

MAC 使用手册

南京博芯电子技术有限公司

2009-04

This document contains information on a product under development. Prochip Corp reserves the right to change or discontinue this product without notice.
Prochip Corp, 2009. All rights reserved.

版权说明

版权所有，未经南京博芯信息技术有限公司的授权，本说明文档不可以被复制或以其他方式（电子的或是机械的）传播，包括影印，记录或是用其他任何信息存储及检索系统。文档所描述的任何一种电路对于第三方没有专利权及专利特许权。

否认书：

南京博芯信息技术有限公司保留对文档随时进行修改的权利，无须任何申明。南京博芯信息技术有限公司所提供的信息是精确可靠的。对于它的应用以及由于应用而导致违反专利权或是第三方的其他权利，本公司不负任何责任。

版本历史

| 日期 | 版本 | 描述 | 备注 |
|---------|-----|----|----|
| 2009-04 | 1.0 | 初稿 | 蔡浩 |
| | | | |

目 录

| | |
|----------------------------|----|
| 版本历史..... | 2 |
| 一. MAC在SEP4020 中的位置 | 4 |
| 二. MAC介绍..... | 4 |
| 2.1 功能介绍 | 4 |
| 2.1.1 概述 | 4 |
| 2.1.2 功能综述 | 5 |
| 2.1.3 工作模式 | 5 |
| 2.1.4 帧格式..... | 6 |
| 2.1.5 PHY管理..... | 6 |
| 2.2 寄存器介绍 | 6 |
| 三. 实现原理..... | 7 |
| 3.1 硬件原理 | 8 |
| 3.1.1 接口定义 (DM9161E) | 8 |
| 3.1.2 接口信号时序..... | 9 |
| 3.2 软件原理 | 10 |
| 3.2.1 头文件定义说明..... | 10 |
| 3.2.2 核心数据结构声明..... | 10 |
| 3.2.3 初始化MAC的配置..... | 11 |
| 3.2.4 代码实现流程图..... | 11 |
| 3.2.5 主要函数及参数，返回值介绍..... | 11 |
| 四. 测试说明..... | 12 |
| 4.1 测试流程..... | 12 |
| 4.2 结果说明..... | 13 |
| 五. 其他注意事项..... | 13 |

