

# 融信网通 RXNOE 专用网络卸载处理器

RXNOE (TCP/UDP/ICMP/IGMP/ARP)

## 用户指南

文档版本：H0008

发布日期：2016.2

版权所有 © 深圳融信网通科技有限公司 2016. 保留一切权利

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本档内容的部分或全部，并不得以任何形式传播。

## 深圳融信网通科技有限公司

地址：深圳市南山区东滨路濠盛商务中心 7 栋 707

网址：<http://www.rxwt.com.cn>

技术支持邮箱：[support@rxwt.com.cn](mailto:support@rxwt.com.cn)

客户服务邮箱：[service@rxwt.com.cn](mailto:service@rxwt.com.cn)

## TABLE OF CONTENTS

1	前言	1
2	产品概述	3
2.1	功能描述	3
2.2	系统结构	5
2.3	接口描述	5
2.4	集成指导	7
3	典型应用	8
3.1	RXNoE 初始化	8
3.2	TCP 连接建立	9
3.2.1	client 模式	9
3.2.2	server 模式	10
3.3	TCP 连接删除	10
3.3.1	正常删除	10
3.3.2	强制删除	11
3.4	TCP 数据收发	11
3.5	UDP 服务	11
3.5.1	UDP Client	11
3.5.2	UDP Server	11
3.5.3	删除 UDP 服务	12
3.5.4	UDP 广播	12
3.5.5	UDP 组播	12
3.6	IGMP 服务	12
3.6.1	加入组播域	12
3.6.2	退出组播域	13
4	寄存器描述	14
4.1	寄存器地址分配	14
4.2	寄存器详细描述	14
4.2.1	公共寄存器	14
4.2.2	UDP 寄存器	14
4.2.3	TCP 寄存器	16
4.2.4	IGMP 寄存器	19
4.2.5	MAC 寄存器	20

## List of Abbreviations

FPGA	Field Programmed Logic Array
HFT	High Frequency Trading
GiBe	Giga Bit Ethernet
JTAG	Joint Test Action Group
SGDMA	Scatter-Gather Direct Memory Access
FSM	Finite State Machine

# 1 前言

## 概述

本文档介绍了 RXNoE 产品的特性、逻辑结构，详细描述各个模块的功能、工作方式、相关寄存器定义，用图表的方式给出了接口时序关系和相关参数。

## 读者对象

本文档主要适用于以下工程师：

- RXNoE FPGA/ASIC 集成设计工程师；
- 系统应用与驱动开发工程师；
- 模块与系统测试工程师；

## 通用格式约定

格式	说明
宋体	正文采用宋体表示。
黑体	一级、二级、三级标题采用黑体。
楷体	警告、提示等内容一律用楷体，并且在内容前后增加线条与正文隔离。
Terminal Display 格式	Terminal Display 格式表示屏幕输出信息。此外，屏幕输出信息中夹杂的用户从终端输入的信息采用加粗字体

## 寄存器访问类型约定

类型	说明	类型	说明
RO	只读，不可写。	W0C	可读，写 0 清零，写 1 保持不变。
WO	只写。	W1S	可读，写 1 置 1，写 0 保持不变。
RW	可读可写。	W0S	可读，写 0 置 1，写 1 保持不变。
RC	读清零。	OSW	可读，写 1 后片内自清零，即产生一个脉冲。
W1C	可读，写 1 清零，写 0 保持不变。		

## 数值单位约定

数据容量、频率、数据速率等的表达方式说明如下。

类别	符号	对应的数值
数据容量（如 RAM 容量）	1K	1024
1M	1,048,576	
1G	1,073,741,824	
频率、数据速率等	1k	1000
1M	1,000,000	
1G	1,000,000,000	

地址、数据的表达方式说明如下。

符号	举例	说明
0x	0xFE04、0x18	用 16 进制表示的数据值、地址值。
0b	0b000、0b00 00000000	表示 2 进制的数据值以及 2 进制序列（寄存器描述中除外）

## 修订记录

修订日期	版本	修订说明
2016-10-12	H0008	增加部分功能寄存器设置
2016-04-28	H0007	增加部分功能寄存器设置
2015-12-22	H0006	增加部分功能寄存器设置
2015-02-04	H0005	增加 ICMP 功能
2014-06-09	H0004	增加部分功能寄存器设置
2013-11-02	H0003	增加了 TCP 部分新增规格，包括 keepalive，坚持定时器等；
2013-04-02	H0002	增加 XGMII 接口部分的协议描述； 增加 IGMP 功能描述以及寄存器描述；
2012-12-18	H0001	初稿版本。

# 2 产品概述

RXNoE 是一款专业的网络处理器 IPcore，完全卸载 TCP/UDP/ARP/ICMP/IGMP/IP/MAC 等功能，可应用于网络存储，音视频等终端数据传输，为其提供高带宽，低延迟，高质量的网络数据传输，也可以应用于高频交易（HFT）等金融领域，提供超低延时网络接入以及 TCP 协议栈处理等，同时，可以完全释放因为 TCP/IP 所耗用的 CPU 资源，可替换的 MAC 接口可以适用于当前常见的 10M/100M/1G/10G/40G 等以太网接口，根据具体 FPGA/ASIC，定制开发可支持 256 最大 TCP/UDP 连接数。

