# 目录

# CS8900A 产品数据手册中文翻译部分

## 3.0 功能描述

## 3.1 概述

在正常操作下，CS8900A 表现出两个基本的功能：以太网数据包发送和接收。在可以发送或接收前，必须配置 CS8900A。

### 3.1.1 配置

在启动或重置时，必须配置 CS8900A 来进行数据包发送和接收。许多参数必须写进它的配置和控制寄存器例如内存基地址；以太网物理地址；接收什么类型的帧；和使用哪些媒体接口。配置数据可以由主机（通过 ISA 总线）写到 CS8900A，或自动从外部 EEPROM 加载。配置完成后就可以进行操作了。

请看 18 页 3.3 节和 20 页 3.4 节关于配置过程的细节描述。46 页 4.4 节提供配置和控制寄存器的比特位的细节描述。

### 3.1.2 数据包发送

数据包发送发送有两个阶段。在第一阶段，主机移动以太网帧到 CS8900A 的缓冲内存。第一阶段以主机提交发送命令为开始。这通知 CS8900A 一个帧将要被发送和告诉芯片什么时候开始发送（例如 5, 381, 1021 或所有字节已经传送到网卡时）和帧应该怎样被发送（例如含不含有 CRC，含不含有填充比特，等）。主机先提交发送命令然后填写发送长度，表明需要多少缓冲空间。当缓冲空间可用，主机通过内存或 I/O 空间操作，写以太网帧到 CS8900A 内部内存。

在发送的第二阶段，CS8900A 转换帧成为以太网数据包，然后发送到网络。第二阶段以当有适当数目的字节已经传送到它的发送缓冲区时（5, 381, 1021 或整个帧，依据配置）CS8900A 进行发送报文头和帧开始定界符为开始。在报文头和帧开始定界符之后是目的地址，源地址，长度域和 LLC 数据（全部由主机提供）。

如果帧少于 64 字节，包括 CRC 在内，CS8900A 会根据配置来判断是否添加填充比特。最后 CS8900A 添加适当的 32 位 CRC 值。

98 页 5.7 节提供关于数据包发送的细节描述。

### 3.1.3 数据包接收

像数据包发送一样，数据包接收发送有两个阶段。在第一阶段，CS8900A 接收以太网数据包并储存它到片上内存。数据包接收的第一阶段以接收帧通过模拟前端和曼彻斯特解码器，解码器负责把曼彻斯特数据转换为不归零数据。然后，报文头和帧开始定界符被剥去并且接收帧传送到地址过滤器。如果帧目的地址符合地址过滤器的配置标准，数据包就储存在 CS8900A 的内部内存。CS8900A 然后检查 CRC，并根据配置通知处理器有接收帧。

在第二阶段，主机通过 ISA 总线传送接收帧到主机内存。接收帧可以通过内存空间操作，I/O 空间操作，或使用主机的 DMA 进行 DMA 操作来传送。同时 CS8900A 使用自动转换 DMA 和流发送，提供在内存或 I/O 操作和 DMA 操作之间转换的能力。

78 页 5.2 节到 95 页 5.6 节提供对数据包接收的细节描述。

## 3.2 ISA 总线接口

CS8900A 提供直接接口和以 8 到 11MHz 的时钟频率运行的 ISA 总线连接。它的片上总线驱动有能力发送 24mA 的驱动电流，允许 CS8900A 直接驱动 ISA 总线，而不用添加额外的“胶连逻辑”。

CS8900A 优化 16 位操作在内存空间，I/O 空间，或作为 DMA 从设备下的数据传送。

注意 ISA 总线操作在 8MHz 以下时，应该使用 CS8900A 的接收 DMA 模式来减少帧丢失。请看 89 页 5.4 节关于接收 DMA 操作的描述。

### 3.2.1 内存模式操作

当配置为内存模式操作时，CS8900A 的内部寄存器和帧缓冲映射到连续的 4K

字节的主机内存块，来使主机直接存取 CS8900A 的内部寄存器和帧缓存。主机通过使 MEMR 引脚接低电平来初始化读操作，和使 MEMW 引脚接低电平来初始化写操作。

请看 73 页 4.9 节关于内存模式的资料。

### 3.2.2 I/O 模式操作

当配置为 I/O 模式操作时，通过映射到主机 16 个连续的 I/O 位置的 8 个 16 位 I/O 端口来存取 CS8900A。I/O 模式是 CS8900A 一个默认的配置，并且始终使能。

进行 I/O 读写操作时，AEN 引脚必须是低电平，而且 ISA 系统地址总线(SA0 - SA15)的 16 位 I/O 地址必须符合 CS8900A 的地址空间。进行 I/O 读操作时，IOR 必须是低电平，进行 I/O 写操作时，IOW 必须是低电平。

请看 75 页 4.10 节关于内存模式的资料。

### 3.2.3 中断请求信号

CS8900A 有 4 个可以直接连接到 ISA 总线上任意 4 个中断请求信号的中断请求输出引脚。任意时刻只能使用一个中断输出。在初始化阶段通过向 PacketPage Memory base + 0022h 地址写入中断号(0 to 3)来选择中断引脚。不使用的中断请求引脚置为高阻抗状态。当使能了的中断被激活，选择的中断请求引脚就变为高电平。不断读取中断状态队列(ISQ)寄存器直至其值为 0 时，引脚变为低电平（请看 78 页 5.1 节关于 ISQ 的描述）。

表格 1 提供一种使用一般可用中断且减轻开发板布置的可行方法来把中断请求引脚连接到 ISA 总线。

| CS8900A 中断请求引脚 | ISA 总线中断 | PacketPage base + 0022h |
|---|---|---|
| INTRQ3 (Pin 35) | IRQ5 | 0003h |
| INTRQ0 (Pin 32) | IRQ10 | 0000h |
| INTRQ1 (Pin 31) | IRQ11 | 0001h |
| INTRQ2 (Pin 30) | IRQ12 | 0002h |

表格 1. 中断分配

### 3.2.4 DMA 信号

CS8900A 直接和主机的 DMA 控制器连接，从而为来自 CS8900A 内存的接收帧提供到主机内存的 DMA 传送。CS8900A 有 3 对 DMA 引脚来直接连接 ISA 总线的 3 个 16 比特的 DMA 通道。同一时刻只能使用一个 DMA 通道。在初始化期间通过写入要使用的通道号码(0, 1 或 2)到 PacketPage Memory base + 0024h 来选择通道。不使用的 DMA 引脚置为高阻抗状态。当 CS8900A 接收到帧并通过 DMA 传送到主机内存时，选择了的 DMA 请求引脚变为高电平。如果 DMABurst 比特位(寄存器 17, BusCTL, Bit B)清 0，DMA 操作完成后引脚就变为低电平。如果 DMABurst 比特位置 1，DMA 传送开始后引脚维持 32 微秒的低电平。

DMA 引脚对排布在 CS8900A 里来减少开发板的布局。连接到总线时，Crystal 公司推荐表格 2 的配置。

| CS8900A DMA 信号 (Pin #) | ISA DMA 信号 | PacketPage base + 0024h |
|---|---|---|
| DMARQ0 (Pin 15) | DRQ5 | 0000h |
| DMACK0 (Pin 16) | DACK5 | |
| DMARQ1 (Pin 13) | DRQ6 | 0001h |
| DMACK1 (Pin 14) | DACK6 | |
| DMARQ2 (Pin 11) | DRQ7 | 0002h |
| DMACK2 (Pin 12) | DACK7 | |

表格 2. DMA 分配

请看 89 页 5.4 节关于 DMA 模式的描述。

## 3.3 重置和初始化

### 3.3.1 重置

7 种不同的情况引起 CS8900A 重置内部寄存器和电路。

#### 3.3.1.1 外部重置，或 ISA 重置

当 RESET 引脚持续至少 400ns 的高电平时，发生一个芯片宽重置。在芯片宽重置期间，所有 CS8900A 的电路和寄存器重置。

#### 3.3.1.2 加电重置

当提供电力时，CS8900A 维持重置直到在供电引脚的电压到达大约 2.5 伏特。一旦 Vcc 大于大约 2.5 伏特并且水晶振荡器平稳后，CS8900A 重置结束。

#### 3.3.1.3 掉电重置

如果提供电压跌到大约 2.5 伏特以下，出现芯片宽重置。一旦电力提供回到大于大约 2.5 伏特并且水晶振荡器平稳后，CS8900A 重置结束。

#### 3.3.1.4 EEPROM 重置

如果检测到 EEPROM 校验和错误，出现芯片宽重置（参考 20 页 3.4 节）。

#### 3.3.1.5 软件初始重置

当 RESET 比特位(寄存器 15, SelfCTL, Bit 6)置 1，出现芯片宽重置。

#### 3.3.1.6 硬件(HW) 待命或暂停

当 CS8900A 进入或退出 HW 待命模式或 HW 暂停模式（参考 25 页 3.7 节关于 HW 待命或暂停的更多资料）时，CS8900A 芯片宽重置结束。

#### 3.3.1.7 软件(SW) 暂停

当 CS8900A 进入 SW 暂停模式时，除了 ISA I/O 基地址寄存器（位于 PacketPage base + 0020h）和 SelfCTL 寄存器(寄存器 15)外，所有寄存器和电路重置。一旦退出，出现芯片宽重置（参考 25 页 3.7 节关于 SW 暂停的更多资料）。

### 3.3.2 允许重置时间操作

重置后，CS8900A 经过一个自配置过程。它包括校准片上模拟逻辑，和读取 EEPROM 来验证和配置。重置校准需要的时间一般为 10ms。这时软件驱动不应该存取 CS8900A 的内部寄存器。当校准完成后，Self 状态寄存器(寄存器 16)的 INITD 比特位置 1，这意味着初始化完成，并且同一个寄存器的 SIBUSY 比特位清 0，这

表明 EEPROM 不再被读取或编程。

### 3.3.3 总线重置补偿

重置后，CS8900A 从 IO base+0Ah 读取 3000h，直到软件写入一个非零的值到 IO base+0Ah。当系统扫描 CS8900A 时，3000h 这个值可以被看做为 CS8900A 签名的一部分。请看 75 页 4.10 节。

重置后，ISA 总线输出引脚 INTRQx 和 DMARQx 是三态门的，这样避免 ISA 总线加电时的任何中断或 DMA 通道冲突。

### 3.3.4 初始化

每种重置后（除了 EEPROM 重置），CS8900A 检查 EEDataIn 引脚的电平，来看是否有外部 EEPROM 存在。如果 EEDI 是高电平，则 EEPROM 存在并且 CS8900A 自动加载储存在 EEPROM 里的配置数据到它的内部寄存器（参考下一节）。如果 EEDI 是低电平，则 EEPROM 不存在并且 CS8900A 重置后的默认配置如表格 3。

| PacketPage 地址 | 寄存器内容 | 寄存器描述 |
|---|---|---|
| 0020h | 0300h | I/O 基地址* |
| 0022h | XXXX XXXX XXXX X100 | 中段号码 |
| 0024h | XXXX XXXX XXXX XX11 | DMA 通道 |
| 0026h | 0000h | DMA Start of Frame Offset |
| 0028h | X000h | DMA 帧计数 |
| 002Ah | 0000h | DMA 字节计数 |
| 002Ch | XXX0 0000h | 内存基地址 |
| 0030h | XXX0 0000h | 启动 PROM 基地址 |
| 0034h | XXX0 0000h | 启动 PROM 地址掩码 |
| 0102h | 0003h | 寄存器 3 - RxCFG |
| 0104h | 0005h | 寄存器 5 - RxCTL |
| 0106h | 0007h | 寄存器 7 - TxCFG |
| 0108h | 0009h | 寄存器 9 - TxCMD |
| 010Ah | 000Bh | 寄存器 B - BufCFG |
| 010Ch | Undefined | 保留 |
| 010Eh | Undefined | 保留 |
| 0110h | Undefined | 保留 |
| 0112h | 0013h | 寄存器 13 - LineCTL |
| 0114h | 0015h | 寄存器 15 - SelfCTL |
| 0116h | 0017h | 寄存器 17 - BusCTL |

| | | |
|---|---|---|
| 0118h | 0019h | 寄存器 19 - TestCTL |

* I/O 基地址是不会受软件暂停模式影响的。

表格 3. 默认配置

一个低廉的串行 EEPROM 可以用来储存配置消息，在每种重置后（除了 EEPROM 重置）用来自动加载到 CS8900A。EEPROM 是可选择使用的。

CS8900A 和表格 4 中显示的 6 个标准 EEPROM 的任意一个一起操作。

| EEPROM 类型 | Size (16 比特字) |
|---|---|
| ‘C46 (non-sequential) | 64 |
| ‘CS46 (sequential) | 64 |
| ‘C56 (non-sequential) | 128 |
| ‘CS56 (sequential) | 128 |
| ‘C66 (non-sequential) | 256 |
| ‘CS66 (sequential) | 256 |

表格 4. 提供 EEPROM 类型

# 4.9 内存模式操作

要把 CS8900A 配置成内存模式，PacketPage 内存必须被映射进一个主机内存的连续的 4-kbyte 空间内。这块空间必须以 X000h 为开始边界，即 PacketPage 的基地址映射到 X000h。当 CS8900A 被重置后，它默认的的配置是 I/O 模式。一旦 Memory 模式被选择后，CS8900A 所有的寄存器可以被直接存取。

在内存模式下，CS8900A 支持标准的或者准备好的且没有引入额外的等待状态的总线周期类型。

内存写入可以用 MOVD 命令（双字传送）只要 CS8900A 的内存基地址是在双字界上。因为 286 处理器不支持 MOVD 指令，字和字节传送必须使用 286 指令集。

| 描述 | 符号 | 读/写 | 位于: PocketPage base + |
|---|---|---|---|
| 接收状态 | RxStatus | 只读 | 0400h-0401h |
| 接收长度 | RxLength | 只读 | 0402h-0403h |
| 接收帧 | RxFrame | 只读 | 起始于 0404h |
| 发送帧 | TxFrame | 只写 | 起始于 0A00h |

表 16. 接收/发送内存位置

## 4.9.1 进入内存模式

CS8900A 允许对内部 PacketPage 内存的读/写，和可选 Boot PROM 的写操作。（参考 25 页 3.7 节对可选 Boot PROM 的描述。）当下面所有条件都成立时才可以进行内存存取：

- ISA 系统地址总线（SA0 - SA19）上的地址在 CS8900A 或 Boot PROM 的内存空间范围内。
- CHIPSEL 输入引脚是低电平。
- MEMR 引脚或 MEMW 引脚是低电平。

## 4.9.2 把 CS8900A 配置成内存模式

有两种不同的方法将 CS8900A 配置成内存模式操作。一种方法允许 CS8900A 的内部内存被映射进主机系统的 24 位内存空间的任意地方。另一种方法限制内存

映射到主机内存空间的第一个 1M 字节空间。

一般的内存模式的操作：配置 CS8900A 要使它的内部内存可以映射到主机内存空间的任意地方需要以下操作：

- 一个简单电路必须被添加来解码可鎖存的地址总线（LA20-LA23）和 BALE 引脚信号。
- 主机必须如下把外部逻辑配置成正确地址范围：

1）查看 INITD 位（寄存器 16，SelfST，Bit 7）是否设为 1，如果是则表明初始化是完整的。

2）查看 ELpresent 位（寄存器 16，SelfST，Bit B）是否设为 1。这一位表明存在 LA 总线解码的外部逻辑。

3）设置 EEPROM 命令寄存器的 ELSEL 位为 1 来激活 ELCS 引脚来使用外部解码电路。

4）连续配置外部逻辑。

- 主机必须把内存基地址写进 Memory Base Address 寄存器（PacketPage base + 002Ch）；
- 主机必须设置 MemoryE 位为 1（寄存器 17，BusCTL，Bit A）；
- 并且主机必须设置 UseSA 位为 1（寄存器 17，BusCTL，Bit 9）。

限制内存模式到主机内存空间第一个 1 Mbyte 里：需要以下条件配置 CS8900A，使内部内存只能映射主机内存空间第一个 1 Mbyte 里：

- CHIPSEL 引脚必须连接到低电平；
- ISA-bus SMEMR 信号必须被连接到 MEMR 引脚；
- ISA-bus SMEMW 信号必须被连接到 MEMW 引脚；
- 主机必须把内存基地址写进 Memory Base Address 寄存器（PacketPage base + 002Ch）；
- 主机必须设置 MemoryE 位为 1（寄存器 17，BusCTL，Bit A）；
- 并且主机必须把 UseSA 位清 0（寄存器 17，BusCTL，Bit 9）。

### 4.9.3 基本内存模式发送

内存模式发送操作以如下顺序发生（使用中断）：

1）主机通过向 TxCMD 寄存器(memory base + 0144h)写入发送命令和向 TxLength 寄存器(memory base + 0146h)写入发送帧的长度来争取帧的保存。如果发送长度是错误的，命令就会被丢弃并且 TxBidErr 位 (寄存器 18，BusST，Bit 7) 设为 1。