一. MAC 在 SEP4020 中的位置

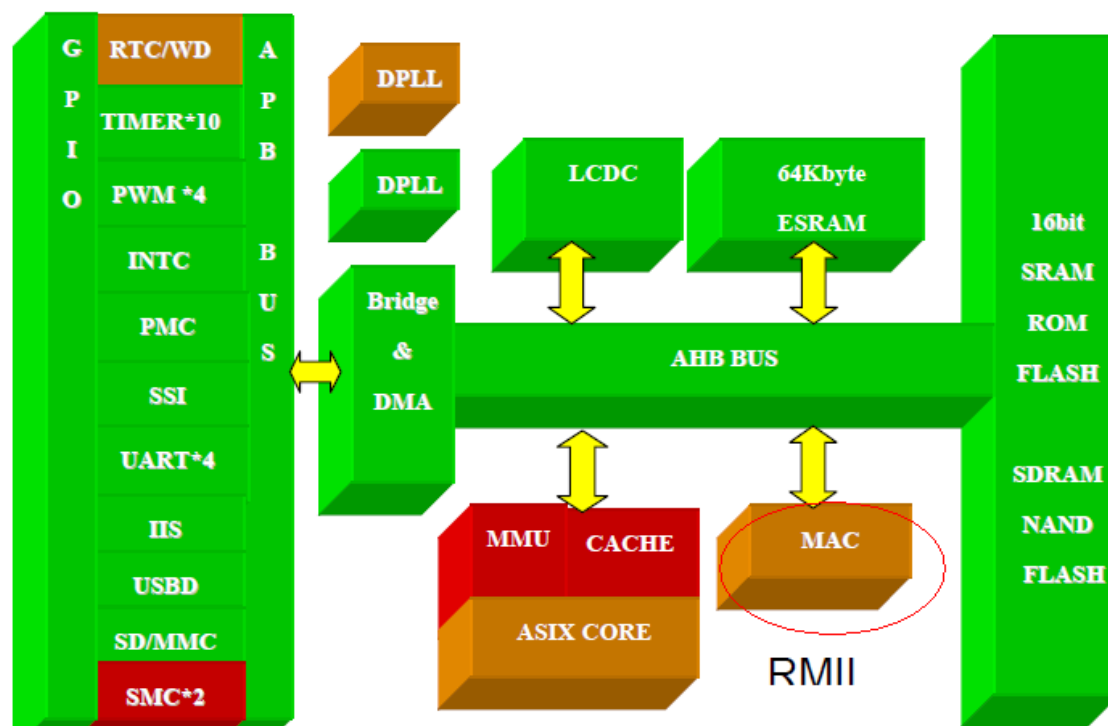


图 1—1 MAC 在 SEP4020 整体中的位置

二. MAC 介绍

2.1 功能介绍

2.1.1 概述

随着技术的发展，嵌入式微处理器实现了高度整合，越来越多的功能模块集成在一个芯片中。此时，各个微处理器之间的通信也越来越迫切。在有效利用串口、并口等数据接口的同时，如何实现更广泛的以太网内数据传输成为设计的重点。为了实现这个目的，引进 ETHERNET MAC 控制器，以实现微处理器在以太网或局域网内的通信功能。

在标准的 ISO 七层网络模型中，MAC 属于数据链路层，MAC 通过物理层 PHY 可以和其他网络设备通信。ETHERNET MAC（以下简称 MAC）是 IEEE 802.3 工作组规定的以太网设备访问以太网网络并进行数据通信的方式。以太网上传的数据以帧为单位，帧格式由 IEEE 802.3 协议规定。

本芯片中，ETHERNET MAC 控制器与 AMBA AHB 总线连接，支持 10/100Mbps 以太网应用，同时需要一个外接 PHY 组成完整地以太网应用方案。

使用思博伦公司的 SmartBits 6000 测试，Nucleus 上 TCP/IP 协议栈实际吞吐率为 1400PPS

(128字节的Ping包)，MAC单向发包的速度12000PPS以上。Linux2.6.16环境下，实际测试TFTP下载速度250K Bytes/Second以上。以上测试基于UB4020EVB (V1.4) 评估板，SEP4020工作在88M Hz，正常工作环境，LCDC使能。

2.1.2 功能综述

兼容IEEE 802.3和802.3u标准，支持10/100M自适应以太网

支持半双工/全双工操作

仅支持RMII接口

自动CRC填充和校验

自动抛弃错误帧

支持网络监听

支持物理层PHY管理

支持全双工流控

支持硬件流控

支持超短帧丢弃，支持超长帧自动截取

支持地址过滤和网络监听

兼容AMBA 规范 2.0

AHB MASTER接口和AHB SLAVE接口

发送接收独立的32×8 FIFO

2.1.3 工作模式

IEEE802.3 协议规定了以太网数据帧在网络上传输的方式，MAC 控制器通过对数据包的封装，包括传输前的帧组装和在帧接收中的分析和接收后的检测；媒介访问控制，包括帧传输初始化和帧传输失败的恢复来实现数据链路层的功能。

MAC 控制器支持全双工、半双工模式下的 10Mbps 和 100Mbps 自适应传输，同时支持独立地址、多地址和广播地址的地址模式，可配置实现数据帧和控制帧的接受和响应。

以独立地址的帧收发为例：

帧发送：当系统有数据需要发送时，首先通过 AHB 总线将发送数据传送给 MAC，MAC 将数据按照以太网数据帧的格式打包，主要是加上 PREAMBLE、SFD 和 CRC 校验，然后按照 IEEE 802.3 协议以 Nibble (4bits, 半字节) 的格式将数据发送给 PHY，PHY 在收到数据后转换成模拟差分信号送到以太网网络。

帧接收：当 MAC 收到从 PHY 来的数据后，先检查目的地址和本机地址是否匹配，如果一

致，将数据接收到 FIFO，如果错误，则将帧丢弃。最后检查数据的 CRC 是否正确。

2.1.4 帧格式

10/100 Mbps 以太网中的基本 IEEE 802.3 MAC 数据格式如下：

| | | | | | | | |
|-----|-----|-----|----|----|-------------|-----------------|-----|
| 字节数 | 7 | 1 | 6 | 6 | 2 | 46-1500 | 4 |
| 内容 | Pre | SFD | DA | SA | Length Type | Data unit + pad | FCS |

表格 9-1 以太网帧格式

Preamble (Pre)：7 字节。Pre 字段中 1 和 0 交互使用，接收站通过该字段知道导入帧，并且该字段提供了同步化接收物理层帧接收部分和导入比特流的方法。

