

# 用 Ax88796 实现 SA1110 的以太网接口

**摘要:** 讨论嵌入式微处理器 SA1110 与以太网控制器 Ax88796 在实际应用中的硬件连接问题; 给出软件设计方面的一些注意事项, 为 SA1110 的网络应用提供一种参考。

**关键词:** 嵌入式 SA1110 Ax88796 以太网 CPU MII 网络变压器 YL18-2050S YL37-1107S

Intel 公司的 StrongARM SA1110 是一款主要面向嵌入式应用的高性能 32 位微处理器, 目前已经被广泛应用在 PDA 等手持设备上。SA1110 最高主频可达 206 MHz, 具有内存管理单元(MMU)和大容量的指令/数据高速缓存(16 KB/8 KB), 兼容 SDRAM、SMROM 等多种存储设备, 并带有 LCD 和 PCMCIA 控制器。

## 1 Ax88796 简介

Ax88796 是台湾 Asix 公司推出的 NE2000 兼容快速以太网控制器。其内部集成有 10/100 Mb/s 自适应的物理层收发器和 8K×16 位的 SRAM, 支持 MCS-51 系列、80186 系列以及 MC68K 系列等多种 CPU 总线类型。

Ax88796 执行基于 IEEE802.3/IEEE802.3u 局域网标准的 10Mb/s 和 100Mb/s 以太网控制功能, 并提供 IEEE802.3u 兼容的媒质无关接口 MII(Media Independent Interface), 用以支持在其它媒质上的应用。此外, Ax88796 还提供可选用的标准打印接口, 可用于连接打印设备或用作通用 I/O 端口。其结构框图如图 1 所示。

Ax88796 的地址总线 SA[9:0]与数据总线 SD[15:0]分别与 CPU 的地址/数据总线相连。CPU 通过 I/O 读写 NE2000 寄存器来控制 Ax88796 的工作状态, 通过远程 DMA FIFOs 与 Ax88796 的内部缓存 SRAM 进行数据交换。SRAM 与 MAC 核之间进行 Local DMA 将数据发送至 MAC 层, 再经由内部的 PHY 层经过网络变压器 YL18-2050S 发送至 RJ45 接口, 或者经过 MII 接口送至外部的物理层芯片, 再由网络变压器 YL18-2050S 或 YL37-1107S 发送到 RJ45 接口。

EEPROM 接口可以用来连接串行 EEPROM。EEPROM 可用于存储 MAC 地址, 供 Ax88796 每次初始化时读取。

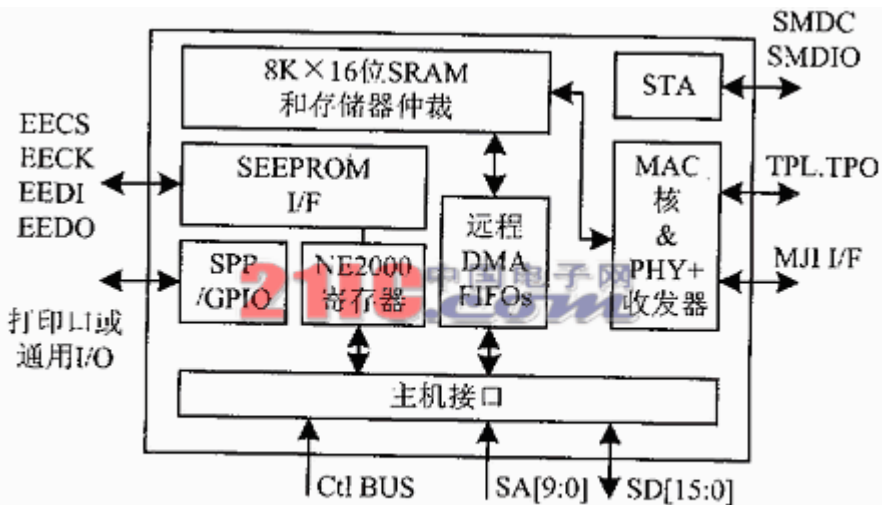


图 1 Ax88796 结构框图

## 2 硬件接口电路

Ax88796 的 CPU[1:0]两个输入引脚用来设置与不同 CPU 总线连接时 Ax88796 的工作模式。在 Asix 公司官方提供的资料中, Ax88796 在与 SA1110 连接时, 是将这两个引脚都拉低, 也即设为 ISA 总线模式。但实际应用中发现这样设置, 在对 Ax88796 奇地址寄存器进行 8 位数据宽度的访问时存在问题: 无论写入何值, 读出值总为 00H (由于 Ax88796 是 NE2000 系列的以太网控制器, 其 MAC 层的控制寄存器都采用 8 位数据宽度, 因此存放地址有奇偶之分; 而 CPU 在对 Ax88796 的控制中, 需要对其寄存器进行 8 位数据宽度的读写)。