2）主机读取 BusST 寄存器(寄存器 18，memory base + 0138h)。如果 Rdy4TxNOW 位(Bit 8)设为 1，帧就可以被写入。如果清零，主机必须等待 CS8900A 缓冲区内存变成可用。如果 Rdy4TxiE(寄存器 B,BufCFG, Bit 8) 设为 1，当 Rdy4Tx(寄存器 C, BufEvent, Bit 8) 设为 1 时主机就会被中断。

3）一旦 CS8900A 准备好接收帧，主机执行重复的内存到内存的 move 指令（REP MOVS）来向 memory base + 0A00h 这个地址把整个帧从主机内存传进 CS8900A 内存。

想得到更多的关于发送的细节描述，请看 98 页的 5.7 节。

### 4.9.4 基本内存模式接收

内存模式接收操作以如下顺序发生（中断用来通知一个有效的接收帧的存在)：

1）一个帧被 CS8900A 接收到后触发一个激活了的中断。

2）主机读取中断状态队列(memory base + 0120h)并且被通知收到帧。

3）主机读取 RxStatus 寄存器(memory base +0400h)来知道接收到的帧的状态。

4）主机读取 RxLength 寄存器(memory base +0402h) 来知道接收到的帧的长度。

5）主机读取帧数据时通过执行重复的内存到内存的 move 指令(REP MOVS)向 memory base + 0404h 从 CS8900A 内存到主机内存发送整个帧。

想得到更多的关于接收的细节描述，请看 78 页的 5.2 节。

### 4.9.5 在内存模式下轮询 CS8900A

如果中断不被使用，主机可以轮询 CS8900A 来检查是否有接收帧的存在和内存空间是否可用于发送。但是，这是不在数据手册的范围之内。

## 4.10 I/O 空间操作

再 I/O 模式，PacketPage 内存通过映射到主机系统的 I/O 空间的 16 个连续 I/O 位置的 16 位的 I/O 端口来存取。I/O 模式是 CS8900A 默认的配置并且总是激活的。当通电之后，I/O 基地址的默认值设置为 300h。（请注意 300h 通常被分配给局域网外围设备）。I/O 基地址可以被改为任何的可用的 XXX0h 地址位置，或者是通过从 EEPROM 下载配置数据，或者是系统启动期间。表格 17 展示了 CS8900A 的 I/O 模式映射。

| 偏移量 | 类型 | 描述 |
|---|---|---|
| 0000h | 读/写 | 接收/发送数据（端口 0） |
| 0002h | 读/写 | 接收/发送数据（端口 1） |
| 0004h | 只写 | TxCMD（发送命令） |
| 0006h | 只写 | TxLength（发送长度） |
| 0008h | 只读 | 中断状态队列 |
| 000Ah | 读/写 | PacketPage 指针 |
| 000Ch | 读/写 | PacketPage 数据（端口 0） |
| 000Eh | 读/写 | PacketPage 数据（端口 1） |

表 17. I/O 模式映射

### 4.10.1 接收/发送数据端口 0 和 1

这两个端口当向 CS8900A 发送发送数据时被使用。端口 0 用于 16 位操作并且端口 0 和 1 用于 32 位操作（低字位放在端口 0）。

### 4.10.2 TxCMD 端口

主机在每次发送操作开始都把发送命令(TxCMD)写进这个端口。发送命令告诉 CS8900A 主机有帧要发送，同时帧应该怎样被发送。这端口被映射到 PacketPage base + 0144h。请看 46 页 4.4 节寄存器 9 来获得更多信息。

### 4.10.3 TxLength 端口

发送帧的长度在发送命令写入后马上被写进这里。这个端口映射到 PacketPage base+ 0146h。

### 4.10.4 中断状态队列端口

这端口包含当前中断状态队列的值(ISQ)。ISQ 在 PacketPage base + 0120h 这个位置。想得到更多关于 ISQ 的描述，请看 78 页 5.1 节。

### 4.10.5 PacketPage 指针端口

任何时候主机想存取任何的 CS8900A 的内部寄存器，PacketPage 指针端口就会被写入。第一个 12 位（bits 0 到 B）提供了在目前操作期间要存取的目标寄存器的内部地址。后三位（C, D, 和 E）是只读的并且读到的总是 011b。当写入 PacketPage 指针端口时，任何方便的值可以被写入这些位。最后一位（Bit F）表明 PacketPage 指针是否应该被自动增加来指向下一个字的位置。图 18 展示 PacketPage 指针的结构。

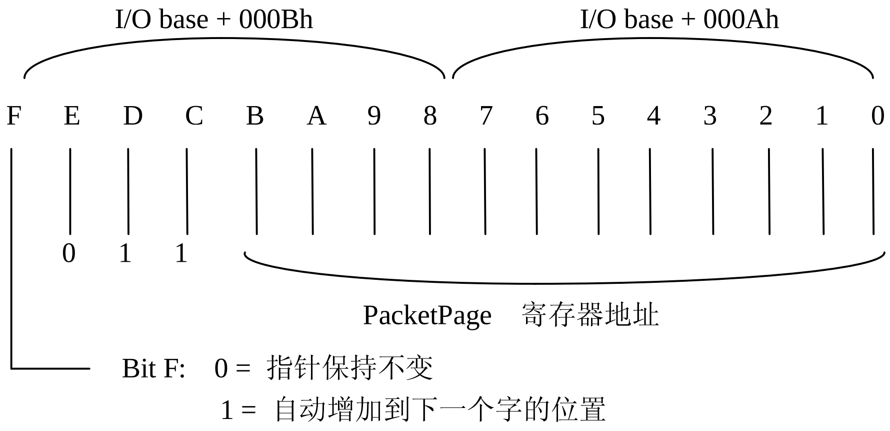

图 18. PacketPage 指针

### 4.10.6 PacketPage 数据端口 0 和 1

PacketPage 数据端口是用来向 CS8900A 内部寄存器交换数据的。端口 0 用于

16-bit 操作并且端口 0 和 1 用于 32-bit 操作（低字位放在端口 0）。

### 4.10.7 I/O 模式操作

要想 I/O 读或写操作，AEN 引脚必须是低电平，并且在 ISA 系统地址总线(SA0 - SA15) 上的 16 位 I/O 地址必须符合 CS8900A 的地址空间。读数据时，IOR 引脚必须是低电平，和写数据时，IOW 引脚必须是低电平。

注意：ISA 可锁存地址总线(LA17 -LA23)对于只用 I/O 模式和接收 DMA 操作的应用是不需要的。

### 4.10.8 基本 I/O 模式发送

I/O 模式发送操作以以下顺序进行（使用中断）：

1） 主机通过向 TxCMD 端口(I/O base + 0004h)写入发送命令并且向 TxLength 端口(I/O base + 0006h) 写入发送帧的长度来请求存储帧。

2） 主机读 BusST 寄存器(Register 18)来看 Rdy4TxNOW 位(Bit 8)是否置为 1。要读 BusST 寄存器，主机必须先通过写入 0138h 到 PacketPage 指针端口(I/O base + 000Ah)设置 PacketPage 指针到正确的位置。它就能从 PacketPage 数据端口(I/O base + 000Ch)读取 BusST 寄存器。如果 Rdy4TxNOW 位设置为 1，这个帧可以被写入。如果清为 0，主机必须等待 CS8900A 缓冲内存变为可用为止。如果 Rdy4TxiE(寄存器 B, BufCFG, Bit 8)设为 1，当 Rdy4Tx (寄存器 C, BufEvent, Bit 8)置为 1，主机就会被中断。如果 TxBidErr 位(寄存器 18, BusST, Bit 7) 置为 1，发送长度就无效。

3） 一旦 CS8900A 准备好接收帧，主机向接收/发送数据端口(I/O base + 0000h)执行重复写指令(REP OUT)来从主机内存向 CS8900A 内存传送整个帧。

想得到更多发送的细节描述，请看 98 页 5.7 节。

### 4.10.9 基本 I/O 接收

I/O 模式接收操作以以下顺序进行（在这个例子里，中断被激活来通知一个有效接收帧的存在）：

1） 一个被 CS8900A 接收的帧，触发并使能中断。

2） 主机读中断状态队列端口(I/O base + 0008h)并被通知有接收帧。

3） 主机读取帧数据时通过执行重复的读指令(REP IN)向接收/发送数据端口(I/O base + 0000h)来从 CS8900A 内存向主机内存来传送数据。在帧数据之前是 RxStatus 寄存器 (PacketPage base + 0400h)和 RxLength 寄存器 (Pack etPage base + 0402h)的内容。

想得到更多接收的细节描述，请看 78 页 5.2 节。

### 4.10.10 采取内部寄存器

要在 I/O 模式下存取 CS8900A 任意的内部寄存器，主机必须先建立 PacketPage 指针。它做到这点是通过写入目标寄存器的 PacketPage 地址到 PacketPage 指针端口(I/O base + 000Ah)。目标寄存器的内容然后就映射进 PacketPage 数据端口(I/O base + 000Ch)。

如果主机需要采取连续的寄存器块，要存取的第一个字的 PacketPage 地址的 MSB(最高有效位)应该设为 1。PacketPage 指针就会自动移到下一个字的位置，消除了在连续存取时建立 PacketPage 指针需要（查看图 18）。

### 4.10.11 I/O 模式下轮询 CS8900A

如果不使用中断，主机可以轮询 CS8900A 来检查接收帧是否存在和内存空间是否可用于发送

## 5.2　基本接收操作

#### 5.2.0.1　概述

一旦一个传入的数据包传到模拟前端和曼彻斯特解码器，它会经过以下三个发送处理步骤：

1）　预处理

2）　临时缓冲

3）　传送到主机

图 20 展示帧的接收步骤。

如图所示，不管任何传送方法，所有接收帧经过相同预处理和临时缓冲阶段。

一旦一个帧被预处理和缓冲，它可以在内存或 I/O 空间被主机取出。另外，CS8900A 可以通过主机 DMA 传送接收帧到主机内存。这一节描述接收帧的预处理和内存和 I/O 空间的接收操作。89 页 5.4 节到 92 页 5.5 节描述 DMA 操作。

### 5.2.1　专业术语：数据包，帧，传送

词汇数据包，帧，传送广泛在下面的章节中使用。他们的清晰定义在下面：

#### 5.2.1.1　数据包

词汇“数据包”是指在以太网发送的整个比特序列串。包括报文头，帧的开始分隔符（SFD），目的地址（DA），源地址（SA），长度域，数据域，填充位（如果有需要），还有帧检查序列（FCS，也叫做 CRC）。图 9 展示了数据包的格式。

#### 5.2.1.2　帧

词汇“帧”是指一个数据包从目的地址到帧检查序列的部分。这包括目的地址（DA），源地址（SA），长度域，数据域，填充位（如果有需要），还有帧检查序列（FCS，也叫做 CRC）。图 9 展示了数据包的格式。词汇“帧数据”是指从目的地址到帧检查序列之间将要发送，或已经被接收的所有数据。

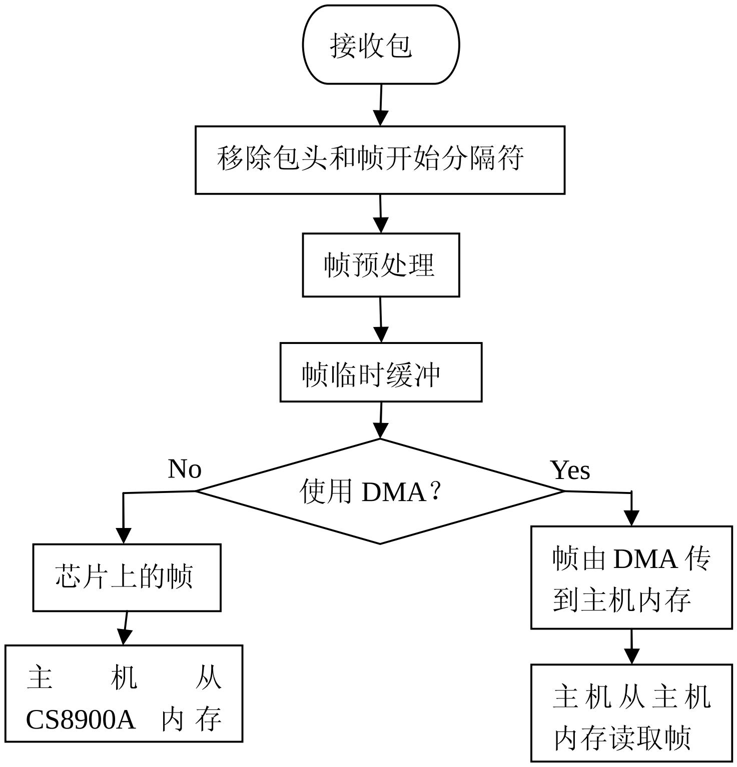

图 20. 帧接收

### 5.2.1.3 传送

词汇“传送” 是指通过 ISA 总线，向 CS8900A 或从他那里移动数据。在接收操作期间，只有帧数据被从 CS8900A 向主机传送（报文头和帧开始分隔符被 CS8900A 的 MAC 引擎去除）。帧检查序列可以也可以不被传送，这依赖于怎样配置。所有向或来自 CS8900A 的传送是以字节来计算的，但可能被填充为双字对齐。

## 5.2.2 接收配置

在每次重置后，必须为接收操作配置 CS8900A。可以通过使用连接的 EEPROM 或写入配置命令到 CS8900A 的内部寄存器（请看 20 页 3.4 节）做到这点。必需配置的项目包括：

- 使用哪个物理接口；
- 接收哪种类型的帧；
- 哪个接收事件引起中断；还有，

· 怎样传送接收帧。

### 5.2.2.1 配置物理接口

配置物理接口包括决定哪个以太网接口应该被激活，和为连续接收激活接收逻辑。这可以通过 LineCTL 寄存器(寄存器 13)做到和在表 18 中有描述。

| 寄存器 13，LineCTL | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 6 | SerRxON | 置 1 后，接收被激活 |
| 8 | AUIonly | 置 1 后，AUI 被选中（优先于 AutoAUI/10BT） |
| 9 | AutoAUI/10BT | 置 1 后，激活自动接口选择。当 8 和 9 位清 0 后，选择了 10BASE-T |
| E | LoRx Squelch | 置 1 后，接收器静噪水平降低约 6 分贝 |

表格 18. 物理接口配置

### 5.2.2.2 选择接收哪个帧类型

RxCTL 寄存器(寄存器 5)用来决定 CS8900A 接收哪个帧类型（当接收帧被缓冲，在芯片或者通过 DMA 在主机内存上，这个帧就认为成功接收）。表 19 描述了这个寄存器的配置位。参考 86 页 5.3 节来获得目的地址过滤的细节描述。

| 寄存器 5，RxCTL | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 6 | IAHashA | 置 1 后，通过哈希过滤器的私有地址帧被接受 |
| 7 | PromiscuousA | 置 1 后，所有帧被接受 |
| 8 | RxOKA | 置 1 后，通过 DA 过滤器含有效长度和 CRC 的帧被接受 |
| 9 | MulticastA | 置 1 后，通过哈希过滤器的组播帧被接受 |
| A | IndividualA | 置 1 后，含有符合 PacketPage base+ 0158h 的 IA 的 DA 的帧被接受 |
| B | BroadcastA | 置 1 后，所有广播帧被接受 |
| C | CRCerrorA | 置 1 后，通过 DA 过滤器且含有错误 CRC 的帧被接受 |
| D | RuntA | 置 1 后，通过 DA 过滤器且短过 64 字节的帧被接受 |
| E | ExtradataA | 置 1 后，通过 DA 过滤器且长过 1518 字节的帧被接受（只有第一个 1518 字节被缓冲） |

表格 19. 帧接收标准

* 必须符合 bits 8，C，D，和 E 的编程标准.

### 5.2.2.3 选择哪个事件引发中断

RxCFG 寄存器(寄存器 3)和 BufCFG 寄存器 (寄存器 B)用来决定哪个接收事件引起主机处理器中断。表格 21 描述这些寄存器的中断使能(iE)位。

| 寄存器 3，RxCFG | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 8 | RxOKiE | 置 1 后，接收到有有效长度和 CRC 的帧后产生中断 |
| C | CRCerroriE | 置 1 后，接收到含错误 CRC 的帧后产生中断 |
| D | RuntiE | 置 1 后，接收到短过 64 字节的帧后产生中断 |
| E | ExtradataiE | 置 1 后，接收到长过 1518 字节的帧后产生中断 |

* 产生中断之前必须通过 DA 过滤器.