Start-of-Frame Delimiter (SFD)：1 字节。字段中 1 和 0 交互使用，结尾是两个连续的 1，表示下一位是利用目的地址的重复使用字节的重复使用位。

Destination Address (DA)：6 字节。DA 字段用于识别需要接收帧的站。

Source Addresses (SA)：6 字节。SA 字段用于识别发送帧的站。

Length/Type：2 字节。如果是采用可选格式组成帧结构时，该字段既表示包含在帧数据字段中的 MAC 客户机数据大小，也表示帧类型 ID。

Data：是一组 n ($46 \leq n \leq 1500$) 字节的任意值序列。帧总值最小为 64 字节。

Frame Check Sequence (FCS)：4 字节。该序列包括 32 位的循环冗余校验 (CRC) 值，由发送 MAC 方生成，通过接收 MAC 方进行计算得出以校验被破坏的帧。

2.1.5 PHY 管理

通过简单的串行单数据接口，实现了 MAC 对 PHY 的配置和状态读取。本芯片的 MAC 控制器可以实现对 PHY 的 3 项命令支持。

写控制字（将控制字写到 PHY 的配置寄存器）

读状态（读 PHY 的控制和状态寄存器）

扫描状态（持续的读 PHY 的状态寄存器）

（详细的 PHY 配置请参考相应 PHY 的用户手册）

注意：当操作在进行的时候，BUSY 状态始终是置高，直到操作完成才会撤消。

2.2 寄存器介绍

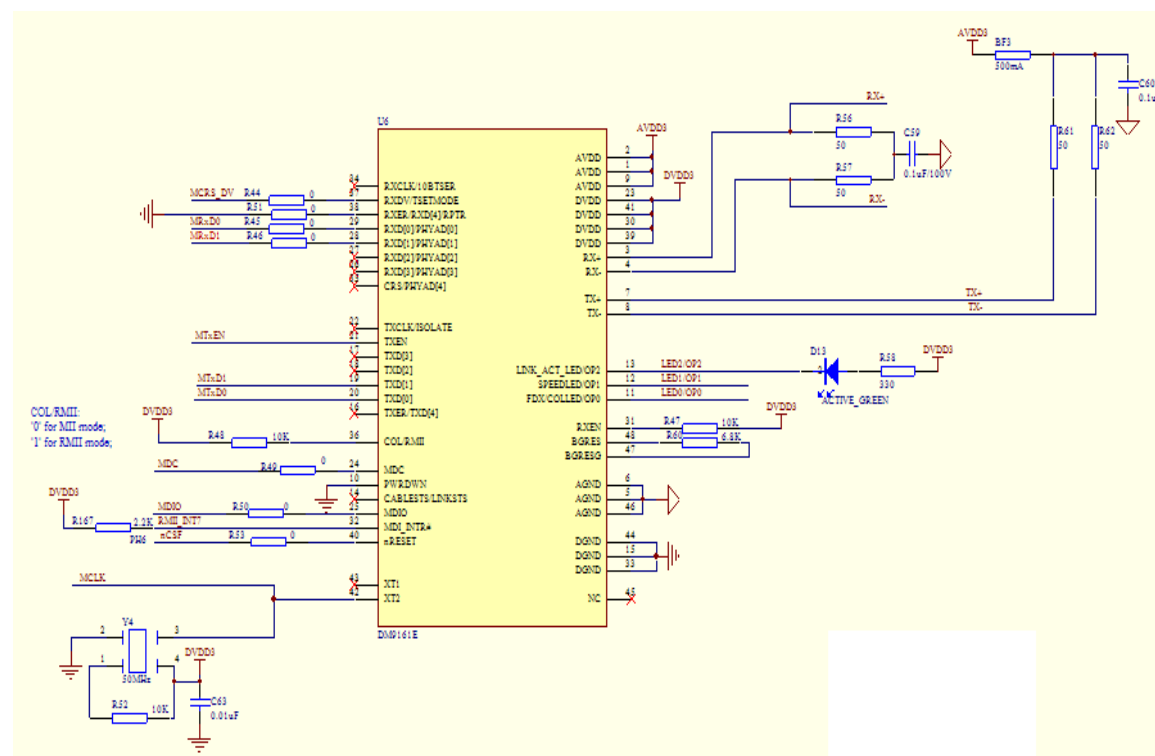
MAC 模块的基址：0x11003000

| 寄存器名 | 偏移地址 | 复位值 | 描述 |
|---------------|-------------|------------|------------------|
| MAC_CTRL | 0x00 | 0x0000A000 | MAC控制寄存器。 |
| MAC_INTSRC | 0x04 | 0x00000000 | MAC中断源寄存器 |
| MAC_INTMASK | 0x08 | 0x00000000 | MAC中断屏蔽寄存器 |
| MAC_IPGT | 0x0C | 0x00000012 | 连续帧间隔寄存器 |
| MAC_IPGR1 | 0x10 | 0x0000000C | 等待窗口寄存器 |
| MAC_IPGR2 | 0x14 | 0x00000012 | 等待窗口寄存器 |
| MAC_PACKETLEN | 0x18 | 0x00400600 | 帧长度寄存器 |
| MAC_COLLCONF | 0x1C | 0x000F003F | 碰撞重发寄存器 |
| MAC_TXBD_NUM | 0x20 | 0x00000040 | 发送描述符数目寄存器 |
| MAC_FLOWCTRL | 0x24 | 0x00000000 | 流控寄存器 |
| MII_CTRL | 0x28 | 0x00000064 | PHY控制寄存器 |
| MII_CMD | 0x2C | 0x00000000 | PHY命令寄存器 |
| MII_ADDRESS | 0x30 | 0x00000000 | PHY地址寄存器 |
| MII_TXDATA | 0x34 | 0x00000000 | PHY写数据寄存器 |
