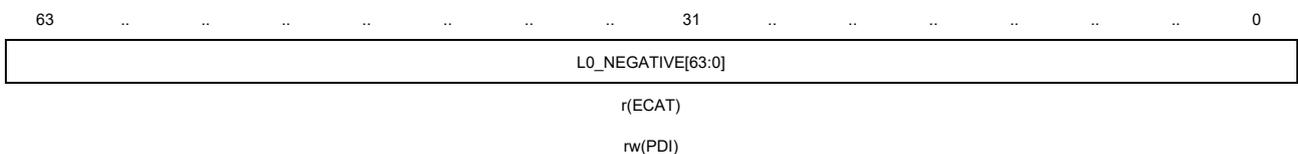
**注意:** 当读取位[7:0]时, 位[63:8]会在ECAT/PDI内部独立地被锁存, 这保证了读取到的值是一致的。从ECAT读取此寄存器时, 如果0x0980[4]=0, 则会清除Latch0状态0x09AE[0]。从ECAT写入此寄存器是不可能的。

### 9.4.99. ESC LATCH0 时间下降沿寄存器 (ESC\_LATCH0\_NEGATIVE)

地址偏移: 0x09B8

复位值: 0x0

该寄存器可以按字 (32位) 访问。



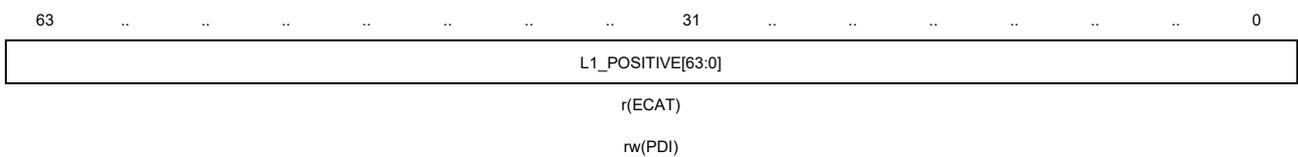
位/位域	名称	描述
63:0	L0_NEGATIVE[63:0]	<p>在Latch0信号的下降沿的系统时间。</p> <p><b>注意：</b> PDI寄存器功能通过写命令确认被禁用：如果0x0980[4]=1，从PDI读取此寄存器将清除Latch0状态0x09AE[1]。从PDI写入此寄存器是不可能的。</p> <p>PDI寄存器功能通过写命令确认被启用：如果0x0980[4]=1，从PDI写入此寄存器将清除Latch0状态0x09AE[1]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。</p> <p><b>注意：</b> 当读出位[7:0]时，位[63:8]会在内部被锁定（ECAT/PDI独立），这保证了读取到一致的值。如果0x0980[4]=0，从ECAT读取此寄存器会清除Latch0状态0x09AE[1]。从ECAT写入此寄存器是不可能的。</p>

#### 9.4.100. ESC LATCH1 时间上升沿寄存器（ESC\_LATCH1\_POSITIVE）

地址偏移：0x09C0

复位值：0x0

该寄存器可以按字（32位）访问。



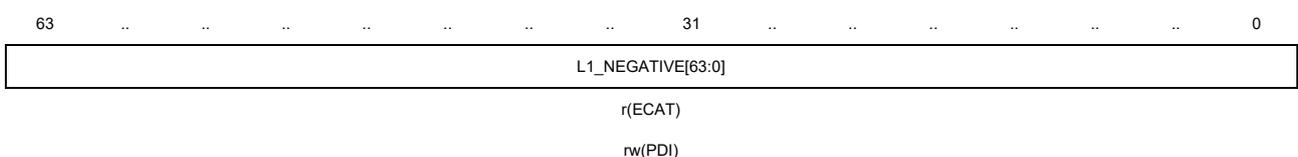
位/位域	名称	描述
63:0	L1_POSITIVE[63:0]	<p>在Latch1信号的上升沿的系统时间。</p> <p><b>注意：</b> PDI寄存器功能通过写命令确认被禁用：如果0x0980[5]=1，从PDI读取此寄存器将清除Latch0状态0x09AF[0]。从PDI写入此寄存器是不可能的。</p> <p>PDI寄存器功能通过写命令确认被启用：如果0x0980[5]=1，从PDI写入此寄存器将清除Latch0状态0x09AF[0]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。</p> <p><b>注意：</b> 当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到的值是一致的。如果0x0980[5]=0，从ECAT读取此寄存器会清除Latch0状态0x09AF[0]。从ECAT写入此寄存器是不可能的。</p>