表格 20.

| 寄存器 B，RufCFG | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 7 | RxDMAiE | 置 1 后，如果有一个或多个帧通过 DMA 传送就产生中断 |
| A | RxMissiE | 置 1 后，如果帧由于接收缓冲区空间不够而丢弃就产生中断 |
| B | Rx128iE | 置 1 后，如果接收数据的第一个 128 字节被缓冲就产生中断 |
| D | MissOvfloiE | 置 1 后，如果 RxMISS 遇到溢出就产生中断 |
| F | RxDestiE | 置 1 后，如果传入的帧的 DA 被缓冲就产生中断 |

表格 21. 寄存器 3 和 B 中断配置

### 5.2.2.4 选择如何传送帧

RxCFG 寄存器(寄存器 3)和 BusCTL 寄存器(寄存器 17)用来决定帧怎样被传送到主机内存，正如表格 22 的描述。

| 寄存器 3，RxCFG | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 7 | StreamE | 置 1 后，使能流发送器 |
| 9 | RxDMAonly | 置 1 后，DMA 从操作用于所有接收帧 |
| A | AutoRX DMAE | 置 1 后，使能自动转换 DMA |

| | | |
|---|---|---|
| B | BufferCRC | 置1后，缓冲接收到的CRC |
| 寄存器17，BusCTL | | |
| 位 | 位名字 | 操作 |
| B | DMABurst | 置1后，DMA操作占用总线约28 μs。清0后，DMA操作持续。 |
| D | RxDMAsize | 置1后，DMA缓冲区大小是64 Kbytes。清0后，是16 Kbytes。 |

表22. 接收帧预处理

## 5.2.3 接收帧预处理

CS8900A有四个步骤预处理所有接收帧：

1) 目的地址过滤；

2) 早期中断产生；

3) 验收滤波；和，

4) 正常中断产生。

图21提供帧的预处理图。

### 5.2.3.1 目标地址过滤

所有进来的帧都通过目标地址过滤器(DA过滤器)。如果帧的DA通过DA过滤器，帧就会传到下一个预处理。如果它通不过DA过滤器，帧就被丢弃。请看86页5.3节来获得更多DA过滤器的细节描述。

### 5.2.3.2 早期中断产生

CS8900A支持以下两个用来通知主机收到帧的早期中断：

- RxDest：当传入的帧的目的地址(DA)通过DA过滤器，RxDest位(寄存器 C, BufEvent,Bit F)就置为1。如果RxDestiE位(寄存器B, BufCFG, bit F) 置为1，CS8900A产生相应的中断。一旦RxDest置为1，主机被允许读取传入的帧的DA（帧的第一个6字节）。

- Rx128：当传入的帧的第一个128字节已经被接受Rx128位(寄存器 C, BufEvent, Bit B)就置为1。如果Rx128iE位(寄存器B, BufCFG, bit B) 置为1，CS8900A产生相应的中断。一旦Rx128位置为1，RxDest位清0并且主机被允许

读进来的帧的第一个 128 字节。Rx128 位会在主机读 BufEvent 寄存器(直接地或通过中断状态队列)时或者 CS8900A 检测到帧结束(EOF)序列时清 0。

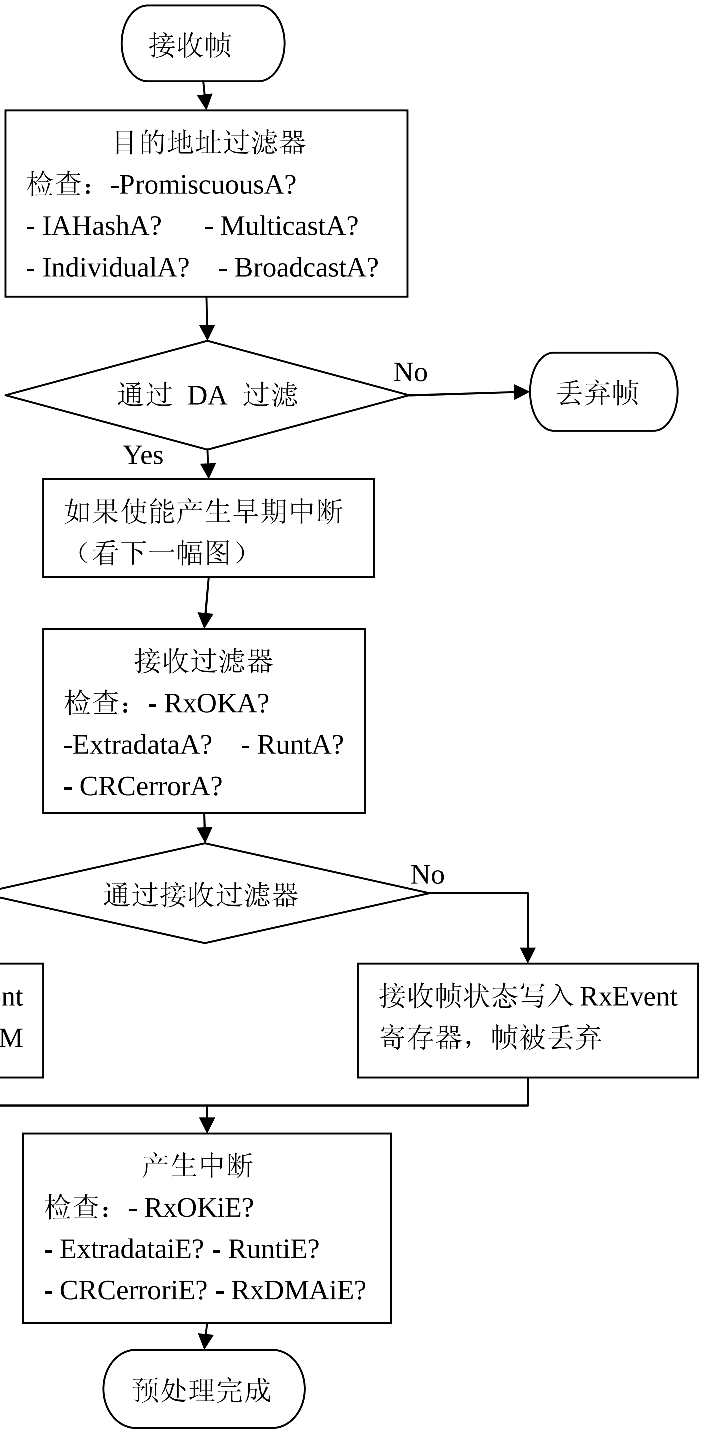

图 21. 接收帧预处理

像所有的事件比特位，RxDest 和 Rx128 当适当事件发生时由 CS8900A 置 1。不像别的事件比特位，RxDest 和 Rx128 可能被 CS8900A 清 0 而无需主机介入。所有别的事件位清 0 只当主机读适当的事件寄存器，直接地或通过中断状态队列

(ISQ)。(RxDest 和 Rx128 也可以在主机读 BufEvent 寄存器时清 0，直接地或通过中断状态队列)。图 22 通过一个早期中断处理过程图。

#### 5.2.3.3 接受过滤

预处理第 3 步是借助比较帧和 RxCTL 寄存器(寄存器 5)设置标准来决定是否接收帧。如果接收帧通过接收过滤器的检查，帧被缓冲，或者在芯片上或者通过 DMA 传到主机内存。如果帧没能通过接收过滤器，它被丢弃。接收过滤器的结果被报告到 RxEvent 寄存器(寄存器 4)。

#### 5.2.3.4 正常中断产生

预处理最后一步是产生任何使能了的并且被传入的帧触发的中断。中断在整一个帧被缓冲时产生(最多 1518 个字节)。要得到更多中断产生的信息，请看 78 页 5.1 节。

### 5.2.4 被持有的和 DMA 方式接收帧的比较

所有接受帧或者在片上持有直到被主机处理，或者通过 DMA 存储到主机内存。一个保存在片上 RAM 的接收帧叫做被持有的接收帧。一个通过 DMA 保存在主机内存的接收帧叫做 DMA 方式接收帧。这一章节描述缓冲和转发被持有的接收帧。89 页 5.4 节到 95 页 5.6 节描述 DMA 方式接收帧。

### 5.2.5 缓冲被持有的接收帧

如果空间可用，一个进来的帧会被临时放入片上 RAM 来等待主机的处理。虽然接收帧占用片上内存，CS8900A 不会提供内存空间给它直到以下两个情况之一出现：

1) 整个帧已经被接收并且主机已经直接地或通过 ISQ 读取 RxEvent 寄存器(寄存器 4)来了解帧。或者：

2) 帧部分被接收时引起 RxDest 位(寄存器 C, BufEvent, Bit F)或者 Rx128 位(寄存器 C, BufEvent, Bit B) 置 1，并且主机已经直接地或通过 ISQ 读取 BufEvent

寄存器(寄存器 C)来了解帧。

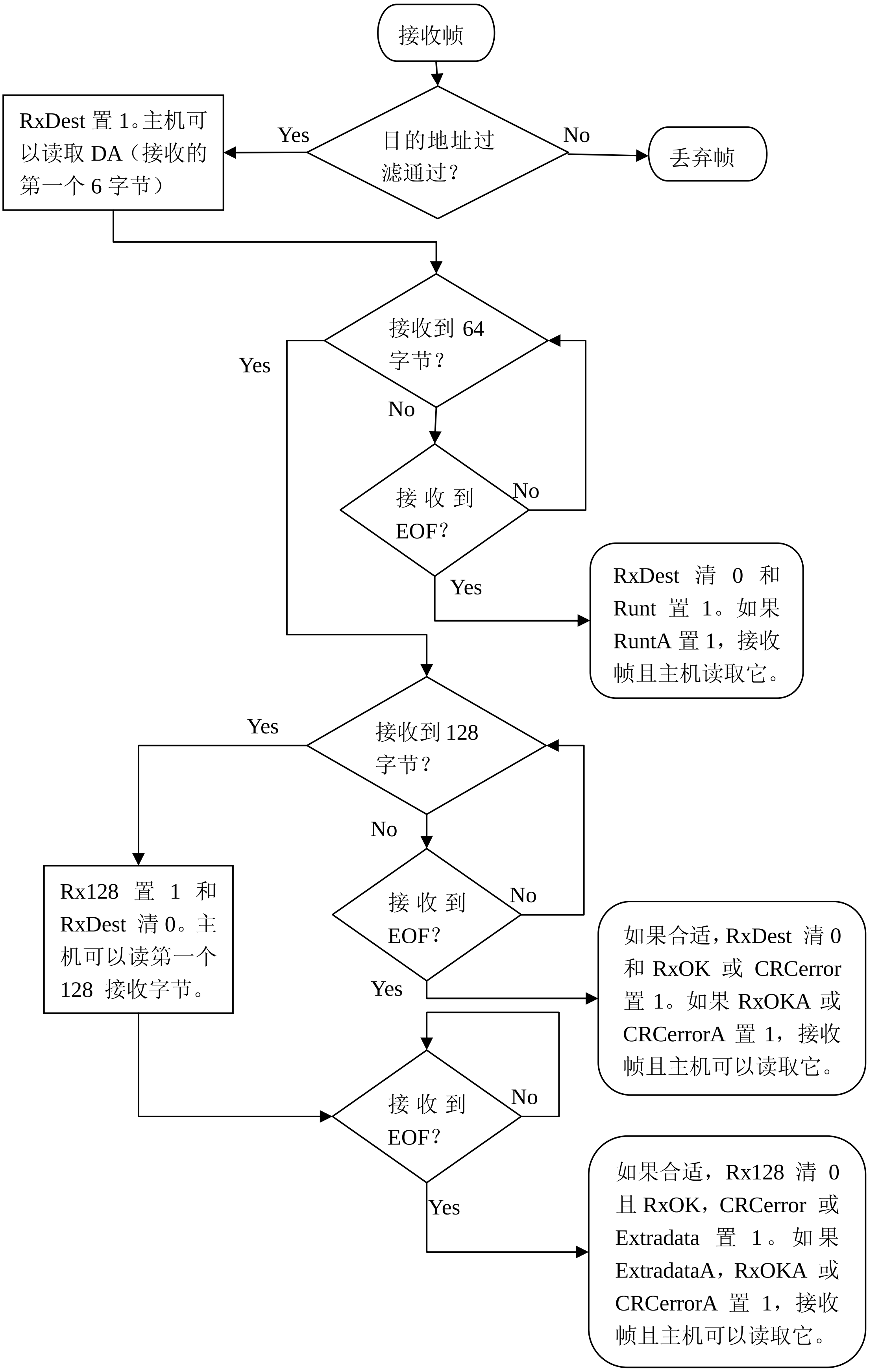
接收帧
目的地址过滤通过？
Yes
No
RxDest 置 1。主机可以读取 DA（接收的第一个 6 字节）
丢弃帧
接收到 64 字节？
Yes
No
接 收 到 EOF？
No
Yes
RxDest 清 0 和 Runt 置 1。如果 RuntA 置 1，接收帧且主机读取它。
接收到 128 字节？
Yes
No
Rx128 置 1 和 RxDest 清 0。主机可以读第一个 128 接收字节。
接 收 到 EOF？
No
Yes
如果合适，RxDest 清 0 和 RxOK 或 CRCerror 置 1。如果 RxOKA 或 CRCerrorA 置 1，接收帧且主机可以读取它。
接 收 到 EOF？
No
Yes
如果合适，Rx128 清 0 且 RxOK，CRCerror 或 Extradata 置 1。如果 ExtradataA，RxOKA 或 CRCerrorA 置 1，接收帧且主机可以读取它。

图 22. 早期中断产生

当CS8900A提供缓冲空间给某个的被持有的接收帧(被称为被保障的接收帧)，随后的帧没有数据被写入缓冲空间直到帧不被保障。(被保障的接收帧可以是无差错的也可以不是。)

在以下任一情况下一个帧不再被保障：

1) 主机按照接收顺序读整一个帧(写入一个字节，读出一个字节)。或者：

2) 主机读取或不读帧的某部分时，通过设置 Skip_1 位(寄存器 3, RxCFG, bit 6)来写入 Skip 命令。或者：

3) 主机直接地或通过 ISQ 读取帧的某部分和读取 RxEvent 寄存器(寄存器 5)，并了解别的接收帧。这情况叫做“隐式跳过”。要保证主机不会“隐式跳过”。

当被保障的接收帧在等待主机来处理时那两个早期中断会失效。

### 5.2.6 转发被持有的接收帧

在内存空间或 I/O 空间下主机可以读出被持有的接收帧。在内存空间下转发帧，主机执行重复的 Move 指令(REP MOVS)从 PacketPage base +0404h 下读取。在 I/O 空间下转发帧，主机执行重复的 In 指令(REP IN) 从 I/O base +0000h 下读取，还有帧的状态和长度处理。

有三个可能的方法主机可以知道某个帧的状态。可以：

1) 读中断状态队列；

2) 直接读 RxEvent 寄存器(寄存器 4)；或

3) 读 RxStatus 寄存器(PacketPage base +0400h )。

### 5.2.7 接收帧的可见性

每次只有一个接收帧对主机可见。接收帧的状态可以从 RxStatus 寄存器(PacketPage base + 0400h)读出，它的长度可以从 RxLength 寄存器(PacketPage base + 0402h)读出。要得到更多内存空间操作的信息，请看 73 页 4.9 节。要得到更多 I/O 空间操作的信息，请看 75 页 4.10 节。

### 5.2.8 内存模式接收操作的例子

短帧的普通长度是 64 字节，包括 4 字节 CRC。假设接收这个帧时 CS8900A 是这样配置的：

- BufferCRC 位(寄存器 3, RxCFG, Bit B)置 1 时，使得 4 字节 CRC 和剩下的接收数据一起被缓冲。
- RxOKA 位(寄存器 5, RxCTL, Bit 8)置 1 时，使得 CS8900A 会接收正确的帧(一个正确的帧是有合法长度和有效 CRC 的)。
- RxOKiE 位(寄存器 3, RxCFG, Bit 8)置 1 时，使得接收到正确帧时产生中断。

然后转发到主机是这样的：

1) CS8900A 对主机产生 RxOK 中断来报告有正确帧抵达。

2) 在 RxOK 置 1 前提下，主机读 ISQ(PacketPage base +0120h)来获得接收帧的状态和查看 RxEvent 寄存器(寄存器 4)的内容。

3) 主机读 RxLength 寄存器(PacketPage base +0402h)来得到接收帧长度。

4) 主机通过执行 32 次连续的 MOV 指令从 PacketPage base + 0404h 这里读帧数据。

64 字节帧的内存映射关系在表 23 里。

表 23. 内存映射举例

| 内存空间字偏移 | 片上 RAM 数据存储描述 |
|---|---|
| 0400h | RxStatus 寄存器(由于 RxEvent 被从 ISQ 读出主机可以跳过读 0400h) |
| 0402h | RxLength 寄存器(在这个例子长度是 40h 个字节，帧从 0404h 开始到 0443h) |
| 0404h to 0409h | 6 字节源地址 |
| 040Ah to 040Fh | 6 字节目的地址 |
| 0410h to 0411h | 2 字节长度或类型域 |
| 0412h to 043Fh | 46 字节数据 |
| 0440h | CRC，字节 1 和 2 |
| 0442h | CRC，字节 3 和 4 |

### 5.2.9 接收帧字节计数器

接收帧字节计数器描述当前接收帧的字节数。计数器随着从以太网接收到字节而实时地增加。字节计数器可以被驱动用来决定可以从 CS8900A 读出多少字节。用 I/O 模式或内存模式，和把 CPU 用在读计数器上，并且当以太网的帧被 CS8900A 接收的同时使用计数来从 CS8900A 读取帧可以达到最大以太网吞吐量(并行帧接收和帧读出任务)。