| MII_RXDATA | 0x38 | 0x00000000 | PHY读数据寄存器 |
| MII_STATUS | 0x3C | 0x00000000 | PHY状态寄存器 |
| MAC_ADDR0 | 0x40 | 0x00000000 | MAC地址寄存器 |
| MAC_ADDR1 | 0x44 | 0x00000000 | MAC地址寄存器 |
| MAC_HASH0 | 0x48 | 0x00000000 | MAC HASH寄存器 |
| MAC_HASH1 | 0x4C | 0x00000000 | MAC HASH寄存器 |
| MAC_TXPAUSE | 0x50 | 0x00000000 | MAC控制帧寄存器 |
| MAC_BD | 0x400~0x7FF | | 描述符，共128个，处于RAM区 |

三. 实现原理

3.1 硬件原理

3.1.1 接口定义 (DM9161E)



3.1.1.1 SEP4020 MAC部分管脚定义

| MAC (9) | | | | | | |
|---------|-----------|-----|-----------|---------------|----|------|
| 序号 | 管脚名 | 方向 | 描述 | 驱动 电流 (mA) | 属性 | 复位值 |
| 167 | MCLK | I | 50MHz时钟 | 4 | | — |
| 168:169 | MTXD[1:0] | O | 发送数据 | 4 | | 1'H0 |
| 170 | MTXEN | O | 发送数据使能 | 4 | | 1'H0 |
| 171 | MCRS_DV | I | 接收数据Valid | 4 | | — |
| 173:174 | MRXD[1:0] | I | 接收数据 | 4 | | — |
| 175 | MDC | O | 管理接口时钟 | 4 | | 1'H0 |
| 176 | MDIO | I/O | 双向管理接口 | 4 | | — |

3.1.1.2 GPIO模块MAC接口信号（管脚复用情况）

| 管脚名称 | 方向 | 缺省功能 | 复用功能1 | 复用功能2 | 管脚名 |
|------|----|------|-------|-------|-----|
|------|----|------|-------|-------|-----|

| | | | | | |
|---------|-----|----------------|------|--|---|
| | | | | | 称 |
| PORT I | | | | | |
| MCLK | I | MAC时钟 | GPI8 | | |
| MTXD1 | O | MAC发送数据第1位 | GPI7 | | |
| MTXD0 | O | MAC发送数据第0位 | GPI6 | | |
| MTXEN | O | MAC发送使能 | GPI5 | | |
| MCRS_DV | I | MAC载波信号及接收有效 | GPI4 | | |
| MRXD1 | I | MAC接收数据第1位 | GPI3 | | |
| MRXD0 | I | MAC接收数据第0位 | GPI2 | | |
| MDC | O | MAC到PHY的时钟输出 | GPI1 | | |
| MDIO | I/O | MAC到PHY的输入输出数据 | GPI0 | | |

