

# 用 51 单片机控制 RTL8019AS 实现以太网通讯

**摘要:** 介绍以太网的帧协议和以太网控制芯片 RTL8019AS 的结构特性; 介绍 51 单片机控制 RTL8019AS 实现以太网通讯的硬件设计方案; 采用 C51 语言实现 ARP 协议 (地址解析协议), 并进行了系统的调试与验证。

**关键词:** RTL8019AS Ethernet 51 单片机 TCP/IP 协议 滤波器 YL18-1001D

互联网络硬件、软件的迅猛发展, 使得网络用户呈指数增长, 在使用计算机进行网络互联的同时, 各种家电设备、仪器仪表以及工业生产中的数据采集与控制设备在逐步地走向网络化, 以便共享网络中庞大的信息资源。在电子设备日趋网络化的背景下, 利用廉价的 51 单片机来控制 RTL8019AS 实现以太网通讯具有十分重要的意义。

## 1 以太网 (Ethernet) 协议

一个标准的以太网物理传输帧由七部分组成 (如表 1 所示, 单位: 字节)。

表 1 以太网的物理传输帧结构表

PR	SD	DA	SA	TYPE	DATA	FCS
同步位	分隔位	目的地址	源地址	类型字段	数据段	帧校验序列
7	1	6	6	2	46~1500	4

除了数据段的长度不定外, 其他部分的长度固定不变。数据段为 46~1500 字节。以太网规定整个传输包的最大长度不能超过 1514 字节 (14 字节为 DA、SA、TYPE), 最小不能小于 60 字节。除去 DA、SA、TYPE 14 字节, 还必须传输 46 字节的数据, 当数据段的数据不足 46 字节时需填充, 填充字符的个数不包括在长度字段里; 超过 1500 字节时, 需拆成多个帧传送。事实上, 发送数据时, PR、SD、FCS 及填充字段这几个数据段由以太网控制器自动产生; 而接收数据时, PR、SD 被跳过, 控制器一旦检测到有效的前序字段 (即 PR、SD), 就认为接收数据开始。

## 2 RTL8019AS 以太网控制器简介

由台湾 Realtek 公司生产的 RTL8019AS 以太网控制器, 由于其优良的性能、低兼的价格, 使其在市场上 10Mbps 网卡中占有相当的比例。

### 2.1 主要性能

- (1) 符合 Ethernet II 与 IEEE802.3 (10Base5、10Base2、10BaseT) 标准;
- (2) 全双工, 收发可同时达到 10Mbps 的速率;
- (3) 内置 16KB 的 SRAM, 用于收发缓冲, 降低对主处理器的速度要求;
- (4) 支持 8/16 位数据总线, 8 个中断申请线以及 16 个 I/O 基地址选择;
- (5) 支持 UTP、AUI、BNC 自动检测, 还支持对 10BaseT 拓扑结构的自动极性修正;
- (6) 允许 4 个诊断 LED 引脚可编程输出;
- (7) 100 脚的 PQFP 封装, 缩小了 PCB 尺寸。