字节计数寄存器在 PacketPage base+ 50h。RxDest 或 Rx128 中断产生之后这个寄存器包含 CPU 可读的字节的数目。当到达帧尾，计数包含帧的最终计数值，包括 BufferCRC 选项导致的额外的值。当最终计数被 CPU 读出计数寄存器设为 0。因此可以使用字节计数寄存器来读完整的帧，寄存器可以被读取并且移动数据直到检测到计数为 0。然后 RxEvent 寄存器可以被读取来决定最终的帧状态。

顺序如下：

1) 在开始接收帧时，字节计数器和传入字符计数器是一致的。在帧接收结束前也相等。

2) 帧接收结束时，最后计数，包括 CRC 部分（如果使能 BufferCRC 选项），被保留到字节计数器被读取。

3) 当读取字节计数器的值为 0 时，前一个计数是最后值。这个计数也可能现在是一个奇怪的值。

4) Skip 命令后 RxEvent 寄存器应该被读出以获得帧的最终状态，来完成操作。

注意所有 RxEvent 的位应该在使用字节计数器前被处理。当 RxDest 或 Rx128 中断设置了，字节计数器应该在 BufEvent 有事件后被使用。

## 5.7 发送操作

### 5.7.1 概要

数据包发送发生在两个阶段。在第一个阶段，主机移动以太网帧到 CS8900A 的缓冲内存。第一阶段以主机填写发送命令为开始。这告诉 CS8900A 帧将要发送一个帧和何时（即 5，381，还是 1021 字节还是整个帧已经被传送到 CS8900A 后）

怎样发送帧（即含有还是不含 CRC，含有还是不含填充位，等）。填写发送命令后主机接着填写发送长度，表明需要多少缓冲空间。当缓冲空间可用，主机使用内存或 I/O 空间，把以太网帧写到 CS8900A 的内部存储器。在发送的第二阶段，CS8900A 转换帧为以太网数据包然后发送它到网络。第二阶段以 CS8900A 发送报文头和帧的开始分隔符当足够多的字节被传送到它的发送缓冲（5，381，1021 字节或整个帧，根据寄存器的配置）。报文头和帧的开始分隔符之后是主机传送到片上缓冲的数据（目的地址，源地址，长度域和 LLC 数据）。如果帧少于 64 字节，包括 CRC，CS8900A 如果有相应配置就会添加填充位。最后 CS8900A 添加相应的 32 位 CRC 值。

### 5.7.2 发送配置

每次重置，必须配置 CS8900A 的发送操作。使用相连 EEPROM 的可以自动做到这点，或者写配置命令到 CS8900A 的内部寄存器（请看 20 页 3.4 节）。必须被配置的项目包括使用哪个物理接口和哪个发送事件引起中断。

#### 5.7.2.1 配置物理接口

配置物理接口包括决定哪个以太网接口应该被激活(10BASE-T 还是 AUI)，并且使能串行发送的发送逻辑。配置物理接口由 LineCTL 寄存器（寄存器 13）完成和表 30 是它的描述。

表 30. 物理接口配置

| 寄存器 13，LineCTL | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 7 | SerTxON | 置 1 后，使能发送。 |
| 8 | AUIonly | 置 1 后，选择 AUI（优先于 AutoAUI/10BT）。清 0 后，选择 10BASE-T。 |
| 9 | AutoAUI/10BT | 置 1 后，使能自动接口选择。 |
| B | Mod BackoffE | 置 1 后，使用修改的退避算法。清 0 后，使用标准退避算法。 |
| D | 2-part DefDis | 置 1 后，两部分延时失效。 |

注意当现在没有接收到链路脉冲时，只有测试控制寄存器（寄存器 19）DisableLT 位置 1，CS8900A 在 10BASE-T 模式下才能发送。

#### 5.7.2.2 选择哪个事件产生中断

TxCFG寄存器(寄存器 7)和BufCFG寄存器(寄存器 B)用来决定哪个发送事件会引起主机处理器中断。表 31 和 32 描述这些寄存器的中断使能(iE)位。

### 5.7.3 更改配置

| 寄存器 7，TxCFG | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 6 | Loss-of-CRSiE | 置 1 后，当发送报文头后 CS8900A 检测载波监听失败就产生中断（只应用于 AUI）。 |
| 7 | SQErroriE | 置 1 后，出现 SQE 错误时产生中断。 |
| 8 | TxOKiE | 置 1 后，发送帧成功后产生中断。 |
| 9 | Out-of-windowiE | 置 1 后，检测到晚冲突就产生中断。 |
| A | JabberiE | 置 1 后，当出现发送时间过长就产生中断。 |
| B | AnycolliE | 置 1 后，任何冲突出现就产生中断。 |
| F | 16colliE | 置 1 后，CS8900A 尝试发送单一的帧 16 次就产生中断。 |

表 31. 发送时的中断配置

当主机配置了这些寄存器后在后续的包发送时就不用改变他们了。主机也可以随时修改 TxCFG 或 BufCFG 寄存器。修改后的效果会马上看到。即任何中断使能(iE)位的改变可以影响目前发送的包。

| 寄存器 B，BufCFG | | |
|---|---|---|
| 位 | 位名字 | 操作 |
| 8 | Rdy4TxiE | 置 1 后，当缓冲空间可用于发送帧就产生中断（和发送请求一起使用）。 |
| 9 | TxUnderruniE | 置 1 后，当发送开始后 CS8900A 耗尽数据，就产生中断。 |
| C | TxColOvfloiE | 置 1 后，TxCol 计数器溢出就产生中断。 |

表 32. 发送中断配置

如果主机要在初始化后改变 LineCTL 寄存器，ModBackoffE 位和任何接收相关位（LoRxSquelch，SerRxON）可以随时改变。但是，SerTxON 位置 1 后，不应该修改 Auto AUI/10BT 和 AUIonly 位。如果这三位之一将会再改变，主机应该把

SerTxON 位清 0(寄存器 13, LineCTL,Bit 7)，然后等到修改完成后把它置 1。

### 5.7.4 使能 CRC 生成和添加填充位

当主机提出发送请求命令，它必须申明循环冗余检验(CRC)值应该被添加到发送帧，和填充位是否要添加(如果需要)。表 33 描述怎样配置 CS8900A 的 CRC 生成和添加填充位。

| 寄存器 9，TxCMD | | |
|---|---|---|
| InhibitCRC<br>(Bit C) | TxPadDis<br>(Bit D) | 操作 |
| 0 | 0 | 必要的话在 64 字节后添加填充位(包含 CRC)。 |
| 1 | 0 | 发送短帧，如果指定的长度短于 60 字节。 |
| 0 | 1 | 必要的话在 60 字节后添加填充位(不包含 CRC)。 |
| 1 | 1 | 发送短帧，如果指定的长度短于 64 字节。CS8900A 不会发送短于 3 字节的帧。 |

表 33. CRC 和填充位配置

### 5.7.5 私有数据包发送

当主机有数据包发送，它必须向 CS8900A 提出发送请求包括如下以正确顺序显示的三个操作：

1) 主机必须写发送命令到 TxCMD 寄存器(PacketPage base +0144h)。TxCMD 寄存器的内容可以从 TxCMD 寄存器(寄存器 9)读出来。

2) 主机必须写帧长度到 TxLength 寄存器(PacketPage base + 0146h)。

3) 主机必须读 BusST 寄存器(寄存器 18)。

写入 TxCMD 的消息告诉 CS8900A 怎样发送下一个帧。这些必须写入 TxCMD 寄存器的位如表 34 里描述。

| 寄存器 9，TxCMD | | | |
|---|---|---|---|
| 位 | | 位名字 | 操作 |
| 6 | 7 | Tx Start | |

| | | | |
|---|---|---|---|
| 0 | 0 | | 传送 5 字节到 CS8900A 后发送报文头。 |
| 0 | 1 | | 传送 381 字节到 CS8900A 后发送报文头。 |
| 1 | 0 | | 传送 1021 字节到 CS8900A 后发送报文头。 |
| 1 | 1 | | 传送整个帧到 CS8900A 后发送报文头。 |
| 8 | | Force | 置 1 后，CS8900A 丢弃目前在发送缓冲的所有帧数据 |
| 9 | | Onecoll | 置 1 后，CS8900A 出现冲突后不会尝试再发送任何包。 |
| C | | InhibitCRC | 置 1 后，CS8900A 不会添加 32 位 CRC 值到所有发送包末尾。 |
| D | | TxPadDis | 置 1 后，CS8900A 不会为短帧添加填充位。 |

表 34. 发送命令配置

对于每个私有包的发送，主机必须提交完整的包发送请求。还有主机必须在每次包发送前写 TxCMD 寄存器，即使 TxCMD 寄存器的内容不再改变。以上描述的发送请求可以在内存空间或 I/O 空间。

### 5.7.6 以轮询模式发送

在轮询模式下，Rdy4TxiE 位(寄存器 B, BufCFG, Bit 8)必须清 0（中断失效）。发送操作如以下顺序发生和展示在图 30 里。

1) 主机写发送命令到 TxCMD 寄存器(memory base+ 0144h 在内存模式下和 I/O base + 0004h 在 I/O 模式下)来请求帧存储。

2) 主机写发送帧长度到 TxLength 寄存器(memory base+ 0146h 在内存模式下和 I/O base + 0006h 在 I/O 模式下)。如果发送长度有错误，命令被丢弃并且 TxBidErr 位(寄存器 18, BusST, Bit 7)置 1。

3) 主机读 BusST 寄存器。在内存模式下是读寄存器 18，在 memory base + 0138h。在 I/O 模式下，主机必须先写 0138h 到 PacketPage 指针端口(I/O base + 000Ah)来设置 PacketPage 指针到正确位置。主机可以从 PacketPage 数据端口(I/O base + 000Ch)读 BusST 寄存器。

4) 读寄存器后，要检查 Rdy4TxNOW 位(Bit 8)。如果置 1，可以写帧数据。否者主机必须继续读 BusST 寄存器(寄存器 18)和检查 Rdy4TxNOW 位(Bit 8)直到它为 1。

当 CS8900A 准备好接受帧，主机使用“REP”指令(内存模式下 REP MOVS 从

memory base+ 0A00h 开始，和 I/O 模式下 REP OUT 到接收/发送数据端口(I/O base + 0000h) )从主机内存传送整个帧到 CS8900A 内存。

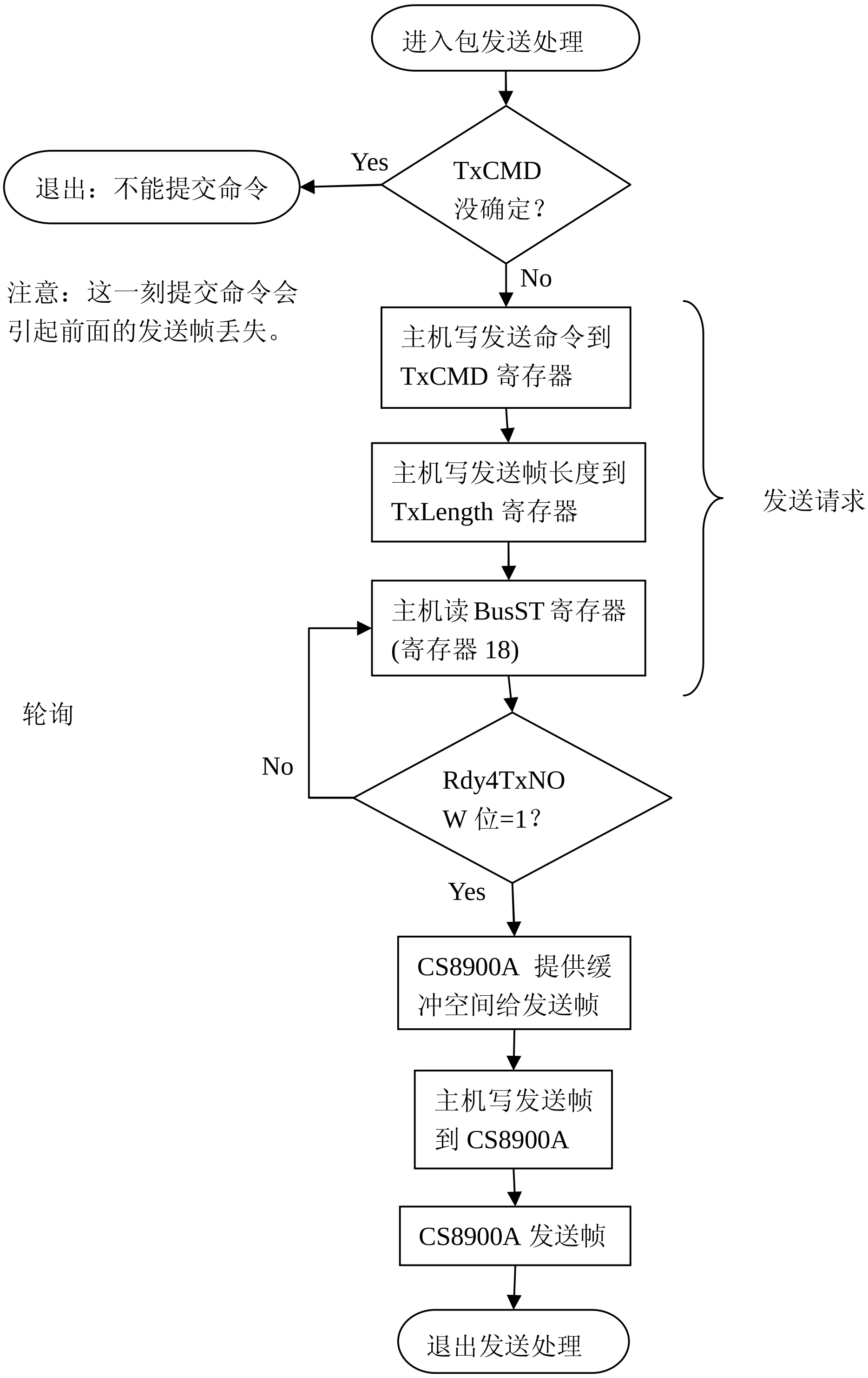

图 30. 轮询模式下的发送操作

### 5.7.7 以中断模式发送

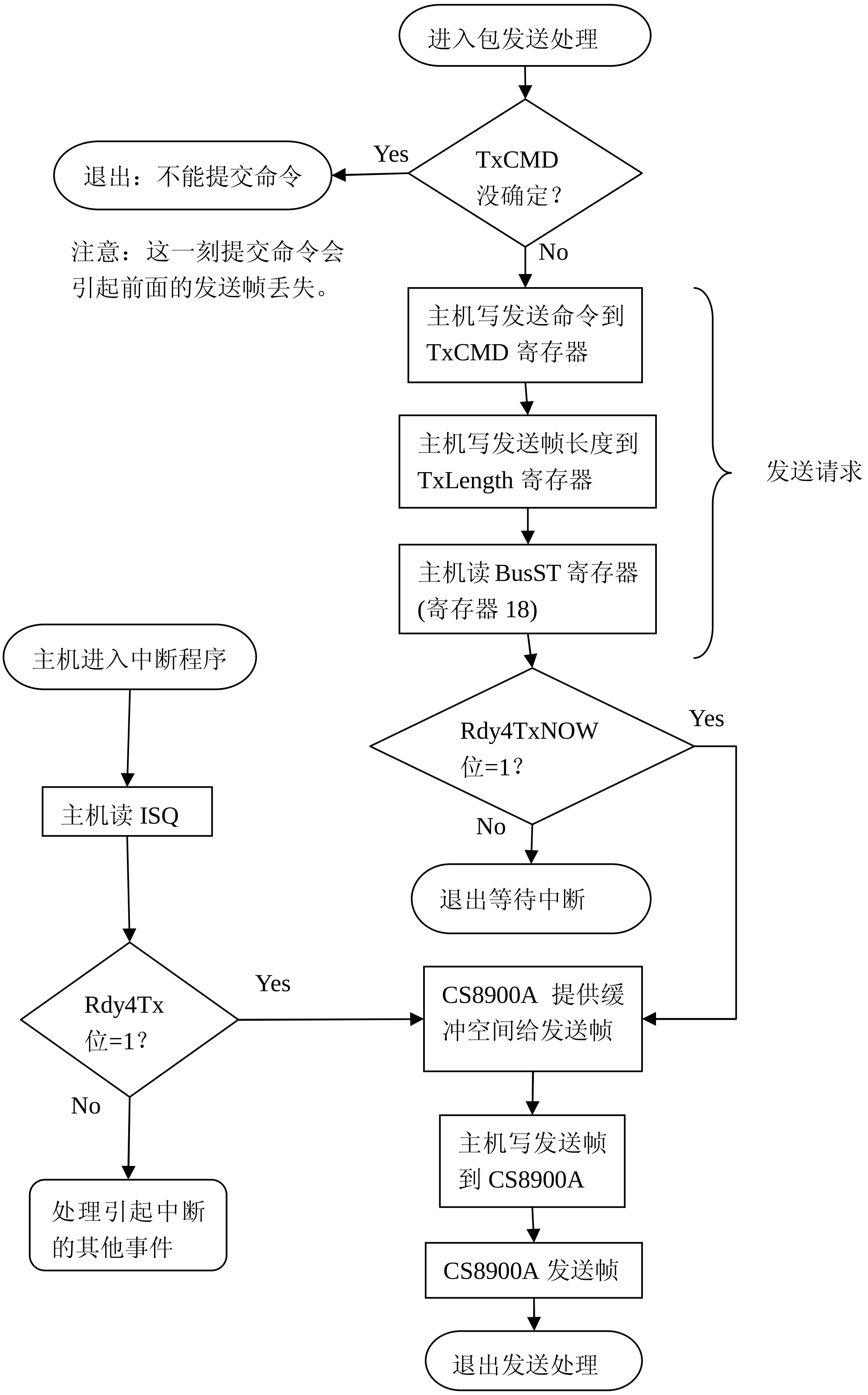

图 31. 中断模式下的发送操作

中断模式下，Rdy4TxiE 位(寄存器 B, BufCFG, Bit 8)必须置 1 来进行发送操作。

发送操作如以下顺序发生和展示在图 31 里。

1) 主机写发送命令到 TxCMD 寄存器(memory base+ 0144h 在内存模式下和 I/O base + 0004h 在 I/O 模式下)来请求帧存储。