3.1.2 接口信号时序

MAC的数据接收和发送：

MAC在RX_CLK或TX_CLK上升沿读取数据。

当RX_EN有效时，MAC开始接受数据。当发生接受错误时，RX_ER置位并引起相应的中断响应。

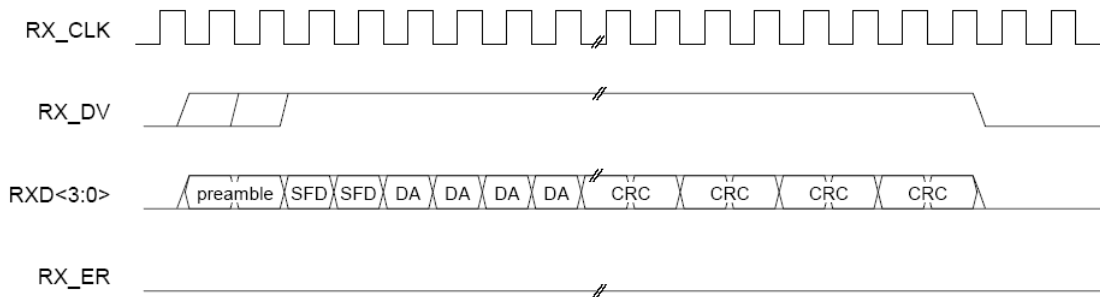


图 3-1 正常接收帧

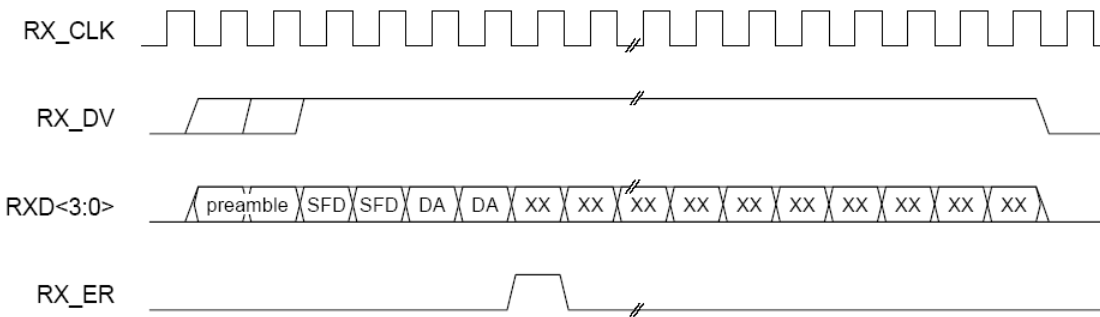


图 3-2 接收帧时发生接收错误

当TX_EN有效时，MAC开始发送数据。当检测到碰撞发生时，COL置位，MAC作出碰撞响应。

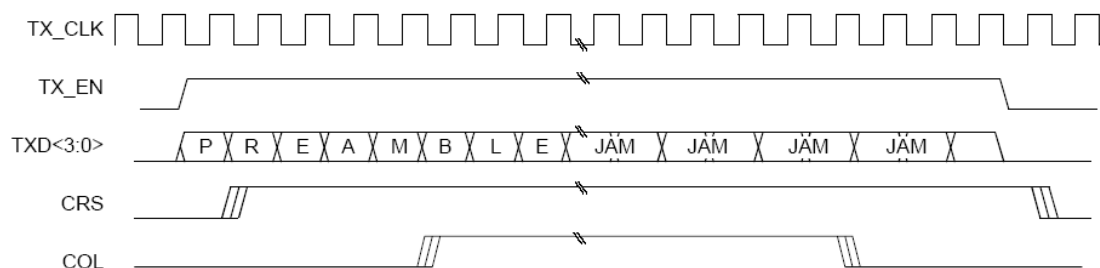


图 3-3 发送帧时发生碰撞

MAC对MII的读写时序

在MDC的上升沿读取MDI或MD0的信号值。完整的MII读写帧包括Preamble、ST、操作码、PHY地址、寄存器地址和数据。

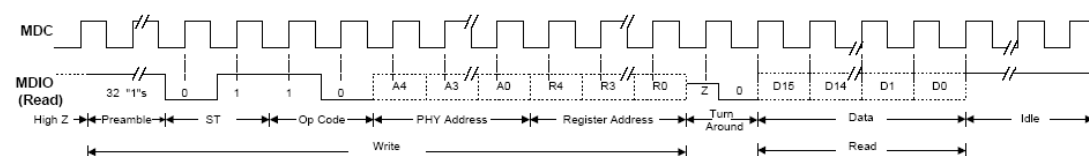


图 3-4 MII读取数据时序图

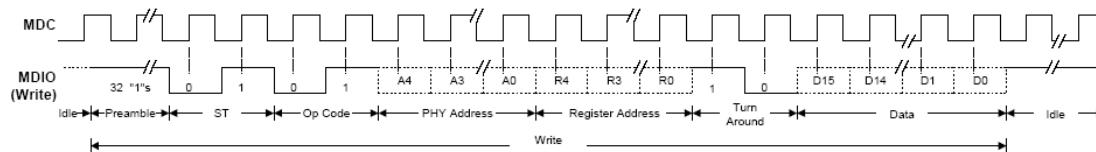


图 3-5 MII写数据时序图

3.2 软件原理

3.2.1 头文件定义说明

| | | |
|----------|----------------|-----------------------------|
| #include | <stdio.h> | 标准输入输出函数库 |
| #include | <stdlib.h> | C语言标准库函数的定义 |
| #include | <string.h> | 字符串头文件 |