### 2.2 内部结构

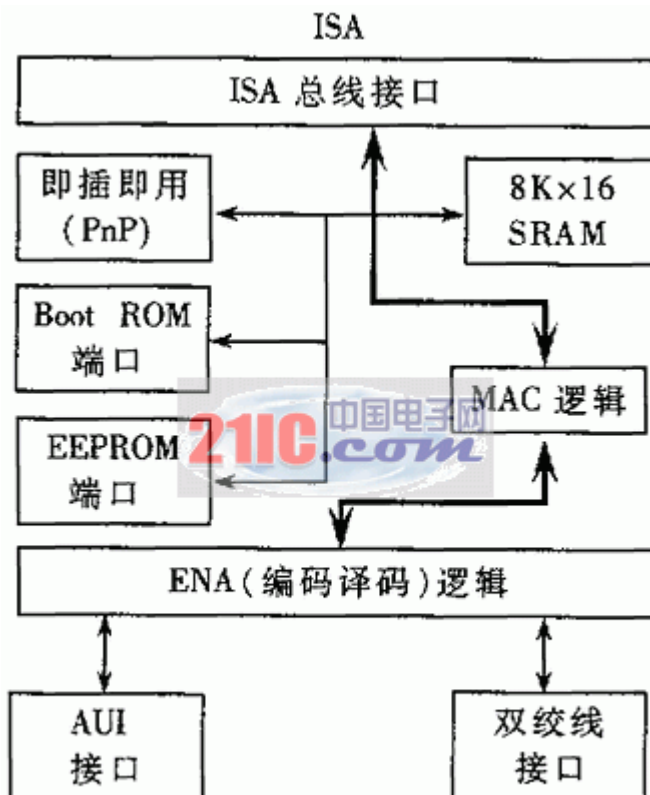


图 1 RTL8019AS 内部结构

RTL8019AS 内部可分为远程 DMA 接口、本地 DMA 接口、MAC (介质访问控制) 逻辑、数据编码解码逻辑和其他端口。内部结构如图 1 所示。

远程 DMA 接口是指单片机对 RTL8019AS 内部 RAM 进行读写的总线, 即 ISA 总线的接口部分。单片机收发数据只需对远程 DMA 操作。本地 DMA 接口是把 RTL8019AS 与网线的连接通道, 完成控制器与网线的数据交换。MAC (介质访问控制) 逻辑完成以下功能: 当单片机向网上发送数据时, 先将一帧数据通过远程 DMA 通道送到 RTL8019AS 中的发送缓存区, 然后发出传送命令; 当 RTL8019AS 完成了上帧的发送后, 再开始此帧的发送。RTL819 接收到的数据通过 MAC 比较、CRC 校验后, 由 FIFO 存到接收缓冲区; 收满一帧后, 以中断或寄存器标志的方式通知主处理器。FIFO 逻辑对收发数据作 16 字节的缓冲, 以减少对本地 DMA 请求的频率。

### 2.3 内部 RAM 地址空间分配

RTL8019AS 内部有两块 RAM 区。一块 16K 字节, 地址为 0x4000~0x7fff; 一块 32 字节, 地址为 0x0000~0x001f。RAM 按页存储, 每 256 字节为一页。一般将 RAM 的前 12 页 (即 0x4000~0x4bff) 存储区作为发送缓冲区; 后 52 页 (即 0x4c00~0x7fff) 存储区作为接收缓冲区。第 0 页叫 Prom 页, 只有 32 字节, 地址为 0x0000~0x001f, 用于存储以太网物理地址。

要接收和发送数据包就必须通过 DMA 读写 RTL8019AS 内部的 16KB RAM。它实际上是双端口的 RAM, 是指有两套总线连接到该 RAM, 一套总线 RTL8019AS 读或写该 RAM, 即本地 DMA; 另一套总线是单片机读或写该 RAM, 即远程 DMA。

### 2.4 I/O 地址分配

RTL8019AS 具有 32 位输入输出地址, 地址偏移量为 00H~1FH。其中 00H~0FH 共 16 个地址, 为寄存器地址。寄存器分为 4 页: PAGE0、PAGE1、PAGE2、PAGE3, 由 RTL8019AS 的 CR (Command Register 命令寄存器) 中的 PS1、PS0 位来决定要访问的页。但与 NE2000 兼容的寄存器只有前 3 页, PAGE3 是 RTL8019AS 自己定义的, 对于其他兼容 NE2000 的芯片如 DM9008 无效。远程 DMA 地址包括 10H~17H, 都可以用来做远程 DMA 端口, 只要用其中的一个就可以了。复位端口包括 18H~1FH 共 8 个地址, 功能一样, 用于 RTL8019AS 复位。

## 3 接口电路设计

下面介绍 51 系列单片机与 RTL8019AS 的接口电路, 实现的网络接口采用 UTP (无屏蔽双绞线) RJ-45 接口。图 2 给出了 89C52 单片机控制 RTL8019AS 实现以太网通讯的接口电路框图。用到的主要芯片有 80C52、RTL8019AS、93C46 (64×16bit 的 EEPROM)、74HC573 (8 位锁存)、62256 (32K 字节的 RAM)。为分配好地址空间, 采用对 93C46 进行读 (或写) 操作来设置 RTL8019AS 的端口 I/O 基地址和以太网物理地址。