## 2.1 功能描述

RXNoE 支持功能如下：

- 支持 16 个 ARP 表项；
- 支持主动建立 ARP 表项；
- 支持主动发送 ARP request；
- 支持响应 ARP Reply，根据 reply 信息自动刷新表项；
- 支持响应 ARP Request，建立表项后自动回复 ARP reply；
- 支持 ARP 表项的主动删除；
- 提供 ARP 表项的查询接口，使用 IP 地址查询对应的 MAC 地址和表项 ID；
  
- 支持被动响应 ping 指令，用于调试网络；
  
- 支持 8 个 IP 组播组；
- 支持根据序号进行组播信息查询；
- 支持根据组播 IP 进行组播详细信息查询；
- 响应 IGMPv1 查询报文；
- 响应 IGMPv2 普遍组查询报文；
- 响应 IGMPv2 特定组查询报文；
- 响应 IGMPv3 普通组查询报文；

- 响应 IGMPv3 特定组查询报文;
- 响应 IGMPv3 特定源查询报文, 最大响应三个源 IP;
- 支持主动报告, 即当初始加入组播组时, 主动发送 IGMPv3 报告报文;
- 支持离开组播组配置接口;
  
- 支持 16TCP 连接, 其中 1 条管理连接, 15 条可应用连接;
- 支持 server 和 client 模式:
  - Server: 被动建立连接;
  - Client: 主动建立连接;
- 支持标准 3 次握手建立连接
- 支持正常的 4 次握手删除以及发送 RST 的强制删除;
- 支持标准 TCP 状态迁移;
- 标准 checksum 计算方式;
- 支持 MSS 选项;
- 支持超时重传与快速重传方式, 重传时间可配, 指数退避策略;
- 支持 RTT 动态计算;
- 依据 TCP 流字节方式重传重排;
- 支持标准的滑动窗口流控机制;
- 支持窗口实时更新与通告;
- 支持 keepalive, 保活时间可配置;
- 支持窗口更新异常情况下的零窗口探查方案;
- 支持 2Gap 条件下的乱序接收, 数据重排;
- 支持发送和接收片内 buffer 缓存, 其中发送 8KB, 接收 16KB;
  
- 支持 16UDP 连接, 其中 1 条管理连接, 15 条可应用连接;
- 支持 MSS 可配置;
- 支持收发 UDP checksum 校验, 开关可配置;
- 支持 IP 组播接收;
- 支持 UDP 广播;
- 不支持 IP 分片接收与发送;

## 2.2 系统结构

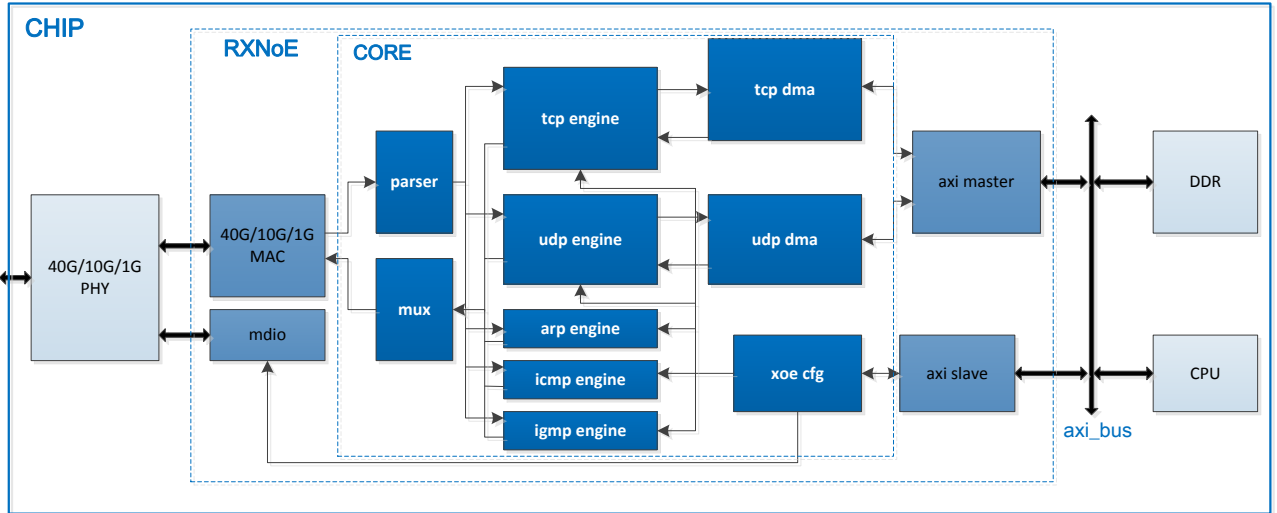


图 2-1 系统集成结构图

如图 2-1 所示系统集成结构图所示，RXNoE 在网络侧提供了 XGMII 接口用于外接万兆 PHY，在用户测试提供了标准的 buffer 接口，根据用户的系统架构不同而接不同的总线系统，图中案例中使用的 ARM AMBA BUS，在数据接口和配置接口均使用的 AXI+DMA 来搬移数据，同样，用户也可以根据需要选择 PCIe 或者存储接口等。