| #include | "intc.h" | INTC 模块的中断源，中断处理，定义中断的向量结构体 |
| #include | "HA_typedef.h" | 代码中用到的typedef |
| #include | "hardware.h" | PMU module clk以及读写函数定义 |
| #include | "sep4020.h" | 各个寄存器地址的定义 |

3.2.2 核心数据结构声明

INT_VECTOR 中断向量结构体包括：中断号和中断处理函数
 p[i] 一个MAC帧，前面6个字节为目的帧的MAC地址，后六个字节为发送MAC的地址，接着为数据

3.2.3 初始化MAC的配置

配置MII寄存器，用于配置PHY的工作模式，包括传输速率（10M/100M）、全/半双工模式、自适应、LED配置；

初始化所有接收和发送描述符寄存器，最好先把所有的BD使能置0。

配置MAC地址；

配置中断寄存器；

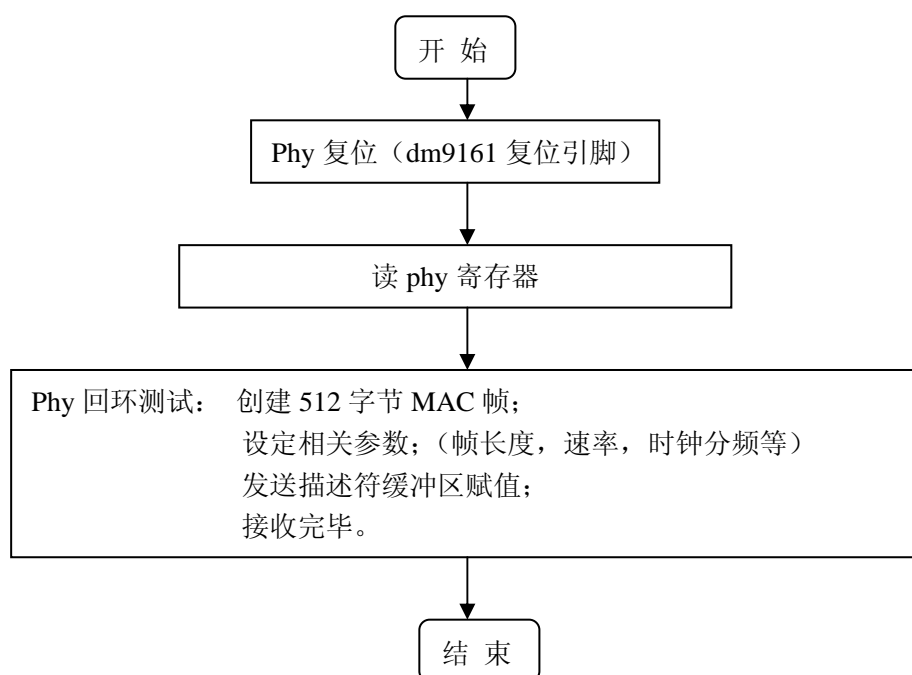
配置描述符个数的分配；

为接收配置相应的描述符寄存器；

配置MAC控制寄存器，MAC开始工作；

当有数据要发送时，配置发送描述符，描述符的地址指针指向需要发送的数据地址

3.2.4 代码实现流程图



3.2.5 主要函数及参数，返回值介绍

- 函数名称: void packet_creat(U32 len)
入口参数: U32 len
返回值: 无
函数功能: 该程序是为发送数据造一个MAC帧, 前面6个字节为目的帧的MAC地址, 后六个字节为发送MAC的地址, 接着为数据。