2) 主机写发送帧长度到 TxLength 寄存器(memory base+ 0146h 在内存模式下和 I/O base + 0006h 在 I/O 模式下)。如果发送长度有错误，命令被丢弃并且 BusST 寄存器的 TxBidErr 位，bit 7，置 1。

3) 主机读 BusST 寄存器。在内存模式下是读寄存器 18，在 memory base + 0138h。I/O 模式下，主机必须先写 0138h 到 PacketPage 指针端口(I/O base + 000Ah)来设置 PacketPage 指针到正确位置，它然后能从 PacketPage 数据端口(I/O base + 000Ch)读 BusST 寄存器。然后检查 Rdy4TxNOW 位。如果置 1，可以写帧数据到 CS8900A 内存。如果 Rdy4TxNOW 清 0，主机必须等到 CS8900A 缓冲内存可用，这时主机会中断。一旦中断，主机进入中断服务程序和读 ISQ 寄存器(内存模式下 Memory base +0120h 和 I/O 模式下 I/O base + 0008h)来检查 Rdy4Tx 位(bit 8)。如果 Rdy4Tx 清 0，则 CS8900A 准备好接受帧。

4) 当 CS8900A 准备好接受帧，主机使用 REP 指令(内存模式下 REP MOVS 从 memory base+ 0A00h 开始，和 I/O 模式下 REP OUT 到接收/发送数据端口(I/O base + 0000h) )从主机内存传送整个帧到 CS8900A 内存。

### 5.7.8 完成发送

当 CS8900A 成功完成发送一个帧，它把 TxOK 位(寄存器 8, TxEvent, Bit 8)置 1。如果 TxOKiE 位(寄存器 7, Tx CFG, bit 8)置 1，CS8900A 就产生相应的中断。

### 5.7.9 Rdy4TxNOW 和 Rdy4Tx 的比较

Rdy4TxNOW 位(寄存器 18, BusST, bit 8)用来告诉主机 CS8900A 准备好接受发送帧。当不使用中断时(即 Rdy4TxiE 位(寄存器 B, BufCFG, Bit 8)清 0)，这一位用来在发送请求过程或之后告诉主机空间可用。还有，Rdy4Tx 位需要 Rdy4TxiE 位置 1 才可以和中断一起使用。

图 30 提供无冲突的无错发送图。

### 5.7.10 为发送帧提供缓冲空间

当主机提交发送请求，CS8900A 检查发送帧长度看是否有足够片上缓冲空间。如果有，CS8900A 置 Rdy4TxNOW 位为 1。如果没有，并且 Rdy4TxiE 位置 1，CS8900A 等到缓冲空间释放然后置 Rdy4Tx 位为 1。如果 Rdy4TxiE 清 0，当空间可用 CS8900A 置 Rdy4TxNOW 位为 1。

尽管发送缓冲空间可能可用，CS8900A 不会为发送帧提供缓冲空间直到以下全成立：

1) 主机必须提交发送请求；

2) 发送请求必须成功；并且，

3) 主机读到 Rdy4TxNOW 位(寄存器 18, BusST, Bit 8)是置 1 的，或者主机读到 Rdy4Tx 位(寄存器 C, BufEvent, bit 8)是置 1 的。

如果 CS8900A 提供了缓冲空间给某个发送帧，它不会允许后面的帧写入缓冲空间只要那个发送帧还在被处理。

缓冲空间被分配后，随后帧被发送，除非以下之一发生：

1) 主机完全写入帧数据，但发送在以太网线失败。有三个这样的错误，并且由 TxEvent 寄存器(寄存器 8)的三个发送错误位：16coll，Jabber，或 Out-of-Window 标明。或：

2) 主机把 Force 位(寄存器 9, TxCMD, bit 8)置 1 来停止发送。这情况下，被处理的发送帧，和所有排队在片上内存马上要发送的帧，都被清空和不被发送。使用 Force 位时主机应该使 TxLength = 0。或：

3) 出现发送欠载，并且 TxUnderrun 位(寄存器 C, BufEvent, Bit 9)置 1

TxOK 位(寄存器 8, TxEvent, Bit 8)置 1 表明发送成功。

### 5.7.11 发送帧的长度

发送请求期间，发送帧的长度由写入 TxLength 寄存器(PacketPage base + 0146h)的值决定。发送帧的长度可以由 TxPadDis 位(寄存器 9, TxCMD, Bit D)和 InhibitCRC 位(寄存器 9, TxCMD, Bit C)的配置修改。表 35 定义这些位怎样影响发送帧的长度。还有，它展示了 CS8900A 会发送哪些帧。

| 寄存器 9, TxCMD | | 主机在 0146h 指定发送长度（按字节） | | | |
|---|---|---|---|---|---|
| TxPadDis (Bit D) | InhibitCRC (Bit C) | 3<TxLength <60 | 60<TxLength<1514 | 1514<TxLength< 1518 | TxLength >1518 |
| 0 | 0 | 填充到60和添加 CRC | 发送帧和添加CRC[一般模式] | 不发送 | 不发送 |
| 0 | 1 | 填充到60和不含 CRC | 发送帧和不含 CRC | 不含 CRC | 不发送 |
| 1 | 0 | 不填充和添加 CRC | 发送帧和添加 CRC | 不发送 | 不发送 |
| 1 | 1 | 不填充和和不含 CRC | 发送帧和不含 CRC | 不含 CRC | 不发送 |

注意：8.如果 TxPadDis 位清 0 和 InhibitCRC 置 1 并且命令 CS8900A 发送长度少于 60 字节的帧，CS8900A 添加填充位。

9. TxLength 少于 3 字节时 CS8900A 不会发送帧。

表 35. 发送帧长度

# CS8900A Product Data Sheet

# 3.0 FUNCTIONAL DESCRIPTION

## 3.1 Overview

During normal operation, the CS8900A performs two basic functions: Ethernet packet transmission and reception. Before transmission or reception is possible, the CS8900A must be configured.

### 3.1.1 Configuration

The CS8900A must be configured for packet transmission and reception at power-up or reset. Various parameters must be written into its internal Configuration and Control registers such as Memory Base Address; Ethernet Physical Address; what frame types to receive; and which media interface to use.Configuration data can either be written to the CS8900A by the host (across the ISA bus), or loaded automatically from an external EEPROM. Operation can begin after configuration is complete.

Section 3.3 on page 18 and Section 3.4 on page 20 describe the configuration process in detail.Section 4.4 on page 46 provides a detailed description of the bits in the Configuration and Control Registers.

### 3.1.2 Packet Transmission

Packet transmission occurs in two phases. In the first phase, the host moves the Ethernet frame into the CS8900A’s buffer memory. The first phase begins with the host issuing a Transmit Command.This informs the CS8900A that a frame is to be transmitted and tells the chip when to start transmission (i.e. after 5, 381, 1021 or all bytes have been transferred) and how the frame should be sent (i.e. with or without

CRC, with or without pad bits,etc.). The Host follows the Transmit Command with the Transmit Length, indicating how much buffer space is required. When buffer space is available, the host writes the Ethernet frame into the CS8900A's internal memory, either as a Memory or I/O space operation.

In the second phase of transmission, the CS8900Aconverts the frame into an Ethernet packet then transmits it onto the network. The second phase begins with the CS8900A transmitting the preamble and Start-of-Frame delimiter as soon as the proper number of bytes has been transferred into its transmit buffer (5, 381, 1021 bytes or full frame, depending on configuration). The preamble and Start-of-Frame delimiter are followed by the Destination Address, Source Address, Length field and LLC data (all supplied by the host). If the frame is less than 64 bytes, including CRC, the CS8900A adds pad bits if configured to do so. Finally, the CS8900A appends the proper 32-bit CRC value.

The Section 5.7 on page 98 provides a detailed description of packet transmission.

### 3.1.3 Packet Reception

Like packet transmission, packet reception occurs in two phases. In the first phase, the CS8900A receives an Ethernet packet and stores it in on-chip memory. The first phase of packet reception begins with the receive frame passing through the analog front end and Manchester decoder where Manchester data is converted to NRZ data. Next, the preamble and Start-of-Frame delimiter are stripped off and the receive frame is sent through the address filter. If the frame's Destination Address matches the criteria programmed into the address filter, the packet is stored in the CS8900A's internal memory. The CS8900A then checks the CRC, and depending on the configuration, informs the processor that a frame has been received.

In the second phase, the host transfers the receive frame across the ISA bus and into host memory.Receive frames can be transferred as Memory space operations, I/O space operations, or as DMA operations using host DMA. Also, the CS8900A provides the capability to switch between Memory or I/O operation and DMA operation by using

Auto-Switch DMA and StreamTransfer.

The Section 5.2 on page 78 through Section 5.6 on page 95 provide a detailed description of packet reception.

## 3.2 ISA Bus Interface

The CS8900A provides a direct interface to ISA buses running at clock rates from 8 to 11 MHz. Its on-chip bus drivers are capable of delivering 24 mA of drive current, allowing the CS8900A to drive the ISA bus directly, without added external "glue logic".

The CS8900A is optimized for 16-bit data transfers, operating in either Memory space, I/O space,or as a DMA slave.

Note that ISA-bus operation below 8 MHz should use the CS8900A’s Receive DMA mode to minimize missed frames. See Section 5.4 on page 89 for a description of Receive DMA operation.

### 3.2.1 Memory Mode Operation

When configured for Memory Mode operation, the CS8900A’s internal registers and frame buffers are mapped into a contiguous 4-Kbyte block of host memory, providing the host with direct access to the CS8900A’s internal registers and frame buffers. The host initiates Read operations by driving the MEMR pin low and Write operations by driving the MEMW pin low.

For additional information about Memory Mode, see Section 4.9 on page 73.

### 3.2.2 I/O Mode Operation

When configured for I/O Mode operation, the CS8900A is accessed through eight, 16-bit I/O ports that are mapped into sixteen contiguous I/O locations in the host system’s I/O space. I/O Mode is the default configuration for the CS8900A and is always enabled.

For an I/O Read or Write operation, the AEN pin must be low, and the 16-bit I/O address on the ISA System Address bus (SA0 - SA15) must match the address space of the CS8900A. For a Read, IOR must be low, and for a Write, IOW must be low.

For additional information about I/O Mode, see Section 4.10 on page 75.

### 3.2.3 Interrupt Request Signals

The CS8900A has four interrupt request output pins that can be connected directly to any four of the ISA bus Interrupt Request signals. Only one interrupt output is used at a time. It is selected during initialization by writing the interrupt number (0 to 3) into PacketPage Memory base + 0022h. Unused interrupt request pins are placed in a high-impedance state. The selected interrupt request pin goes high when an enabled interrupt is triggered. The pin goes low after the Interrupt Status Queue (ISQ) is read as all 0’s (see Section 5.1 on page 78 for a description of the ISQ).

Table 1 presents one possible way of connecting the interrupt request pins to the ISA bus that utilizes commonly available interrupts and facilitates board layout.

| CS8900A Interrupt Request Pin | ISA Bus Interrupt | PacketPage base + 0022h |
|---|---|---|
| INTRQ3 (Pin 35) | IRQ5 | 0003h |
| INTRQ0 (Pin 32) | IRQ10 | 0000h |
| INTRQ1 (Pin 31) | IRQ11 | 0001h |
| INTRQ2 (Pin 30) | IRQ12 | 0002h |

Table 1. Interrupt Assignments

### 3.2.4 DMA Signals

The CS8900A interfaces directly to the host DMA controller to provide DMA transfers of receive frames from CS8900A memory to host memory. The CS8900A has three pairs of DMA pins that can be connected directly to the three 16-bit DMA channels of the ISA bus. Only one DMA channel is used at a time. It is selected during initialization by writing the number of the desired channel (0, 1 or 2) into PacketPage Memory base + 0024h. Unused DMA pins are placed in a high-impedance state. The selected DMA request pin goes high when the CS8900A has received frames to transfer to the host memory via DMA. If the DMABurst bit (register 17, BusCTL, Bit B) is clear, the pin goes low after the DMA operation is complete. If the DMABurst bit is set, the pin goes low 32 µs after the start of a DMA transfer.

The DMA pin pairs are arranged on the CS8900A to facilitate board layout. Crystal recommends the configuration in Table 2 when connecting these pins to the ISA bus.

| CS8900A DMA Signal (Pin #) | ISA DMA Signal | PacketPage base + 0024h |
|---|---|---|
| DMARQ0 (Pin 15) | DRQ5 | 0000h |
| DMACK0 (Pin 16) | DACK5 | |
| DMARQ1 (Pin 13) | DRQ6 | 0001h |
| DMACK1 (Pin 14) | DACK6 | |
| DMARQ2 (Pin 11) | DRQ7 | 0002h |
| DMACK2 (Pin 12) | DACK7 | |

Table 2. DMA Assignments

For a description of DMA mode, see Section 5.4 on page 89.

## 3.3 Reset and Initialization

### 3.3.1 Reset

Seven different conditions cause the CS8900A to reset its internal registers and circuits.

#### 3.3.1.1 External Reset, or ISA Reset

There is a chip-wide reset whenever the RESET pin is high for at least 400 ns. During a chip-wide reset, all circuitry and registers in the CS8900A are reset.

#### 3.3.1.2 Power-Up Reset

When power is applied, the CS8900A maintains reset until the voltage at the supply pins reaches approximately 2.5 V. The CS8900A comes out of reset once Vcc is greater than approximately 2.5 V and the crystal oscillator has stabilized.

#### 3.3.1.3 Power-Down Reset

If the supply voltage drops below approximately 2.5 V, there is a chip-wide reset. The CS8900A comes out of reset once the power supply returns to a level greater than approximately 2.5 V and the crystal oscillator has stabilized.

#### 3.3.1.4 EEPROM Reset

There is a chip-wide reset if an EEPROM checksum error is detected (see Section 3.4 on page 20).

#### 3.3.1.5 Software Initiated Reset

There is a chip-wide reset whenever the RESET bit (Register 15, SelfCTL, Bit 6) is set.

#### 3.3.1.6 Hardware (HW) Standby or Suspend

The CS8900A goes though a chip-wide reset whenever it enters or exits either HW Standby mode or HW Suspend mode (see Section 3.7 on page 25 for more information about HW Standby and Suspend).

#### 3.3.1.7 Software (SW) Suspend

Whenever the CS8900A enters SW Suspend mode, all registers and circuits are reset except for the ISA I/O Base Address register (located at PacketPage base + 0020h) and the SelfCTL register (Register 15). Upon exit, there is a chip-wide reset (see Section 3.7 on page 25 for more information about SW Suspend).

### 3.3.2 Allowing Time for Reset Operation

After a reset, the CS8900A goes through a self configuration. This includes calibrating on-chip analog circuitry, and reading EEPROM for validity and configuration. Time required for the reset calibration is typically 10 ms. Software drivers should not access registers internal to the CS8900A during this time. When calibration is done, bit INITD in the Self Status Register (register 16) is set indicating that initialization is complete, and the SIBUSY bit in the same register is cleared indicating the EEPROM is no longer being read or programmed.

### 3.3.3 Bus Reset Considerations

The CS8900A reads 3000h from IObase+0Ah after the reset, until the software writes a non-zero value at IObase+0Ah. The 3000h value can be used as part of the CS8900A signature when the system scans for the CS8900A. See Section 4.10 on page 75.

After a reset, the ISA bus outputs INTRx and DMARQx are 3-Stated, thus avoiding any interrupt or DMA channel conflicts on the ISA bus at powerup time.