## 2.3 接口描述

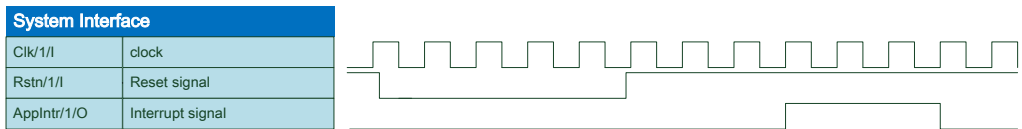


图 2-2 系统接口时序图

**Clk:** 系统工作时钟，整个系统提供唯一的时钟接口，所以要求工作时钟和 MAC 的接口时钟一致，比如 XGMII 接口时钟为 165.25MHz，即使用 PHY 恢复输出时钟作为工作时钟；

**Rstn:** 全局复位信号，低有效，支持同步和异步复位；

**AppIntr:** 用户侧中断信号，RXNOE 中断内容需定制，采用三段中断模式；

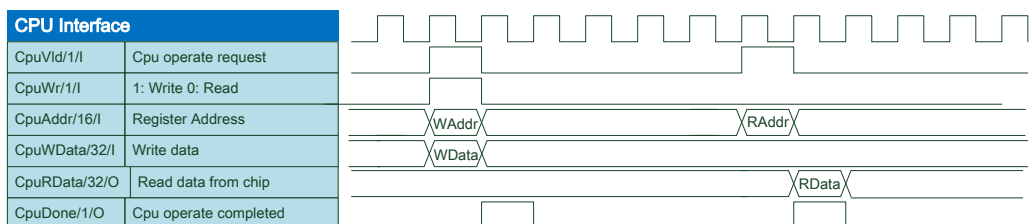


图 2-3 配置接口时序图

配置接口使用常用的 CPU 配置接口信号模式，包括 16bit 地址配置空间。

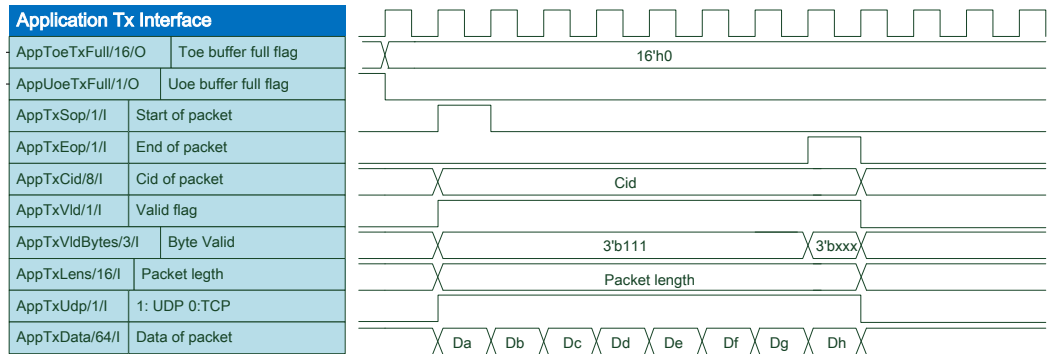


图 2-4 应用层发送接口时序图

应用侧发送接口采用交互握手的方式，其中 TCP 和 UDP 采用独立的 buffer full flag:

**AppToeFull**: 表示 TCP 连接的 full 状态，每一个对应的 bit 表示对应 TCP 连接的状态;

**AppUoeTxFull**: 表示 UDP 共用 buffer 的状态;

TCP 和 UDP 连接采用分离的 buffer，用户可以根据需求在同有 TCP 和 UDP 连接是自行决定发送优先级;

AppTx 数据接口采用数据包络的数据格式，其中:

**AppTxSop**: 报文头，需保证为单拍有效;

**AppTxEop**: 报文尾，需保证为单拍有效;

**AppTxVld**: 报文有效位包络信号，可不连续;

**AppTxVldBytes**: AppTxVld 有效时数据有效 bytes 数，小端数据格式，地位开始有效，其中 3'b000 表示一个 byte 有效，依次累加;

**AppTxLens**: 本次发送的有效 bytes 数;

**AppTxUdp**: UDP 报文标志，高有效;

**AppTxCid**: 发送报文对应的 connection ID，通过 AppTxUdp 判断是选择 TCP/UDP;

**AppTxData**: 64 位数据总线。

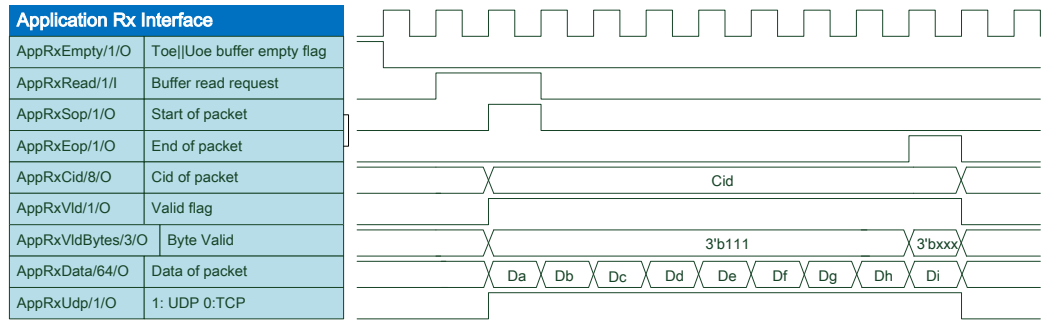


图 2-5 应用层接收接口时序图

应用层接收接口同样采用握手的方式，但是相较于发送接口，这个不需要使用 TCP/UDP 分离的状态接口，采用统一的 AppRxEmpty 接口表示 buffer 状态。

**AppRxEmpty:** 表示当前接收 Buffer 有经过处理过的 payload 数据；

**AppRxRead:** 接收 Buffer 读使能信号，当 AppRxEmpty 为 0 后打拍产生，AppRxRead 与 AppRxSop 形成一对握手信号，当 AppRxRead 拉高后，等待 AppRxSop 有效后 AppRxRead 即可拉低，这里的 AppRxRead 是一个报文级的读使能信号；

接收接口仍然使用包络的数据时序结构，详细信号描述与 AppTx 一致。

## 2.4 集成指导

RXNoE 为客户提供具体要求的应用指导，客户可联系技术支持咨询。

# 3 典型应用

RXNoE 在完善的功能基础上，尽可能的提供了简洁易用的配置接口供用户方便调用，本章将阐述常用的一些功能的基本配置流程和使用注意点，以此来指导用户开发适合用户自身环境的驱动和应用程序。

## 3.1 RXNoE 初始化

初始化主要包括 4 个部分：

### 1. 公共寄存器初初始化配置

本机 MAC/IP 地址配置：

- 配置本机 MAC 地址低 32bit 到 local\_mac\_low 寄存器
- 配置本机 MAC 地址高 16bit 到 local\_mac\_high 寄存器
- 配置本机 IP 地址到 local\_ip 寄存器

本机 IP 头 TTL 和 TOS 设置，有特殊需求配置 ip\_ttl\_tos 寄存器，通常情况下使用默认配置即可。

MAC 使能寄存器：

- 设置 mac\_enable 值为 0x7；

### 2. TCP 寄存器初始化配置

- 不启动动态统计 RTT 状态时初始化配置 RTT 值到 tcp\_rtt\_val 寄存器；
- 配置最大重传次数到 max\_rty\_times 寄存器，默认值 16
- 配置 Max Send Size (MSS) 到 tcp\_mss 寄存器，默认 1460；
- 根据业务配置 TTL 和 TOS 到 tcp\_ttl\_tos 寄存器；
- 非万兆网情况系配置 1ms 统计值到 tick\_1ms\_cnt，统计单位为一个工作时钟时间单位；
- 配置 toe\_enable，根据需要选择对应开关即可。

3. UDP 寄存器初始化配置
  - 初始化 MSS 值到 `udp_mss` 寄存器;
  - 配置 `uoe_enable` 寄存器使能 UOE;
4. IGMP 寄存器初始化配置
  - 根据业务需要配置 `igmp_enable` 为 0x1 即可

注意：用户可根据需要配置中断使能，具体中断功能需根据用户具体需要定制，详细可咨询融信网通技术支持。

## 3.2 TCP 连接建立

TCP 连接建立分为两种 `client` 和 `server` 两种模式，其中 `client` 是由用户发起指令主动建立连接，`server` 是由用户开启连接服务，等待远端来建立连接。

### 3.2.1 client 模式

建立 TCP `client` 连接的过程分为两个过程：

- 检查要建立连接的 `CID` 是否被占用，如果被占用，则换一个 1~15 范围内的 `CID` 继续查询，直到找到可用的 `CID` 为止；
- 获取到可用的 `CID` 后，配置对应的寄存器建立连接。

寄存器配置过程如下：

- 步骤 1：配置写 `conn_stat_rd` 寄存器，低 4bit 即为 `CID`，bit5 读请求置 1；
- 步骤 2：读 `conn_stat_rd`，如果 bit5 为 1，则表明该读操作还未完成，重复步骤 2，否则进入步骤 3；
- 步骤 3：读取 `conn_stat` 寄存器，如果处于 `IDLE (0x0)`，获 `ERROR_STATE (0xd)` 状态，则说明该连接 `CID` 可用，进入步骤 4，否则重复步骤 1；
- 步骤 4：配置 TCP 连接初始化设置：