- 函数名称: void phy_hard_reset(int mii_hard_rst)
入口参数: int mii_hard_rst
返回值: 无
函数功能: phy复位

- 函数名称: void mii_reg_read()
入口参数: 无
返回值: 无
函数功能: 读PHY寄存器

- 函数名称: void write_mii_reg(U16 address, U16 data)
入口参数: U16 address, PHY寄存器地址
 U16 data, 待写的数据
返回值: 无
函数功能: 写PHY寄存器

- 函数名称: void phy_selfloop_back()
入口参数: 无
返回值: 无
函数功能: phy回环测试

四. 测试说明

4.1测试流程

在ADS下面运行代码, 测试PHY回环。

4. 2结果说明

代码运行结果:

PHY lookback ok,MAC test allright!!!!

五. 其他注意事项

- 关于 MAC 的发送描述符和接收描述符的使用方法

发送和接收描述符默认都是 64 个，这 64 个描述符实际上是一个环形队列形式的。

对于发送描述符，MAC 控制器在发送数据包的时候是按顺序填充第 0~63 个发送描述符的，当第 63 个描述符填完后，就会回到第 0 个发送描述符的（但第 63 个发送描述符的 MAC_TX_BD 寄存器的第 13 位一定要置位才行），程序实现是必须这样，否则就会出现尽管有空闲描述符，但还是数据包无法发送出去的现象。

对于接收描述符也是一样，MAC 控制器在接收数据包的时候是按顺序填充第 0~63 个接收描述符的，所以在中断接收程序中，也得从前往后依次接收才行的。

在初始化过程中，一定要注意接收描述符前面第 0~62 个描述符的低 32 位置为 0xc000；第 63 个描述符的低 32 位置为 0xe000，发送描述符前面第 0~62 个描述符的低 32 位置为 0xd800；第 63 个描述符的低 32 位置为 0xf800。

- 关于 MAC 发送描述符和接收描述符的数量

SEP4020 的 MAC 可以任意指定发送描述符和接收符的数量。发送描述符的数量可以通过 MAC_TXBD_NUM 寄存器设置，推荐设定为 1-4 个发送描述符，每个描述符指定一块 1500 Bytes 的发送缓冲区，共占用 1500 Bytes-6000 Bytes 的内部 ESRAM；接收描述符从最后一个发送描述符的下一个描述符开始，每个指向一个 1500 Bytes 大小的缓冲区。接收描述符最多可有 128-MAC_TXBD_NUM 个，可以通过接收描述符的第 13 位（WRAP 位）设定接收描述符的数量。如，只需要 64 个接收描述符，则可以将第 64 个接收描述符的 WRAP 位置 1，MAC 在接收数据时如果检测到该位，则回卷到第一个接收描述符。

- 关于同时使用 LCDC 与 MAC 时需要注意的问题

在 SEP4020 内部设计了 64KB 的 ESRAM，默认地址：0x04000000-0x04010000。该 RAM 空间具有零等待、访问速度快等优点。

需要注意的是，在同时使用 LCDC 和 MAC 的时候，必须将 MAC 发送描述符指向的发送缓

缓冲区设置在内部的 ESRAM 中。即必须将 MAC 发送描述符的高 32 位设置为 0x04000000 开始的地址空间。否则可能导致数据错误，甚至 CPU 出错。接收描述符与 LCDC 缓冲的设置可以根据具体应用灵活设置，可以指向内部 ESRAM，也可以指向外部的 SDRAM。

这一点仅针对于需要同时使用 LCDC 和 MAC 的情况。如果只需要使用 LCDC 或 MAC，对发送和接收缓冲区则没有特殊要求，可以将发送和接收缓冲区同时设置为外部的 SDRAM。

- **关于网络部分的测试数据和测试仪器的问题**

使用思博伦公司的 SmartBits6000 测试，Nucleus 上 TCP/IP 协议栈实际吞吐率为 1400PPS（128 字节的 Ping 包），MAC 单向发包的速度 12000PPS 以上。Linux 实际测试 TFTP 下载速度 250K Bytes/Second 以上。以上测试均为正常使用环境，LCDC 使能。