最初的怀疑是, 对 Ax88796 的奇地址寄存器无法正确写入数据。但用示波器抓取数据总线上的信号, 发现 SA1110 已经正确送出数据; 并且当对 Ax88796 的奇地址寄存器进行读操作时, Ax88796 在数据总线 D

[7:0]上送出的数据正是此前 SA1110 向其奇地址寄存器写入的数据。也就是说，数据已经被正确写入了 Ax88796 的奇地址寄存器中，但 SA1110 从中读取数据时出现了问题。

根据 Ax88796 资料中的相关说明，在 ISA 总线模式下，对其 NE2000 寄存器进行 8 位数据宽度操作时，高 8 位数据线 D[15:8]是被内部拉低的。因此，数据都是在数据总线 D[7:0]上进行传送的，而 Ax88796 由地址线 A0 上信号电平的高低来判断所访问地址的奇偶，如表 1 所列。

**表 1 Ax88796 在 ISA 总线模式下的读写操作**

(a) ISA 总线模式下的读操作

操作模式	CS	BHE	A0	IORD	IOWR	SD[15:8]	SD[7:0]
空闲模式	H	X	X	X	X	高阻	高阻
字节访问	L	H	L	L	H	无效	偶字节数据
	L	H	H	L	H	无效	奇字节数据
双字节访问	L	L	L	L	H	奇字节数据	偶字节数据

(b) ISA 总线模式下的写操作

操作模式	CS	BHE	A0	IORD	IOWR	SD[15:8]	SD[7:0]
空闲模式	H	X	X	X	X	X	X
字节访问	L	H	L	H	L	X	偶字节数据
	L	H	H	H	L	X	奇字节数据
双字节访问	L	L	L	H	L	奇字节数据	偶字节数据

H-高电平，L-低电平，X-不确定

SA1110 最大支持 32 位数据总线，它在进行不同宽度的数据读写时，有其自身的一套机制：

在对奇地址进行 8 位数据宽度的写操作时（如写 19H），地址线 A0 送出 1，而数据总线 D[31:24]、D[23:16]、D[15:8]、D[7:0]上同时送出该 8 位数据（即送出 19191919H）；在对奇地址进行 8 位数据宽度的读操作时，地址线 A0 送出 1，数据总线 D[15:8]上的数据作为有效数据被读入 SA1110 的内部寄存器，而其余数据线上的数据被丢弃。

由此可见，当 SA1110 对 Ax88796 的奇地址寄存器进行 8 位数据宽度的读操作时，Ax88796 在数据总线 D[7:0]上送出的有效数据，被 SA1110 丢弃了；而数据总线 D[15:8]上的值 00H 被误认作有效数据读入了 SA1110。这就解释了为什么对 Ax88796 奇地址寄存器可以正确写入却无法正确读取的问题了。

综合上面的分析，在实际中采用的接口电路如图 2 所示。

Ax88796 的 CPU[0]上拉，CPU[1]下拉，设置它工作在 186 总线模式下。SA1110 的地址线 A0 经反向器后接至 Ax88796 的引脚，用来确保 SA1110 对 Ax88796 的控制寄存器的访问都为 8 位数据宽度。实验证明，这并不影响 Ax88796 作 Remote DMA 时与 SA1110 之间的 16 位数据宽度传输。

SA1110 的 GPIOx 引脚可以复用作中断输入，因此用来连接 Ax88796 的中断输出 IREQ。

### 3 软件方面

由于 Ax88796 是 NE2000 兼容芯片，所以 Linux kernel 中的驱动 ne.c、8390.c 只需作较少的改动就可以用来驱动 SA1110 平台上的 Ax88796：

将 ne.c 中的侦测端口地址列表 netcard\_portlist[] 的初始赋值改为 kernel 中为 nCSx 所映射的虚拟地址。

将 ne.c 里的中断侦测程序段去掉，分配 `dev->irq=IRQ_GPIOx`；而在 8390.c 中的每次 `enable_irq` 函数前，将 `GPIOx` 设置为上升沿触发。

由于 Ax88796 具有独立的端口被用来支持对 EEPROM 的访问，因此 ne.c 中访问 EEPROM 所采取的 Remote DMA 方式应改写为对端口的访问方式。

另外值得一提的是，SA1110 的片选输出 `nCSx` 与读写信号输出 `nOE`、`nWE` 之间的时序关系可以由对应的 SA1110 寄存器 `MSCx` 中的控制位来调节。一个合适的取值可以大大提高 SA1110 与 Ax88796 之间的数据传输速度。