### 3.3.4 Initialization

After each reset (except EEPROM Reset), the CS8900A checks the sense of the EEDataIn pin to see if an external EEPROM is present. If EEDI is high, an EEPROM is present and the CS8900A automatically loads the configuration data stored in the EEPROM into its internal registers (see next section). If EEDI is low, an EEPROM is not present and the CS8900A comes out of reset with the default configuration shown in Table 3.

| PacketPage Address | Register Contents | Register Descriptions |
|---|---|---|
| 0020h | 0300h | I/O Base Address* |
| 0022h | XXXX XXXX XXXX X100 | Interrupt Number |
| 0024h | XXXX XXXX XXXX XX11 | DMA Channel |
| 0026h | 0000h | DMA Start of Frame Offset |
| 0028h | X000h | DMA Frame Count |
| 002Ah | 0000h | DMA Byte Count |
| 002Ch | XXX0 0000h | Memory Base Address |
| 0030h | XXX0 0000h | Boot PROM Base Address |
| 0034h | XXX0 0000h | Boot PROM Address Mask |
| 0102h | 0003h | Register 3 - RxCFG |

| | | |
|---|---|---|
| 0104h | 0005h | Register 5 - RxCTL |
| 0106h | 0007h | Register 7 - TxCFG |
| 0108h | 0009h | Register 9 - TxCMD |
| 010Ah | 000Bh | Register B - BufCFG |
| 010Ch | Undefined | Reserved |
| 010Eh | Undefined | Reserved |
| 0110h | Undefined | Reserved |
| 0112h | 0013h | Register 13 - LineCTL |
| 0114h | 0015h | Register 15 - SelfCTL |
| 0116h | 0017h | Register 17 – BusCTL |
| 0118h | 0019h | Register 19 – TestCTL |

* I/O base address is unaffected by Software Suspend mode.

Table 3. Default Configuration

A low-cost serial EEPROM can be used to store configuration information that is automatically loaded into the CS8900A after each reset (except EEPROM reset). The use of an EEPROM is optional.

The CS8900A operates with any of six standard EEPROM’s shown in Table 4.

| EEPROM Type | Size (16 bit words) |
|---|---|
| ‘C46 (non-sequential) | 64 |
| ‘CS46 (sequential) | 64 |
| ‘C56 (non-sequential) | 128 |
| ‘CS56 (sequential) | 128 |
| ‘C66 (non-sequential) | 256 |
| ‘CS66 (sequential) | 256 |

Table 4. Supported EEPROM Types

## 4.9 Memory Mode Operation

To configure the CS8900A for Memory Mode, the PacketPage memory must be mapped into a contiguous 4-kbyte block of host memory. The block must start at an X000h boundary, with the PacketPage base address mapped to X000h. When the CS8900A comes out of reset, its default configuration is I/O Mode. Once Memory Mode is selected, all of the CS8900A's registers can be accessed directly.

In Memory Mode, the CS8900A supports Standard or Ready Bus cycles without introducing additional wait states.

Memory moves can use MOVD (double-word transfers) as long as the CS8900A's memory base address is on a double word boundary. Since 286 processors don't support the MOVD instruction, word and byte transfers must be used with a 286.

| Description | Mnemonic | Read/Write | Location: PocketPage base + |
|---|---|---|---|
| Receive Status | RxStatus | Read-only | 0400h-0401h |
| Receive Length | RxLength | Read-only | 0402h-0403h |
| Receive Frame | RxFrame | Read-only | starts at 0404h |
| Transmit Frame | TxFrame | Write-only | starts at 0A00h |

Table 16. Receive/Transmit Memory Locations

### 4.9.1 Accesses in Memory Mode

The CS8900A allows Read/Write access to the internal PacketPage memory, and Read access of the optional Boot PROM. (See Section 3.7 on page 25 for a description of the optional Boot PROM.)A memory access occurs when all of the followingare true:

- The address on the ISA System Address bus (SA0 - SA19) is within the Memory space range of the CS8900A or Boot PROM.
- The CHIPSEL input pin is low.
- Either the MEMR pin or the MEMW pin is low.

### 4.9.2 Configuring the CS8900A for Memory Mode

There are two different methods of configuring the CS8900A for Memory Mode operation. One method allows the CS8900A's internal memory to be mapped anywhere within the host system's 24-bit memory space. The other method limits memory mapping to the first 1 Mbyte of host memory space.

General Memory Mode Operation: Configuring the CS8900A so that its internal memory can be mapped anywhere within host Memory space requires the following:

- a simple circuit must be added to decode the Latchable Address bus (LA20 - LA23) and the BALE signal.

- the host must configure the external logic with the correct address range as follows:

1) Check to see if the INITD bit (Register 16,SelfST, bit 7) is set, indicating that initialization is complete.

2) Check to see if the ELpresent bit (Register 16, SelfST, bit B) is set. This bit indicates that external logic for the LA bus decode is present.

3) Set the ELSEL bit of the EEPROM Command Register to activate the ELCS pin for use with the external decode circuit.

4) Configure the external logic serially.

- the host must write the memory base address into the Memory Base Address register (PacketPage base + 002Ch);
- the host must set the MemoryE bit (Register 17,BusCTL, Bit A); and
- the host must set the UseSA bit (Register 17,BusCTL, Bit 9).

Limiting Memory Mode to the First 1 Mbyte of Host Memory Space: Configuring the CS8900A so that its internal memory can be mapped only within the first 1 Mbyte of host memory space requires the following:

- the CHIPSEL pin must be tied low;
- the ISA-bus SMEMR signal must be connected to the MEMR pin;
- the ISA-bus SMEMW signal must be connected to the MEMW pin;
- the host must write the memory base address into the Memory Base Address register (PacketPage base + 002Ch);
- the host must set the MemoryE bit (Register 17,BusCTL, Bit A); and
- the host must clear the UseSA bit (Register 17,BusCTL, Bit 9).

### 4.9.3 Basic Memory Mode Transmit

Memory Mode transmit operations occur in the following order (using interrupts):

1) The host bids for storage of the frame by writing the Transmit Command to the TxCMD register (memory base + 0144h) and the transmit frame length to the TxLength register (memory base + 0146h). If the transmit length is erroneous, the command is discarded and the TxBidErr bit (Register 18, BusST, Bit 7) is set.

2) The host reads the BusST register (Register 18,memory base + 0138h). If the Rdy4TxNOW bit(Bit 8) is set, the frame can be written. If clear,the host must wait for CS8900A buffer memory to become available. If Rdy4TxiE (Register B,BufCFG, Bit 8) is set, the host will be interrupted when Rdy4Tx (Register C, BufEvent, Bit 8) becomes set.

3) Once the CS8900A is ready to accept the frame, the host executes repetitive memory-to-memory move instructions (REP MOVS) to memory base + 0A00h to transfer the entire frame from host memory to CS8900A memory.

For a more detailed description of transmit, see Section 5.7 on page 98.

### 4.9.4 Basic Memory Mode Receive

Memory Mode receive operations occur in the following order (interrupts used to signal the presence of a valid receive frame):

1) A frame is received by the CS8900A, triggering an enabled interrupt.

2) The host reads the Interrupt Status Queue(memory base + 0120h) and is informed of the receive frame.

3) The host reads RxStatus (memory base + 0400h) to learn the status of the receive frame.

4) The host reads RxLength (memory base + 0402h) to learn the frame’ s length.

5) The host reads the frame data by executing repetitive memory-to-memory move instructions(REP MOVS) from memory base + 0404h to transfer the entire frame from CS8900A memory to host memory.

For a more detailed description of receive, see Section 5.2 on page 78.

### 4.9.5 Polling the CS8900A in Memory Mode

If interrupts are not used, the host can poll the CS8900A to check if receive frames are present and if memory space is available for transmit. However, this is beyond the scope of this data sheet.

## 4.10 I/O Space Operation

In I/O Mode, PacketPage memory is accessed through eight 16-bit I/O ports that are mapped into 16 contiguous I/O locations in the host system' s I/O space. I/O Mode is the default configuration for the CS8900A and is always enabled. On power up, the default value of the I/O base address is set at 300h.(Note that 300h is typically assigned to LAN peripherals). The I/O base address may be changed to any available XXX0h location, either by loading configuration data from the EEPROM, or during system setup. Table 17 shows the CS8900A I/O Mode mapping.

| Offset | Type | Description |
|---|---|---|
| 0000h | Read/Write | Receive/Transmit Data (Port 0) |
| 0002h | Read/Write | Receive/Transmit Data (Port 1) |
| 0004h | Write-only | TxCMD(Transmit Command) |
| 0006h | Write-only | TxLength(Transmit Length) |
| 0008h | Read-only | Interrupt Status Queue |
| 000Ah | Read/Write | PacketPage Pointer |
| 000Ch | Read/Write | PacketPage Data (Port 0) |
| 000Eh | Read/Write | PacketPage Data (Port 1) |

Table 17. I/O Mode Mapping

### 4.10.1 Receive/Transmit Data Ports 0 and 1

These two ports are used when transferring transmit data to the CS8900A and receive data from the CS8900A. Port 0 is used for 16-bit operations and Ports 0 and 1 are used for 32-bit operations (lower-order word in Port 0).

### 4.10.2 TxCMD Port

The host writes the Transmit Command (TxCMD)to this port at the start of each transmit operation.The Transmit Command tells the CS8900A that the host has a frame to be transmitted, as well as how that frame should be transmitted. This port is mapped into PacketPage base + 0144h. See Register 9 in Section 4.4 on page 46 for more information.

### 4.10.3 TxLength Port

The length of the frame to be transmitted is written here immediately after the Transmit Command is written. This port is mapped into PacketPage base + 0146h.

### 4.10.4 Interrupt Status Queue Port

This port contains the current value of the Interrupt Status Queue (ISQ). The ISQ is located at PacketPage base + 0120h. For a more detailed description of the ISQ, see Section 5.1 on page 78.

### 4.10.5 PacketPage Pointer Port

The PacketPage Pointer Port is written whenever the host wishes to access any of the CS8900A’ s internal registers. The first 12 bits (bits 0 through B) provide the internal address of the target register to be accessed during the current operation. The next three bits (C, D, and E) are read-only and will always read as 011b. Any convenient value may be written to these bits when writing to the PacketPage Pointer Port. The last bit (Bit F) indicates whether or not the PacketPage Pointer should be auto-incremented to the next word location. Figure 18 shows the structure of the PacketPage Pointer.

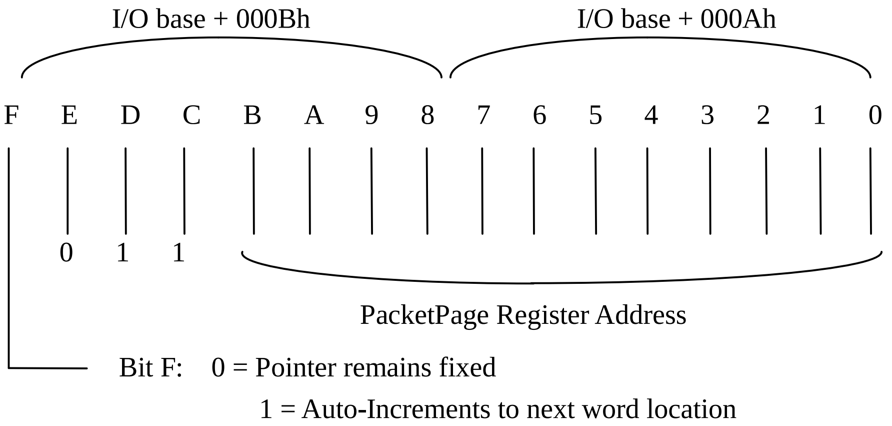
I/O base + 000Bh
I/O base + 000Ah
F E D C B A 9 8 7 6 5 4 3 2 1 0
0 1 1
PacketPage Register Address
Bit F: 0 = Pointer remains fixed
1 = Auto-Increments to next word location

Figure 18. PacketPage Pointer

### 4.10.6 PacketPage Data Ports 0 and 1

The PacketPage Data Ports are used to transfer data to and from any of the CS8900A’ s internal registers. Port 0 is used for 16-bit operations and Port 0 and 1 are used for 32-bit operations (lower-order word in Port 0).

### 4.10.7 I/O Mode Operation

For an I/O Read or Write operation, the AEN pin must be low, and the 16-bit I/O address on the ISA System Address bus (SA0 - SA15) must match the address space of the CS8900A. For a Read, the IOR pin must be low, and for a Write, the IOW pin must be low.

Note: The ISA Latchable Address Bus (LA17 - LA23) is not needed for applications that use only I/O Mode and Receive DMA operation.

### 4.10.8 Basic I/O Mode Transmit

I/O Mode transmit operations occur in the following order (using interrupts):

1) The host bids for storage of the frame by writing the Transmit Command to the TxCMD Port(I/O base + 0004h) and the transmit frame length to the TxLength Port (I/O base + 0006h).

2) The host reads the BusST register (Register 18)to see if the Rdy4TxNOW bit (Bit 8) is set. To read the BusST register, the host must first set the PacketPage Pointer at the correct location by writing 0138h to the PacketPage Pointer Port (I/O base + 000Ah). It can then read the BusST register from the PacketPage Data Port(I/O base + 000Ch). If Rdy4TxNOW is set, the frame can be written. If clear, the host must wait for CS8900A buffer memory to become available. If Rdy4TxiE (Register B, BufCFG, Bit 8) is set, the host will be interrupted when Rdy4Tx (Register C, BufEvent, Bit 8) becomes set. If the TxBidErr bit (Register 18, BusST, Bit 7) is set, the transmit length is not valid.

3) Once the CS8900A is ready to accept the frame, the host executes repetitive write instructions (REP OUT) to the Receive/Transmit Data Port (I/O base + 0000h) to transfer the entire frame from host memory to CS8900A memory.

For a more detailed description of transmit, see Section 5.7 on page 98.

### 4.10.9 Basic I/O Mode Receive

I/O Mode receive operations occur in the following order (In this example, interrupts are enabled to signal the presence of a valid receive frame):

1) A frame is received by the CS8900A, triggering an enabled interrupt.

2) The host reads the Interrupt Status Queue Port(I/O base + 0008h) and is informed of the receive frame.

3) The host reads the frame data by executing repetitive read instructions (REP IN) from the Receive/Transmit Data Port (I/O base + 0000h) to transfer the frame from CS8900A memory to host memory. Preceding the frame data are the contents of the RxStatus register (PacketPage base + 0400h) and the RxLength register (PacketPage base + 0402h).

For a more detailed description of receive, see Section 5.2 on page 78.

### 4.10.10 Accessing Internal Registers

To access any of the CS8900A’ s internal registers in I/O Mode, the host must first setup the PacketPage Pointer. It does this by writing the PacketPage address of the target register to the PacketPage Pointer Port (I/O base + 000Ah). The contents of the target register is then mapped into the PacketPage Data Port (I/O base + 000Ch).

If the host needs to access a sequential block of registers, the MSB of the PacketPage address of the first word to be accessed should be set to "1". The PacketPage Pointer will then move to the next word location automatically, eliminating the need to setup the PacketPage Pointer between successive accesses (see Figure 18).

### 4.10.11 Polling the CS8900A in I/O Mode

If interrupts are not used, the host can poll the CS8900A to check if receive frames are present and if memory space is available for transmit.

# 5.2 Basic Receive Operation

#### 5.2.0.1 Overview

Once an incoming packet has passed through the analog front end and Manchester decoder, it goes through the following three-step receive process:

1) Pre-Processing

2) Temporary Buffering

3) Transfer to Host

Figure 20 shows the steps in frame reception.

As shown in the figure, all receive frames go through the same pre-processing and temporary buffering phases, regardless of transfer method.

Once a frame has been pre-processed and buffered,it can be accessed by the host in either Memory or I/O space. In addition, the CS8900A can transfer receive frames to host memory via host DMA. This section describes receive frame pre-processing and Memory and I/O space receive operation. Section 5.4 on page 89 through Section 5.5 on page 92 describe DMA operation.

## 5.2.1 Terminology: Packet, Frame, and Transfer

The terms Packet, Frame, and Transfer are used extensively in the following sections. They are defined below for clarity:

#### 5.2.1.1 Packet

The term "packet" refers to the entire serial string of bits transmitted over an Ethernet network. This includes the preamble, Start-of-Frame Delimiter(SFD), Destination Address (DA), Source Address(SA), Length field, Data field, pad bits (if necessary), and Frame Check Sequence (FCS, also called CRC). Figure 9 shows the format of a packet.

### 5.2.1.2 Frame

The term "frame" refers to the portion of a packet from the DA to the FCS. This includes the Destination Address (DA), Source Address (SA), Length field, Data field, pad bits (if necessary), and Frame Check Sequence (FCS, also called CRC). Figure 9 shows the format of a frame. The term "frame data" refers to all the data from the DA to the FCS that is to be transmitted, or that has been received.