配置 `tcp_set_rip` 寄存器，设置远端 IP 地址；

配置 `tcp_set_port` 寄存器，设置本地和远端端口号；

配置 `tcp_init_seq` 寄存器，设置连接初始化序号，也可用系统随机值；

- 步骤 5: 使能该 TCP 连接, 配置 `conn_enable` 寄存器, 其中 `bit0` 配置为 1, `bit[7:4]`即为可用的 CID 值;

### 3.2.2 server 模式

建立 TCP server 连接和 client 连接过程一致, 查找可用的 CID 方法和建立 client 连接方式一致, 区别主要是和 client 的步骤 4 和步骤 5 有所区别;

- 在获取到可用 CID 后, 配置 TCP server 连接初始化设置:

配置 `tcp_set_port` 寄存器, 设置监听端口到 `bit[15:0]`;

配置 `tcp_init_seq` 寄存器, 设置连接初始化序号, 也可用系统随机值;

- 使能该 TCP 连接, 配置 `conn_enable` 寄存器, 其中 `bit1` 配置为 1, `bit[7:4]`即为可用的 CID 值;

## 3.3 TCP 连接删除

TCP 连接的删除方式也包括两种, 分为正常删除和强制删除两种:

- 正常删除: 发送 FIN 报文四次握手的方式删除连接, 采用正常删除的条件是连接必须已经建立好, 且当前不在连接删除的状态;
- 强制删除: 发送 RST 强制删除, 这样被删除的连接本地端将强制进入 IDLE 状态, 同时发送 RST 报文强制对端复位;

### 3.3.1 正常删除

连接正常删除也需要分两个过程, 首先查询连接状态, 而后执行删除指令。

- 步骤 1: 配置写 `conn_stat_rd` 寄存器, 低 4bit 即为 CID, `bit5` 读请求置 1;
- 步骤 2: 读 `conn_stat_rd`, 如果 `bit5` 为 1, 则表明该读操作还未完成, 重复步骤 2, 否则进入步骤 3;
- 步骤 3: 读取 `conn_stat` 寄存器, 如果处于 `establish (0x4)`, 或者 `close_wait (0x7)` 状态, 进入步骤 4, 否则可以对该连接执行强制删除或者重复步骤 1 等待 TCP 状态机运行到可删除的状态;
- 步骤 4: 配置连接删除 `conn_delete` 寄存器, 启动 `bit[1:0]=2'b01`, `bit[7:4]`设置为被删除连接的 CID;
- 步骤 5: 因为 TCP 状态机实时运作, 可删除状态会被迁移, 因此需要查看删除指令是否被执行, 直接读取 `conn_delete` 寄存器, 返回值 `bit[0]`位 0 表示指令被响应, 否则重复步骤 5, `bit[0]=0` 时, 判断 `bit[1]==1`, 说明正常删除失败, 则可以重复整个正常删除流程或者直接配置强制删除指令完成连接删除。

### 3.3.2 强制删除

连接强制删除相对正常删除简单，过程如下：

- 步骤 1：配置连接删除 `conn_delete` 寄存器，启动 `bit[1:0]=2'b11`，`bit[7:4]` 设置为被删除连接的 CID；
- 步骤 2：查看删除指令是否被执行，直接读取 `conn_delete` 寄存器，返回值 `bit[0]` 位 0 表示指令被响应，否则重复步骤 2，`bit[0]=0` 时，即可退出删除进程；

## 3.4 TCP 数据收发

TCP 连接的数据收发是和 TCP 连接的状态强相关，在 RXNOE 的内部设计中，使用了本地 TCP 状态机来控制 App 接口的状态信号，以保障 TCP 流程严格执行，所以为了可以正常获取到 TCP 的数据服务，需严格遵守本文 2.3 节定义的接口描述。

## 3.5 UDP 服务

### 3.5.1 UDP Client

UDP Client 是在知悉 UDP server 的 IP 地址和服务端口号之后主动进行数据报文的接收和发送，UDP Client 建立步骤如下：

- 步骤 1：查看 `udp_setup` 寄存器，判断 `bit0` 是否为 0，非零表示当前有建立请求还未响应，重复步骤 1，否则进入步骤 2；
- 步骤 2：配置 UDP Client 信息，包括配置远端 IP 到 `udp_set_rip` 寄存器，配置本地和远端 port 到 `udp_set_port` 寄存器。
- 步骤 3：启动服务，配置对应的 CID 到 `udp_setup` 寄存器 `bit7:4`，低 4bit 配置为 0x1。

### 3.5.2 UDP Server

UDP Server 是在本地开设监听端口号之后 UDP Client 来主动建立数据通路，RXNOE 限制一个 CID 对应的监听端口只允许一个 client 接入，但可以设置相同的监听端口，设置步骤如下：

- 步骤 1：查看 `udp_setup` 寄存器，判断 `bit0` 是否为 0，非零表示当前有建立请求还未响应，重复步骤 1，否则进入步骤 2；
- 步骤 2：配置 UDP server 端口信息，即本地监听 port 到 `udp_set_port` 寄存器低 16bit。
- 步骤 3：启动服务，配置对应的 CID 到 `udp_setup` 寄存器 `bit7:4`，低 4bit 配置为 0x3。