93C46 是采用 4 线 SPI 串行接口的 Serial EEPROM, 容量为 1Kbit (64×16bit), 主要保存 RTL8019AS 的配置信息。00H~03H 的地址空间用于存储 RTL8019AS 内配置寄存器 CONFIG1~4 的上电初始化值; 地址 04H~11H 存储网络节点地址即物理地址; 地址 12H~7FH 内存储即插即用的配置信息。RTL8019AS 通过引脚 EECs、EESK、EEDI 控制 93C46 的 CS、SK、DI 引脚, 通过 EEDO 接收 93C46 的 DO 引脚的状态。RTL8019AS 复位后读取 93C46 的内容并设置内部寄存器的值, 如果 93C46 中内容不正确, RTL8019AS 就无法正常工作。

先通过编程器如 ALL07 把配置好的数据写入 93C46, 再焊入电路。

对 93C46 进行数据配置: 数据 00H 写入 93C46 的地址 00H 内; 93C46 地址 04H~0AH 中存放的是物理地址, 可以写入设置所需的物理地址值, 或不修改, 采用原始值为物理地址。通过编程器读出 04H~0AH 的原始值为 52544CC59906, 即所采用的物理地址。

这样, RTL8019AS 复位后读取 93C46 中配置好的内容, 对应设置配置寄存器 CONFIG1 的值为 00H, CONFIG1 的低 4 位 IOS3~0 用于选择 I/O 基地址。IOS3~0 设置值均为 0 时, RTL8019AS 选择的端口 I/O 基地址为 300H。

RTL8019AS 的地址为 20 位, 那么用到 RTL8019AS 的地址空间为 00300H~0031FH, 用二进制表示 00300H~0031FH, 可以发现第 19 位到第 5 位是固定的: 00000000011000。RTL8019 AS 的 20 根地址线 SA0~SA19 如表 2 连接。

表 2 RTL8019AS 地址线的连接表

SA19~SA10	接地
SA9~SA8	接单片机 P2 口的 P2. 7, 即地址总线 ADDR15
SA7~SA5	接地
SA4~SA0	对应为地址总线的 ADDR0~ADDR4

通过 ADDR15、I/OW、I/OR 来划分 RTL8019AS 和 62256 的地址空间。ADDR15 接 62256 的 CE 脚, 低电平时选择 62256; 高电平时选择 RTL8019AS 的地址空间 80C52 单片机的地址为 16 位, 按图 2 的电路框图连接 RTL8019AS 的地址空间。定义 reg00~reg1f 来对应端口 00300H~0031FH。

```
#define reg00 XBYTE[0x8000]/*300H*/
#define reg01 XBYTE[0x8001]/*301H*/
...
#define reg1f XBYTE[0x801F]/*31FH*/
```

读写操作、复位操作对应的引脚按表 3 连接。

表 3 RTL8019AS 的 ISA 总线接口引脚与单片机的连接表

Pin29	IORB	读信号, 接到单片机的 WR 引脚 (P3. 6)。
Pin30	IOWB	写信号, 接到单片机的 RD 引脚 (P3. 7)。
Pin33	RSTDRV	RESET 信号, 接单片机的 T0 (P3. 4)。
Pin34	AEN	地址有效信号, 接地。
Pin96	IOCS16	采用电阻下拉该引脚, 复位时刻脚为低电平, 选择 8 位模式。
Pin36~Pin43	SD0~SD7	8 位数据总线, 接单片机的 P0 口。(只用到 8 位数据总线, SD8~SD15 不接。)

采用 10BASE-T 布线标准通过双绞线进行以太网通讯, 而 RTL8019AS 内置了 10BASE-T 收发器, 所以网络接口的电路比较简单。外接一个隔离 LPF 滤波器 YL18-1001D, TPIN±为接收线, TPOUT±为发送线, 经 YL18-1001D 隔离后分别与 RJ-45 接口的 RX±、TX±端相连。