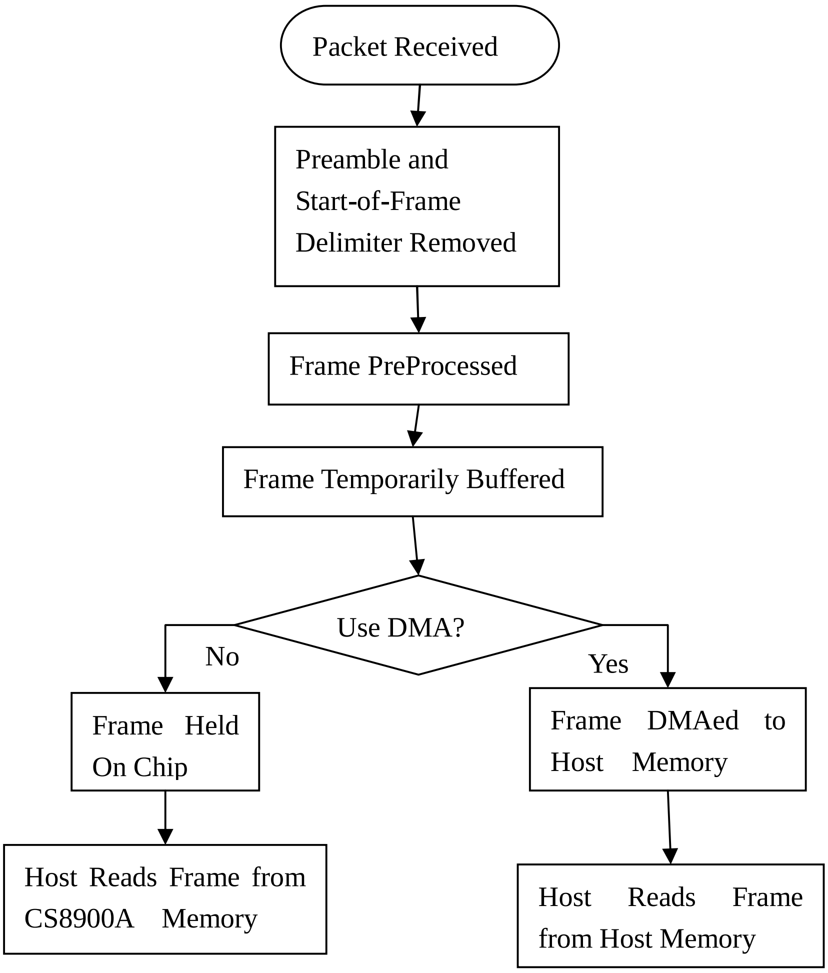
Packet Received
Preamble and Start-of-Frame Delimiter Removed
Frame PreProcessed
Frame Temporarily Buffered
Use DMA?
No
Yes
Frame Held On Chip
Frame DMAed to Host Memory
Host Reads Frame from CS8900A Memory
Host Reads Frame from Host Memory

Figure 20. Frame Reception

### 5.2.1.3 Transfer

The term "transfer" refers to moving data across the ISA bus, to and from the CS8900A. During receive operations, only frame data are transferred from the CS8900A to the host (the preamble and SFD are stripped off by the CS8900A’ s MAC engine). The FCS may or may not be transferred, depending on the configuration. All transfers to and from the CS8900A are counted in bytes, but may be padded for double word alignment.

## 5.2.2 Receive Configuration

After each reset, the CS8900A must be configured for receive operation. This can be done automatically using an attached EEPROM or by writing configuration commands to the CS8900A' s internal registers (see Section 3.4 on page 20). The items that must be configured include:

- which physical interface to use;
- which types of frames to accept;
- which receive events cause interrupts; and,
- how received frames are transferred.

### 5.2.2.1 Configuring the Physical Interface

Configuring the physical interface consists of determining which Ethernet interface should be active, and enabling the receive logic for serial reception. This is done via the LineCTL register(Register 13) and is described in Table18.

| Register 13, LineCTL | | |
|---|---|---|
| Bit | Bit Name | Operation |
| 6 | SerRxON | When set, reception enabled. |
| 8 | AUIonly | When set, AUI selected (takes precedence over AutoAUI/10BT). |
| 9 | AutoAUI/10BT | When set, automatic interface selection enabled. When both bits 8 and 9 are clear, 10BASE-T selected. |
| E | LoRx Squelch | When set, receiver squelch level reduced by approximately 6 dB. |

Table 18. Physical Interface Configuration

### 5.2.2.2 Choosing which Frame Types to Accept

The RxCTL register (Register 5) is used to determine which frame types will be accepted by the CS8900A (a receive frame is said to be "accepted" when the frame is buffered, either on chip or in host memory via DMA). Table 19 describes the configuration bits in this register. Refer to Section 5.3 on page 86 for a detailed description of Destination Address filtering.

| Register 5, RxCTL | | |
|---|---|---|
| Bit | Bit Name | Operation |
| 6 | IAHashA | When set, Individual Address frames that pass the hash filter are accepted*. |
| 7 | PromiscuousA | When set, all frames are accepted*. |
| 8 | RxOKA | When set, frames with valid length and CRC and that pass the DA filter are accepted. |
| 9 | MulticastA | When set, Multicast frames that pass the hash filter are accepted*. |
| A | IndividualA | When set, frames with DA that matches the IA at PacketPage base + 0158h are accepted*. |
| B | BroadcastA | When set, all broadcast frames are accepted*. |
| C | CRCerrorA | When set, frames with bad CRC that pass the DA filter are accepted. |
| D | RuntA | When set, frames shorter than 64 bytes that pass the DA filter are accepted. |
| E | ExtradataA | When set, frames longer than 1518 bytes that pass the DA filter are accepted (only the first 1518 bytes are buffered). |

* Must also meet the criteria programmed into bits 8, C, D, and E.

Table 19. Frame Acceptance Criteria

### 5.2.2.3 Selecting which Events Cause Interrupts

The RxCFG register (Register 3) and the BufCFG register (Register B) are used to determine which receive events will cause interrupts to the host processor. Table 21 describes the interrupt enable (iE) bits in these registers.

| Register 3, RxCFG | | |
|---|---|---|
| Bit | Bit Name | Operastion |
| 8 | RxOKiE | When set, there is an interrupt if a frame is received with valid length and CRC*. |
| C | CRCerroriE | When set, there is an interrupt if a frame is received with bad CRC*. |
| D | RuntiE | When set, there is an interrupt if a frame is received that is shorter than 64 bytes*. |
| E | ExtradataiE | When set, there is an interrupt if a frame is received that is longer than 1518 bytes*. |

* Must also pass the DA filter before there is an interrupt.

Table 20.

| Register B, BufCFG | | |
|---|---|---|
| Bit | Bit Name | Operastion |
| 7 | RxDMAiE | When set, there is an interrupt if one or more frames are transferred via DMA. |
| A | RxMissiE | When set, there is an interrupt if a frame is missed due to insufficient |

| | | |
|---|---|---|
| | | receive buffer space. |
| B | Rx128iE | When set, there is an interrupt after the first 128 bytes of receive data have been buffered. |
| D | MissOvfloiE | When set, there is an interrupt if the RxMISS counter overflows. |
| F | RxDestiE | When set, there is an interrupt after the DA of an incoming frame has been buffered. |

Table 21. Registers 3 and B Interrupt Configuration

#### 5.2.2.4 Choosing How to Transfer Frames

The RxCFG register (Register 3) and the BusCTL register (Register 17) are used to determine how frames will be transferred to host memory, as described in Table 22.

| Register 3, RxCFG | | |
|---|---|---|
| Bit | Bit Name | Operastion |
| 7 | StreamE | When set, Stream Transfer enabled. |
| 9 | RxDMAonly | When set, DMA slave operation used for all receive frames. |
| A | AutoRX DMAE | When set, Auto-Switch DMA enabled. |
| B | BufferCRC | When set, the received CRC is buffered. |
| Register 17, BusCTL | | |
| Bit | Bit Name | Operastion |
| B | DMABurst | When set, DMA operations hold the bus for up to approximately 28 μs. When clear, DMA operations are continuous. |
| D | RxDMAsize | When set, DMA buffer size is 64 Kbytes. When clear, DMA buffer size is 16 Kbytes. |

Table 22. Receive Frame Pre-Processing

### 5.2.3 Receive Frame Pre-Processing

The CS8900A pre-processes all receive frames using a four step process:

1) Destination Address filtering;

2) Early Interrupt Generation;

3) Acceptance filtering; and,

4) Normal Interrupt Generation.

Figure 21 provides a diagram of frame pre-processing.

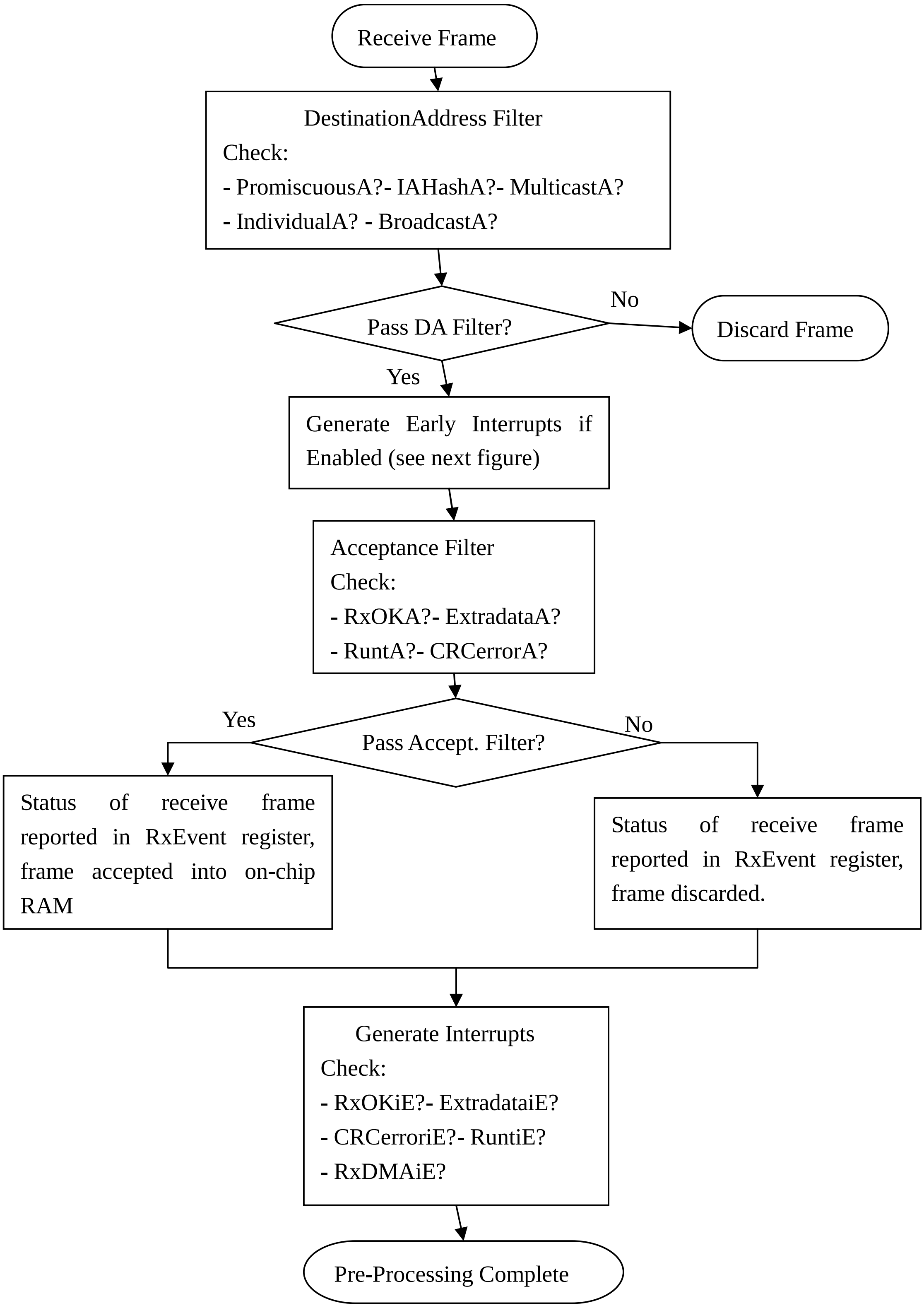
Receive Frame
DestinationAddress Filter
Check:
- PromiscuousA?- IAHashA?- MulticastA?
- IndividualA? - BroadcastA?
Pass DA Filter?
No
Discard Frame
Yes
Generate Early Interrupts if Enabled (see next figure)
Acceptance Filter
Check:
- RxOKA?- ExtradataA?
- RuntA?- CRCerrorA?
Yes
Pass Accept. Filter?
No
Status of receive frame reported in RxEvent register, frame accepted into on-chip RAM
Status of receive frame reported in RxEvent register, frame discarded.
Generate Interrupts
Check:
- RxOKiE?- ExtradataiE?
- CRCerroriE?- RuntiE?
- RxDMAiE?
Pre-Processing Complete

Figure 21. Receive Frame Pre-Processing

### 5.2.3.1 Destination Address Filtering

All incoming frames are passed through the Destination Address filter (DA filter).

If the frame’ s DA passes the DA filter, the frame is passed on for further pre-processing. If it fails the DA filter, the frame is discarded. See Section 5.3 on page 86 for a more detailed description of DA filtering.

#### 5.2.3.2 Early Interrupt Generation

The CS8900A support the following two early interrupts that can be used to inform the host that a frame is being received:

• RxDest: The RxDest bit (Register C, BufEvent,Bit F) is set as soon as the Destination Address(DA) of the incoming frame passes the DA filter. If the RxDestiE bit (Register B, BufCFG,bit F) is set, the CS8900A generates a corresponding interrupt. Once RxDest is set, the host is allowed to read the incoming frame's DA (the first 6 bytes of the frame).

• Rx128: The Rx128 bit (Register C, BufEvent,Bit B) is set as soon as the first 128 bytes of the incoming frame have been received. If the Rx128iE bit (Register B, BufCFG, bit B) is set,the CS8900A generates a corresponding interrupt. Once the Rx128 bit is set, the RxDest bit is cleared and the host is allowed to read the first 128 bytes of the incoming frame. The Rx128 bit is cleared by the host reading the BufEvent register (either directly or through the Interrupt Status Queue) or by the CS8900A detecting the incoming frame's End-of-Frame(EOF) sequence.

Like all Event bits, RxDest and Rx128 are set by the CS8900A whenever the appropriate event occurs. Unlike other Event bits, RxDest and Rx128 may be cleared by the CS8900A without host intervention. All other event bits are cleared only by the host reading the appropriate event register, either directly or through the Interrupt Status Queue(ISQ). (RxDest and Rx128 can also be cleared by the host reading the BufEvent register, either directly or through the Interrupt Status Queue). Figure 22 provides a diagram of the Early Interrupt process.

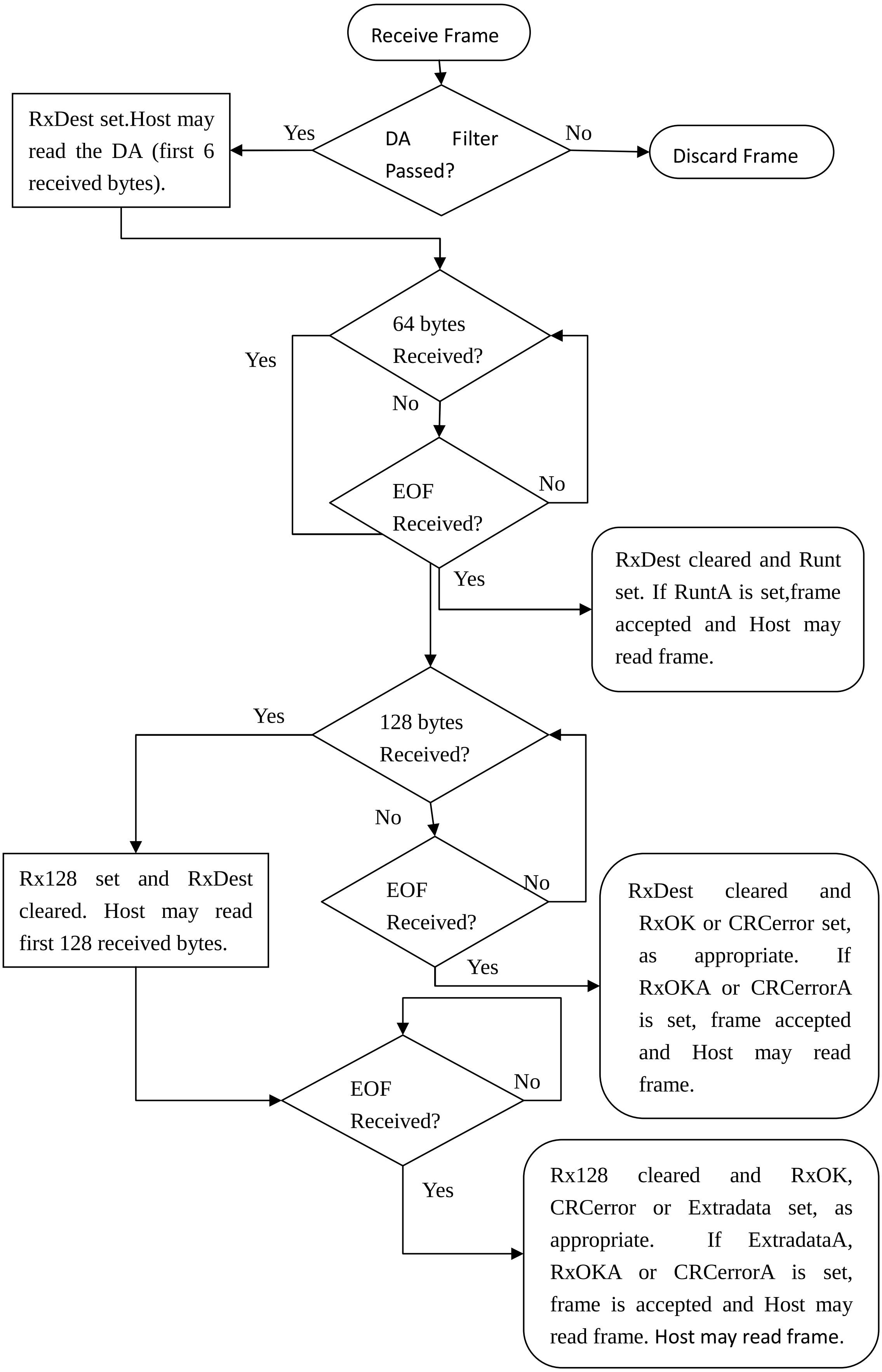
Receive Frame
DA Filter Passed?
Yes
No
RxDest set.Host may read the DA (first 6 received bytes).
Discard Frame
64 bytes Received?
Yes
No
EOF Received?
No
Yes
RxDest cleared and Runt set. If RuntA is set,frame accepted and Host may read frame.
128 bytes Received?
Yes
No
Rx128 set and RxDest cleared. Host may read first 128 received bytes.
EOF Received?
No
Yes
RxDest cleared and RxOK or CRCerror set, as appropriate. If RxOKA or CRCerrorA is set, frame accepted and Host may read frame.
EOF Received?
No
Yes
Rx128 cleared and RxOK, CRCerror or Extradata set, as appropriate. If ExtradataA, RxOKA or CRCerrorA is set, frame is accepted and Host may read frame. Host may read frame.