### 3.5.3 删除 UDP 服务

- 配置 `udp_delete` 寄存器对应 CID 位，将清除对应 CID udp 信息，对应的 udp 服务也将停止

### 3.5.4 UDP 广播

udp 广播是针对 `udp server` 的广播业务，即本地发送的 udp 广播数据报，局域网内所有 `client` 端均可接收到，在 `client` 端只需 `port` 匹配即可接收，配置过程如下：

- 步骤 1：查看 `udp_setup` 寄存器，判断 `bit0` 是否为 0，非零表示当前有建立请求还未响应，重复步骤 1，否则进入步骤 2；
- 步骤 2：配置本地和远端 `port` 到 `udp_set_port` 寄存器
- 步骤 3：启动服务，配置对应的 CID 到 `udp_setup` 寄存器 `bit7:4`，低 4bit 配置为 `0x7`。

### 3.5.5 UDP 组播

udp 组播是针对 `udp server` 的组播业务，即本地可以作为组播源，局域网内所有该组播域 `client` 端均可接收到，同样也可以接收对应组播域内的数据报，配置过程如下：

- 步骤 1：查看 `udp_setup` 寄存器，判断 `bit0` 是否为 0，非零表示当前有建立请求还未响应，重复步骤 1，否则进入步骤 2；
- 步骤 2：配置组播 IP 到 `udp_set_rip` 寄存器，配置本地和远端 `port` 到 `udp_set_port` 寄存器。
- 步骤 3：启动服务，配置对应的 CID 到 `udp_setup` 寄存器 `bit7:4`，低 4bit 配置为 `0xb`。

## 3.6 IGMP 服务

### 3.6.1 加入组播域

RXNOE 共支持 8 个组播域，支持采用 IGMPv3 协议加入组播域，最大支持两个源 IP 选择，且支持 INCLUDE 和 EXCLUDE 两种模式，配置过程如下：

- 步骤 1：配置加入的组播 IP 到 `igmp_join_mip` 寄存器：

- 步骤 2: 配置选择的源 IP 到 igmp\_join\_sip0/ igmp\_join\_sip1 寄存器;
- 步骤 3: 配置加入请求命令到 igmp\_join\_req 寄存器, 可以在该寄存器设置有效源 IP 数目和模式;
- 步骤 4: 回读 igmp\_join\_req 寄存器, 到 bit0 为 0 时, 同时查看 bit4, 判断加入对应组播域是否成功。

### 3.6.2 退出组播域

- 欲退出组播域只需将对应的组播 IP 配置到 igmp\_leave\_mip 寄存器即可。

# 4 寄存器描述

## 4.1 寄存器地址分配

RXNOE 共需占用 16bit 地址空间，地址空间分配如下：

地址空间	使用描述
0x0000-0x00FF	公用地址空间
0x0100-0x01FF	UDP 地址空间
0x0200-0x02FF	TCP 地址空间
0x0300-0x03FF	IGMP 地址空间
0x0400-0x04FF	MAC 地址空间
0x5000-0xFFFF	保留

## 4.2 寄存器详细描述

### 4.2.1 公共寄存器

偏移	属性	名称	详细描述	默认值
0x0000	RO	ver_date	版本日期	0x2014_0308
0x000c	RW	local_mac_low	[31:0]本地 MAC 地址低 32bit	0x0
0x0010	RW	local_mac_high	[31:16] 保留 [15:0] 本地 MAC 地址高 16bit	0x0
0x0014	RW	local_ip	[31:0]本地 IP 地址	0x0
0x0018	RW	ip_ttl_tos	[31:16] 保留保留 [15:8] 本地 IP 头 TTL 字段配置 [7:0] 本地 IP 头 TOS 字段配置	0x4000

### 4.2.2 UDP 寄存器

偏移	属性	名称	详细描述	默认值
----	----	----	------	-----

0x0104	RW	uoe_enable	[31:2]保留 [1]UOE checksum 检查使能开关 [0]UOE 使能开关	0x2
0x0108	RW	udp_mss	[31:16]保留 [15:0]UDP MSS 配置，该设置适用于所有 UDP 连接	0x5b4
0x010c	RW	udp_tab_rd	[31:4]保留 [3:0]读取对应 UDP 连接表项的 CID	0x0
0x0110	RO	udp_tab_data0	[31:0] 回读的 UDP 连接表项信息，该表项信息为远端 MAC 地址的低 32bit；	0x0
0x0114	RO	udp_tab_data1	回读的 UDP 连接信息 1 [31:16] 回读的 UDP 连接表项信息，远端 IP 地址低 16bit； [15:0] 回读的 UDP 连接表项信息，远端 MAC 地址高 16bit；	0x0
0x0118	RO	udp_tab_data2	[31:16] 回读的 UDP 连接信息远端口号； [15:0] 回读的 UDP 连接表项信息，远端 IP 地址高 16bit；	0x0
0x011c	RO	udp_tab_data3	回读的 UDP 连接信息 3 [31:19] 保留 [18] server 模式等待对端接入，补充完整连接信息； [17] client 模式，等待 ARP 回应状态； [16] UDP 连接已就绪 [15:0] 回读的 UDP 连接信息本地端口号；	0x0
0x0150	RW	udp_set_rip	[31:0]建立 UDP 连接时需配置的远端 IP 地址；	0x0
0x0154	RW	udp_set_port	[31:16]建立 UDP 连接时需配置的远端口号； [15:0]建立 UDP 连接时需配置的本地端口号；	0x0
0x0158	RW	udp_setup	[31:8]保留 [7:4]建立 UDP 连接指定的 CID； [3]建立 UDP 组播发包连接，设立为 server 模式； [2]建立 UDP 广播发包连接，设立为 server 模式； [1]建立连接类型 1：server 模式； 0：client 模式 [0]连接建立请求，配置为 1，自动清零；	0x0