时钟电路通过 T1、T2 接口一个 20MHz 晶振以及 2 个电容, 实现全双工方式。

LED0、LED1 各接一发光二极管以反映通讯状态: LED0 表示 LED\_COL, 即通讯有冲突; LED1 表示 LED\_RX, 即接收到网上的信息包。

#### 4 程序设计

所有程序采用 Framclin C51 语言编制, 可读性强, 移植性好, 开发简易。

##### 4.1 复位 RTL8019AS

80C52 的 P3. 4 连接 RTL8019AS 的 RESDRV 来进行复位操作。RSTDRV 为高电平有效, 至少需要 800ns 的宽度。给该引脚施加一个 1  $\mu$ s 以上的高电平就可以复位。施加一个高电平后, 然后施加一个低电平。复位过程将执行一些操作, 至少需要 2ms 的时间, 推荐等待更久的时间之后如 100ms 之后才对它操作, 以确保完全复位。

##### 4.2 初始化 RTL8019AS

初始化页 0 与页 1 的相关寄存器, 页 2 的寄存器是只读的, 不可以设置, 页 3 的寄存器不是 NE2000 兼容的, 不用设置。

- (1) CR=0x21, 选择页 0 的寄存器;
- (2) TPSR=0x45, 发送页的起始页地址, 初始化为指向第一个发送缓冲区的页即 0x40;
- (3) PSTART=0x4c, PSTOP=0x80, 构造缓冲环: 0x4c~0x80;
- (4) BNRV=0x4c, 设置指针;

- (5) RCR=0xcc, 设置接收配置寄存器, 使用接收缓冲区, 仅接收自己地址的数据包 (以及广播地址数据包) 和多点播送地址包, 小于 64 字节的包丢弃, 校验错的数据包不接收;
- (6) TCR=0xe0, 设置发送配置寄存器, 启用 CRC 自动生成和自动校验, 工作在正常模式;
- (7) DCR=0xc8, 设置数据配置寄存器, 使用 FIFO 缓存, 普通模式, 8 位数据 DMA;
- (8) IMR=0x00, 设置中断屏蔽寄存器, 屏蔽所有中断;
- (9) CR=0x61, 选择页 1 的寄存器;
- (10) CURR=0x4d, CURR 是 RTL8019AS 写内存的指针, 指向当前正在写的页的下一页, 初始化时指和 0x4c +1=0x4d;
- (11) 设置多址寄存器 MAR0~MAR5, 均设置为 0x00;
- (12) 设置网卡地址寄存器 PAR0~PAR5;
- (13) CR=0x22, 选择页 1 的寄存器, 进入正常工作状态。

**图 2 89C52 单片机控制 RTL8019AS 实现以太网通讯的接口电路框图**

#### 4.3 发送帧

将待发送的数据按帧格式封装, 通过远程 DMA 通道送到 RTL8019AS 中的发送缓存区, 然后发出传送命令, 完成帧的发送。需要设置以太网目的地址、以太网源地址、协议类型, 再按所设置的协议类型来设置数据段。之后启动远程 DMA, 数据写入 RTL8019AS 的 RAM, 再启动本地 DMA, 将数据发送网上。

RTL8019AS 无法将整个数据包通过 DMA 通道一次存入 FIFO, 则在构成一个新的数据包之前必须先等待前一个数据包发送完成。为提高发送效率, 设计将 12 页的发送缓存区分为两个 6 页的发送缓存区, 一个用于数据包发送, 另一个用于构造端的数据包, 交替使用。

#### 5 软件的调试与验证

调试环境包括 RTL8019AS 实验板、PC 机 (带网卡) 以及网线。

用 C51 语言编程, 实现 TCP/IP 协议中 ARP 数据帧的收发。

实验中, 单片机首先构造一个 ARP 请求包发送给 PC 机, PC 机收到后会发送一个 ARP 应答包给单片机, 单片机收到该应答包后再发一个 ARP 请求包给 PC 机, 如此不断循环, 来测试系统的性能。在 PC 机上采用 Sniffer 软件如 Windump 软件来监视 (或截获) PC 机网卡接收 ARP 包的情况, 取得了满意的效果。