#### 9.4.101. ESC LATCH1 时间下降沿寄存器（ESC\_LATCH1\_NEGATIVE）

地址偏移：0x09C8

复位值：0x0

该寄存器可以按字（32位）进行访问。



位/位域	名称	描述
63:0	L1_NEGATIVE[63:0]	<p>在Latch1信号的下降沿的系统时间。</p> <p><b>注意：</b> PDI寄存器功能通过写命令确认被禁用：如果0x0980[5]=1，从PDI读取此寄存器将清除Latch0状态0x09AF[1]。从PDI写入此寄存器是不可能的。</p> <p>PDI寄存器功能通过写命令确认被启用：如果0x0980[5]=1，从PDI写入此寄存器将清除Latch0状态0x09AF[1]。从PDI写入此寄存器是可能的；写入的值被忽略（写入0）。</p>

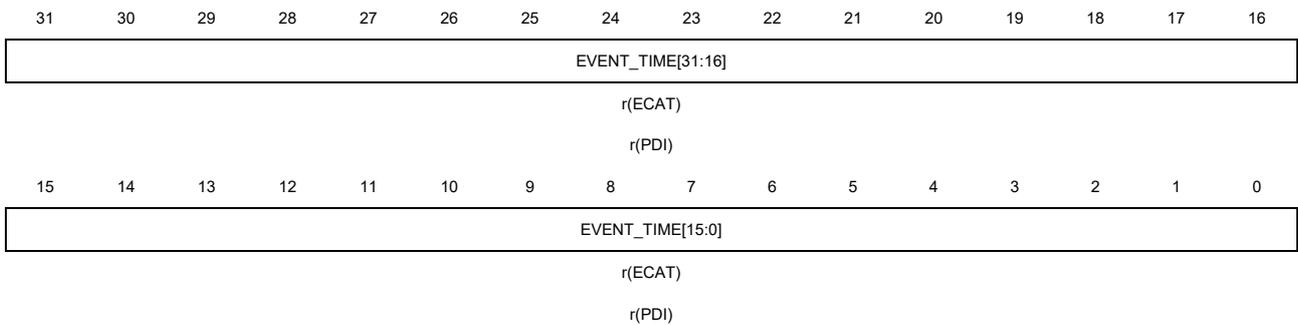
**注意：** 当读取位[7:0]时，位[63:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到的值是一致的。如果0x0980[5]=0，从ECAT读取此寄存器将清除Latch0状态0x09AF[1]。从ECAT写入此寄存器是不可能的。

#### 9.4.102. ESC 缓冲区变化事件时间寄存器（ESC\_EVENT\_TIME）

地址偏移：0x09F0

复位值：0x0

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:0	EVENT_TIME[31:0]	帧开始时的本地时间，该帧至少导致一个同步管理器（SyncManager）有效一个（ECAT）事件。

**注意：** 当读取位[7:0]时，位[31:8]会在内部被ECAT/PDI独立地锁存，这保证了读取到的值是一致的。

#### 9.4.103. ESC PDI 缓冲区起始事件时间寄存器（ESC\_PDI\_SEVENT\_TIME）

地址偏移：0x09F8

复位值：0x0

该寄存器可以按字（32位）访问。



r(ECAT)

r(PDI)

位/位域	名称	描述
31:0	PDI_SEVENT_TIME	当至少一个同步管理器（SyncManager）有效PDI缓冲区开始事件时的本地时间。 31:0]

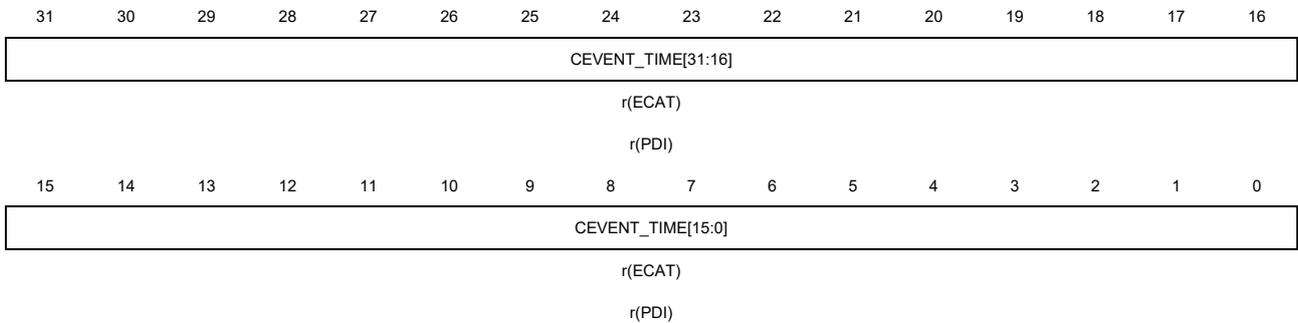
**注意：**当读取位[7:0]时，位[31:8]会在内部被ECAT/PDI独立地锁定，这保证了读取到一致的值。