0x015c	RW	udp_delete	[31:8]保留； [7:4]删除指定 UDP 连接的 CID； [3:1]保留； [0]连接删除请求，配置为 1，自动清零	0x0
--------	----	------------	---	-----

### 4.2.3 TCP 寄存器

偏移	属性	名称	详细描述	默认值
0x0204	RW	toe_enable	[31:10]保留	0x3f0
			[11] 坚持定时器使能开关，默认情况下不使能	
			[10] 保留；	
			[9]保活定时器使能开关，默认情况下使能打开；	
			[8] RTT 动态统计使能： 1:根据网络延时自动计算； 0:使用系统默认配置的 RTT 值；	
			[7] 超时全部重传模式使能开关： 1:发生超时重传时，会关闭发送 buffer，将 buffer 内所有数据重传输出； 0:发送超时重传时，只发送一个重传报文；	
			[6] 快速重传使能开关，高有效，默认 1 使能	
			[5] 半关闭默认退出模式使能开关，默认 1 使能： 1:在接收到远端主动发送的 FIN 报文时，会自动关闭发送通道，待缓存数据发送完毕后，自动发送 FIN_ACK； 0:在接收到远端主动发送的 FIN 报文时，RXNOE 不会自动发送 FIN_ACK，而是发送 ACK 报文，进入半关闭状态，可上报中断，通告 CPU 关闭连接；	
[4] TCP 接收校验和使能开关，默认 1 使能；				

			[3] 保留；	
			[2] TCP 随机初始化序号模式使能，默认 1 使能；	
			[1]保留；	
			[0] TOE 模块使能开关；	
0x0208	RW	tcp_mss	[31:16] 保留 [15:0]TCP MSS 配置；	0x5b4
0x0218	RW	tcp_ttl_tos	[31:16] 保留 [15:8] 针对 TCP 连接使用的 TTL 配置； [7:0] 针对 TCP 连接使用的 TOS 配置；	0x0
0x021c	RW	conn_enable	[31:8] 保留； [7:4]指定建立 TCP 连接的 CID； [3] 保留； [2] 连接建立失败回显标志； [1]建立连接请求，本地用做 server，开辟监听端口，指令响应后自动清零 [0]建立连接请求，本地为 client，指令响应后自动清零	0x0
0x0220	RW	tcp_set_rip	[31:0]建立 TCP 连接时配置的远端 IP 地址；	0x0
0x0224	RW	tcp_set_port	[31:16]建立 TCP 连接时配置的远端端口号； [15:0]建立 TCP 连接时配置的本地端口号；	0x0
0x0228	RW	tcp_init_seq	[31:0]建立 TCP 连接时初始序号，当初始化模式时有效；	0x0
0x022c	RW	conn_stat_rd	[31:5] 保留； [4] 查询请求，当查询成功后该请求自动清零，即可用回读查询结果；[3:0] 查询对应 TCP 连接的 CID；	0x0
0x0230	RO	conn_stat	[31:8] 保留 [7:0] 查询当前连接状态机状态，状态编号如下：  8'h1 ARP request has send; 8'h2 ARP reply received/CLOSE state  8'h3 SYN_SENT  8'h4 ESTABLISHED  8'h5 LISTEN	0x0

			8'h6 SYN_RECEIVED 8'h7 CLOSE_WAIT 8'h8 LAST_ACK 8'h9 FIN_WAIT_1 8'ha FIN_WATI_2 8'hb CLOSING 8'hc TIME_WAIT 8'hd ERROR_STATE	
0x0234	RW	conn_delete	[31:8] 保留； [7:4]指定删除连接的 CID [3] 保留； [2] 连接删除失败状态标志； [1]删除连接类型： 1：强制删除，发送 RST； 0：正常四次握手删除 [0]删除连接请求，删除完成后该请求会自动清零；	0x0
0x0238	RW	tcp_rtt_val	[31:0] 系统初始化 RTT 值；	0x64
0x023c	RW	max_rty_times	[31:8] 保留 [7:0]最大重传次数配置，当重传次数达到该设置后该 TCP 连接将会强制删除；	0xf
0x0240	RW	max_dup_ack	[31:8] 保留； [7:0]触发快速重传的重复 ACK 数目	0x3
0x0244	RW	tick_1ms_cnt	[31:0] 1ms 定时器计数值	0x2625a
0x0248	RW	conn_wnd_size	[31:16] 保留； [15:0] 初始化本地 window size	0x2000
0x024c	RO	keepalive_cfg	[31:24] keepalive 最大重传次数，默认值为 10； [23:16] keepalive 重传间隔，单位 s，默认值为 75s； [15:0] keepalive 启动时间，单位 s，默认值为 7200s ( 2h )	0x0a4b1c20
0x0290	RW	lu_tab_rd	[31:4] 保留； [3:0] 请求读取连接查找表的连接 CID	0x0
0x0294	RO	lu_tab_rdata0	[31:16] 连接查找表读取后获得的远端端口号； [15:0] 连接查找表读取后获得的本地端口号；	0x0
0x0298	RO	lu_tab_rdata1	[31:0] 连接查找表读取后获得的远端 IP 地	0x0

			址；	
0x02a0	RW	conn_tab_rd	[31:9] 保留； [8] TCP 连接管理表项读请求，成功后自动清零； [7] 保留； [6:4] 请求读取对于 CID 管理表项的索引值； [3:0] 请求读取 TCP 连接管理表项的 CID 值；	0x0
0x02a4	RO	conn_tab_rdata0	[31:0] 连接管理表读取结果 0；	0x0
0x02a8	RO	conn_tab_rdata1	[31:0] 连接管理表读取结果 1；	0x0

#### 4.2.4 IGMP 寄存器

偏移	属性	名称	详细描述	默认值
0x0300	RW	igmp_enable	[31:1]保留 [0]IGMP 使能开关；	0x0
0x0304	RW	igmp_join_mip	[31:0] 指定加入的组播组 IP 地址；	0x0
0x0308	RW	igmp_join_sip0	[31:0] 指定接收组播源的 IP 地址 0	0x0
0x030c	RW	igmp_join_sip1	[31:0] 指定接收组播源的 IP 地址 1	0x0
0x0310	RW	igmp_join_req	[31:12]保留 [11]表示 igmp_join_sip0 寄存器有效，即根据 bit10 判断是否接收该源 IP 组播报文； [10]表示指定源模式 1：INCLUDE 0：EXCLUDE [9] 表示 igmp_join_sip1 寄存器有效，即根据 bit10 判断是否接收该源 IP 组播报文； [8:5]保留； [4] join_success，回读该寄存器，当 join_req 清零后，该 bit1 表示加入组播成功，否则失败； [3:1]保留； [0]join_req，请求加入组播域是置 1，响应后自动清零；	0x0
0x0314	RW	igmp_leave_mip	[31:0] 请求退出的组播域 IP，执行配置后自动退出；	0x0
0x0318	RW	igmp_tab_rd_id	[31:4]保留 [3:0]请求读取 IGMP 组播表项的对应的表项 ID；	0x0
0x031c	RO	igmp_tab_rd0	[31:0] 回读 igmp table 表项 sip1	0x0
0x0320	RO	igmp_tab_rd1	[31:0] 回读 igmp table 表项 sip0	0x0

0x0324	RO	igmp_tab_rd2	[31:0] 回读 igmp table 表项 mip	0x0
0x0328	RO	igmp_tab_rd3	[31:6] 保留 [5]表项有效标识 ; [4]sip0 有效 ; [3]源 IP 指定模式 ; [2]sip1 有效 ; [1:0] 保留	0x0
0x032c	RO	igmp_tab_stat	[4*N+3]第 N 个 table 的有效 [4*N+2]第 N 个 table 的 sip0 有效 [4*N+1]第 N 个 table 的指定源 IP 模式 [4*N]第 N 个 table 的 sip1 有效	0x0

## 4.2.5 MAC 寄存器

偏移	属性	名称	详细描述	默认值
0x0400	RW	mac_enable	[31:3]保留 [2]MAC 接收 CRC 校验使能开关 ; [1]MAC 接收使能开关 ; [0]MAC 发送使能开关 ;	0x0
0x0404	RW	mac_rxbuf_th	[31:16] 保留 [15:0] MAC 接收 buffer 启动阈值	0x73e
0x0408	RW	mac_cnt_clr	[31:1] 保留 [0] MAC 统计计数器清零 ;	0x0
0x0410	RO	mac_rxin_cnt	[31:16] MAC 接收 buffer 输入报文统计计数器 ;	0x0
0x0414	RO	mac_rxout_cnt	[31:16] MAC 接收 buffer 输出报文统计计数器 ;	0x0
0x0418	RO	mac_rxdrop_cnt	[31:16] MAC 接收方向丢弃报文统计计数器 ;	0x0
0x041c	RO	mac_crcerr_cnt	[31:16] MAC 接收方向 crc 校验错误报文统计计数器 ;	0x0