#### 9.4.104. ESC PDI 缓冲区变化事件时间寄存器（ESC\_PDI\_CEVENT\_TIME）

地址偏移：0x09FC

复位值：0x0

该寄存器可以按字（32位）访问。



位/位域	名称	描述
31:0	PDI_CEVENT_TIME	当至少一个SyncManager有效PDI缓冲区更改事件时的本地时间 31:0]

**注意：**当读取位[7:0]时，位[31:8]会在ECAT/PDI内部独立地被锁存，这保证了读取到一致的值。

#### 9.4.105. ESC 产品 ID 寄存器（ESC\_PRODUCT\_ID）

地址偏移：0x0E00

复位值：0x0

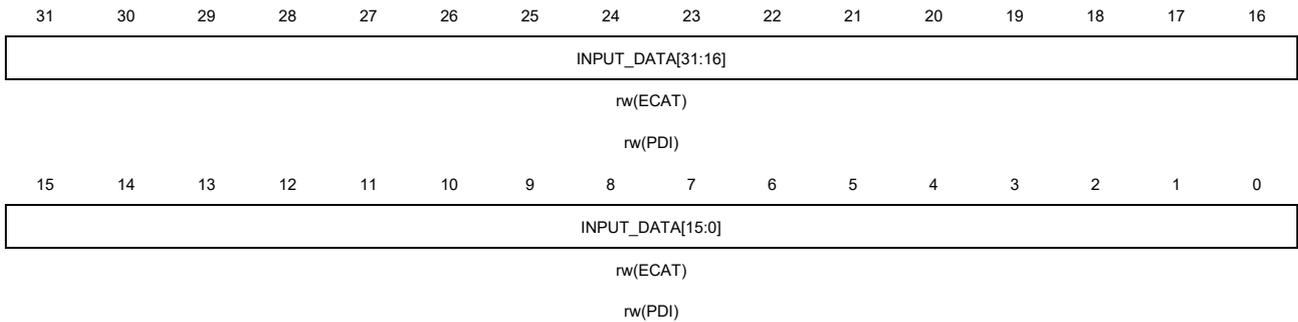
该寄存器可以按字节（8位）、半字（16位）和字（32位）进行访问。



位/位域	名称	描述
63:0	PRODUCT_ID[63:0]	产品ID







位/位域	名称	描述
31:0	INPUT_DATA[31:0]	输入数据

**注意：**只有在正确加载了EEPROM(寄存器0x0110[0] = 1)的情况下,才能访问进程数据RAM。

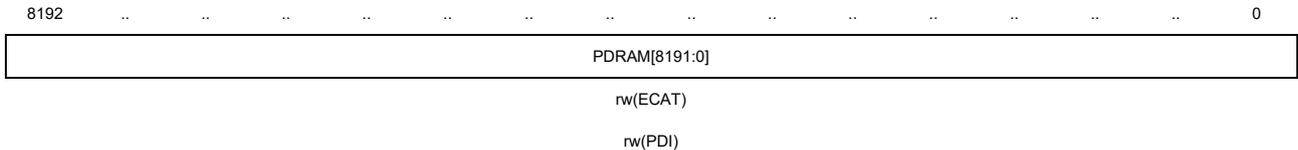
输入数据的大小取决于PDI设置和/或设备配置。如果配置了带有输入的**数字I/O PDI**,则**数字I/O**输入数据将写入这些地址的进程数据RAM中。

#### 9.4.112. ESC 过程数据 RAM (ESC\_PDRAM)

地址偏移: 0x1000

复位值: 未定义

该寄存器可以按字(32位)访问。



位/位域	名称	描述
----	PDRAM[8191:0]	过程数据RAM(8KB)

**注意：**只有当EEPROM正确加载时(寄存器0x0110[0] = 1),才能访问处理数据RAM。

## 10. 修订历史

表 10-1. 修订历史

版本号	描述	日期
1.0	初始版本	2024 年 11 月 8 日

## Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the "Product"), is owned by the Company under the intellectual property laws and treaties of the People's Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as its suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.