Figure 22. Early Interrupt Generation

#### 5.2.3.3 Acceptance Filtering

The third step of pre-processing is to determine whether or not to accept the frame by comparing the frame with the criteria programmed into the RxCTL register (Register 5). If the receive frame passes the Acceptance filter, the frame is buffered,either on chip or in host memory via DMA. If the frame fails the Acceptance filter, it is discarded.The results of the Acceptance filter are reported in the RxEvent register (Register 4).

#### 5.2.3.4 Normal Interrupt Generation

The final step of pre-processing is to generate any enabled interrupts that are triggered by the incoming frame. Interrupt generation occurs when the entire frame has been buffered (up to the first 1518 bytes). For more information about interrupt generation, see Section 5.1 on page 78.

### 5.2.4 Held vs. DMAed Receive Frames

All accepted frames are either held in on-chip RAM until processed by the host, or stored in host memory via DMA. A receive frame that is held in on-chip RAM is referred to as a held receive frame. A frame that is stored in host memory via DMA is a DMAed receive frame. This section describes buffering and transferring held receive frames. Section 5.4 on page 89 through Section 5.6 on page 95 describe DMAed receive frames.

### 5.2.5 Buffering Held Receive Frames

If space is available, an incoming frame will be temporarily stored in on-chip RAM, where it awaits processing by the host. Although this receive frame now occupies on-chip memory, the CS8900A does not commit the memory space to it until one of the following two conditions is true:

1) The entire frame has been received and the host has learned about the frame by reading the RxEvent register (Register 4), either directly or through the ISQ.

Or:

2) The frame has been partially received, causing either the RxDest bit (Register C, BufEvent, Bit F) or the Rx128 bit (Register C, BufEvent, Bit B) to become set, and the host has learned about the receive frame by reading the BufEvent register (Register C), either directly or through the ISQ.

When the CS8900A commits buffer space to a particular held receive frame (termed a committed received frame), no data from subsequent frames can be written to that buffer space until the frame is freed from commitment. (The committed received frame may or may not have been received error free.)

A received frame is freed from commitment by any one of the following conditions:

1) The host reads the entire frame sequentially in the order that it was received (first byte in, first byte out).

Or:

2) The host reads part or none of the frame, and then issues a Skip command by setting the Skip_1 bit (Register 3, RxCFG, bit 6).

Or:

3) The host reads part of the frame and then reads the RxEvent register (Register 5), either directly or through the ISQ, and learns of another receive frame. This condition is called an "implied Skip". Ensure that the host does not do "implied skips ".

Both early interrupts are disabled whenever there is a committed receive frame waiting to be processed by the host.

### 5.2.6 Transferring Held Receive Frames

The host can read-out held receive frames in Memory or I/O space. To transfer frames in Memory space, the host executes repetitive Move instructions (REP MOVS) from PacketPage base + 0404h. To transfer frames in I/O space, the host executes repetitive In instructions (REP IN) from I/O base + 0000h, with status and length preceding the frame.

There are three possible ways that the host can learn the status of a particular frame. It can:

1) Read the Interrupt Status Queue;

2) Read the RxEvent register directly (Register4);

or

3) Read the RxStatus register (PacketPage base + 0400h).

### 5.2.7 Receive Frame Visibility

Only one receive frame is visible to the host at a time. The receive frame's status can be read from the RxStatus register (PacketPage base + 0400h), and its length can be read from the RxLength register (PacketPage base + 0402h). For more information about Memory space operation, see Section 4.9 on page 73. For more information about I/O space operation, see Section 4.10 on page 75.

### 5.2.8 Example of Memory Mode Receive Operation

A common length for short frames is 64 bytes, including the 4-byte CRC. Suppose that such a frame has been received with the CS8900A configured as follows:

• The BufferCRC bit (Register 3, RxCFG, Bit B) is set causing the 4-byte CRC to be buffered with the rest of the receive data.

• The RxOKA bit (Register 5, RxCTL, Bit 8) is set, causing the CS8900A to accept good frames (a good frame is one with legal length and valid CRC).

• The RxOKiE bit (Register 3, RxCFG, Bit 8) is set, causing an interrupt to be generated whenever a good frame is received.

Then the transfer to the host would proceed as follows:

1) The CS8900A generates an RxOK interrupt to the host to signal the arrival of a good frame.

2) The host reads the ISQ (PacketPage base + 0120h) to assess the status of the receive frame and sees the contents of the RxEvent register(Register 4) with the RxOK bit (Bit 8) set.

3) The host reads the receive frame’s length from the RxLength register (PacketPage base + 0402h).

4) The host reads the frame data by executing 32 consecutive MOV instructions starting with PacketPage base + 0404h.

The memory map of the 64-byte frame is given in Table 23.

| Memory Space Word Offset | Description of Data Stored in Onchip RAM |
|---|---|
| 0400h | RxStatus Register (the host may skip reading 0400h since RxEvent was read from the ISQ.) |
| 0402h | RxLength Register (In this example, he length is 40h bytes. The frame starts at 0404h, and runs through 0443h.) |
| 0404h to 0409h | 6-byte Source Address. |
| 040Ah to 040Fh | 6-byte Destination Address. |
| 0410h to 0411h | 2-byte Length or Type Field. |
| 0412h to 043Fh | 46 bytes of data. |
| 0440h | CRC, bytes 1 and 2 |
| 0442h | CRC, bytes 3 and 4 |

Table 23. Example Memory Map

### 5.2.9 Receive Frame Byte Counter

The receive frame byte counter describes the number of bytes received for the current frame. The counter is incremented in real time as bytes are received from the Ethernet. The byte counter can be used by the driver to determine how many bytes are available for reading out of the CS8900A. Maximum Ethernet throughput can be achieved by using I/O or memory modes, and by dedicating the CPU to reading this counter, and using the count to read the frame out of the CS8900A at the same time it is being received by the CS8900A from the Ethernet (parallel frame-reception and frame-read-out tasks).

The byte count register resides at PacketPage base + 50h.

Following an RxDest or Rx128 interrupt the register contains the number of bytes which are available to be read by the CPU. When the end of frame is reached, the count contains the final count value for the frame, including the allowance for the BufferCRC option. When this final count is read by the CPU the count register is set to zero.

Therefore to read a complete frame using the byte count register, the register can be read and the data moved until a count of zero is detected. Then the RxEvent register can be read to determine the final frame status.

The sequence is as follows:

1) At the start of a frame, the byte counter matches the incoming character counter. The byte counter will have an even value prior to the end of the frame.

2) At the end of the frame, the final count, including the allowance for the CRC (if the BufferCRC option is enabled), is held until the byte counter is read.

3) When a read of the byte counter returns a count of zero, the previous count was the final count.The count may now have an odd value.

4) RxEvent should be read to obtain a final status of the frame, followed by a Skip command to complete the operation.

Note that all RxEvent’ s should be processed before using the byte counter. The byte counter should be used following a BufEvent when RxDest or Rx128 interrupts are enabled.

## 5.7 Transmit Operation

### 5.7.1 Overview

Packet transmission occurs in two phases. In the first phase, the host moves the Ethernet frame into the CS8900A’ s buffer memory. The first phase begins with the host issuing a Transmit Command.

This informs the CS8900A that a frame is to be transmitted and tells the chip when (i.e. after 5, 381, or 1021 bytes have been transferred or after the full frame has been transferred to the CS8900A) and how the frame should be sent (i.e. with or without CRC, with or without pad bits, etc.). The host follows the Transmit Command with the Transmit Length, indicating how much buffer space is required. When buffer space is available, the host writes the Ethernet frame into the CS8900A’ s internal memory, using either Memory or I/O space.

In the second phase of transmission, the CS8900A converts the frame into an Ethernet packet then transmits it onto the network. The second phase begins with the CS8900A transmitting the preamble and Start-of-Frame delimiter as soon as the proper number of bytes has been transferred into its transmit buffer (5, 381, 1021 bytes or full frame, depending on configuration). The preamble and Start-of-Frame delimiter are followed by the data transferred into the on-chip buffer by the host (Destination Address, Source Address, Length field and LLC data). If the frame is less than 64 bytes, including CRC, the CS8900A adds pad bits if configured to do so. Finally, the CS8900A appends the proper 32-bit CRC value.

## 5.7.2 Transmit Configuration

After each reset, the CS8900A must be configured for transmit operation. This can be done automatically using an attached EEPROM, or by writing configuration commands to the CS8900A’ s internal registers (see Section 3.4 on page 20). The items that must be configured include which physical interface to use and which transmit events cause interrupts.

### 5.7.2.1 Configuring the Physical Interface

Configuring the physical interface consists of determining which Ethernet interface should be active (10BASE-T or AUI), and enabling the transmit logic for serial transmission. Configuring the Physical Interface is accomplished via the LineCTL register (Register 13) and is described in Table 30.

| Register 13, LineCTL | | |
|---|---|---|
| Bit | Bit Name | Operation |
| 7 | SerTxON | When set, transmission enabled. |
| 8 | AUIonly | When set, AUI selected (takes precedence over AutoAUI/10BT). When clear, 10BASE-T selected. |
| 9 | AutoAUI/10BT | When set, automatic interface selection enabled. |
| B | Mod BackoffE | When set, the modified backoff algorithm is used. When clear, the standard backoff algorithm is used. |
| D | 2-part DefDis | When set, two-part deferral is disabled. |

Table 30. Physical Interface Configuration

Note that the CS8900A transmits in 10BASE-T mode when no link pulses are being received only if bit DisableLT is set in register Test Control (Register 19).

#### 5.7.2.2 Selecting which Events Cause Interrupts

The TxCFG register (Register 7) and the BufCFG register (Register B) are used to determine which transmit events will cause interrupts to the host processor. Tables 31 and 32 describe the interrupt enable (iE) bits in these registers.

### 5.7.3 Changing the Configuration

When the host configures these registers it does not need to change them for subsequent packet transmissions. If the host does choose to change the TxCFG or BufCFG registers, it may do so at any time. The effects of the change are noticed immediately. That is, any changes in the Interrupt Enable (iE) bits may affect the packet currently being transmitted.

| Register 7, TxCFG | | |
|---|---|---|
| Bit | Bit Name | Operation |
| 6 | Loss-of-CRSiE | When set, there is an interrupt whenever the CS8900A fails to detect Carrier Sense after transmitting the preamble (applies to the AUI only). |
| 7 | SQErroriE | When set, there is an interrupt whenever there is an SQE error. |
| 8 | TxOKiE | When set, there is an interrupt whenever a frame is transmitted successfully.. |
| 9 | Out-of-windowiE | When set, there is an interrupt whenever a late collision is detected. |
| A | JabberiE | When set, there is an interrupt whenever there is a jabber condition. |
| B | AnycolliE | When set, there is an interrupt whenever there is a collision. |
| F | 16colliE | When set, there is an interrupt whenever the CS8900A attempts to transmit a single frame 16 times. |

Table 31. Transmitting Interrupt Configuration

| Register B, BufCFG | | |
|---|---|---|
| Bit | Bit Name | Operation |
| 8 | Rdy4TxiE | When set, there is an interrupt whenever buffer space becomes available for a transmit frame (used with a Transmit Request). |
| 9 | TxUnderruniE | When set, there is an interrupt whenever the CS8900A runs out of data after transmit has started. |
| C | TxColOvfloiE | When set, there is an interrupt whenever the TxCol counter overflows. |

Table 32. Transmit Interrupt Configuration

If the host chooses to change bits in the LineCTL register after initialization, the ModBackoffE bit and any receive related bit (LoRxSquelch, SerRxON) may be changed at any time. However, the Auto AUI/10BT and AUIonly bits should not be changed while the SerTxON bit is set. If any of these three bits are to be changed, the host should first clear the SerTxON bit (Register 13, LineCTL, Bit 7), and then set it when the changes are complete.

### 5.7.4 Enabling CRC Generation and Padding

Whenever the host issues a Transmit Request command, it must indicate whether or not the Cyclic Redundancy Check (CRC) value should be appended to the transmit frame, and whether or not pad bits should be added (if needed). Table 33 describes how to configure the CS8900A for CRC generating and padding.

| 寄存器 9，TxCMD | | |
|---|---|---|
| InhibitCRC (Bit C) | TxPadDis (Bit D) | Operation |
| 0 | 0 | Pad to 64 bytes if necessary (including CRC). |
| 1 | 0 | Send a runt frame if specified length less than 60 bytes. |
| 0 | 1 | Pad to 60 bytes if necessary (without CRC). |
| 1 | 1 | Send runt if specified length less than 64. The CS8900A will not transmit a frame that is less than 3 bytes. |

Table 33. CRC and Paddling Configuration

### 5.7.5 Individual Packet Transmission

Whenever the host has a packet to transmit, it must issue a Transmit Request to the CS8900A consisting of the following three operations in the exact order shown:

1) The host must write a Transmit Command to the TxCMD register (PacketPage base + 0144h). The contents of the TxCMD register may be read back from the TxCMD register(Register 9).

2) The host must write the frame’ s length to the TxLength register (PacketPage base + 0146h).

3) The host must read the BusST register (Register 18)

The information written to the TxCMD register tells the CS8900A how to transmit the next frame. The bits that must be programmed in the TxCMD register are described in Table 34.

| Register 9, TxCMD | | | |
|---|---|---|---|
| Bit | | Bit Name | Operation |
| 6 | 7 | Tx Start | |
| 0 | 0 | | Start preamble after 5 bytes have been transferred to the CS8900A. |
| 0 | 1 | | Start preamble after 381 bytes have been transferred to the CS8900A. |
| 1 | 0 | | Start preamble after 1021 bytes have been transferred to the CS8900A. |
| 1 | 1 | | Start preamble after entire frame has been transferred to the CS8900A. |
| 8 | | Force | When set, the CS8900A discards any frame data currently in the transmit buffer. |
| 9 | | Onecoll | When set, the CS8900A will not attempt to retransmit any packet after a collision. |
| C | | InhibitCRC | When set, the CS8900A does not append the 32-bit CRC value to the end of any transmit packet. |
| D | | TxPadDis | When set, the CS8900A will not add pad bits to short frames. |

Table 34. Tx Command Configuration

For each individual packet transmission, the host must issue a complete Transmit Request. Furthermore, the host must write to the TxCMD register before each packet transmission, even if the contents of the TxCMD register does not change. The Transmit Request described above may be in either Memory Space or I/O Space.

### 5.7.6 Transmit in Poll Mode

In poll mode, Rdy4TxiE bit (Register B, BufCFG,Bit 8) must be clear (Interrupt Disabled). The transmit operation occurs in the following order and is shown in Figure 30.

1) The host bids for frame storage by writing the Transmit Command to the TxCMD register (memory base+ 0144h in memory mode and I/O base + 0004h in I/O mode).

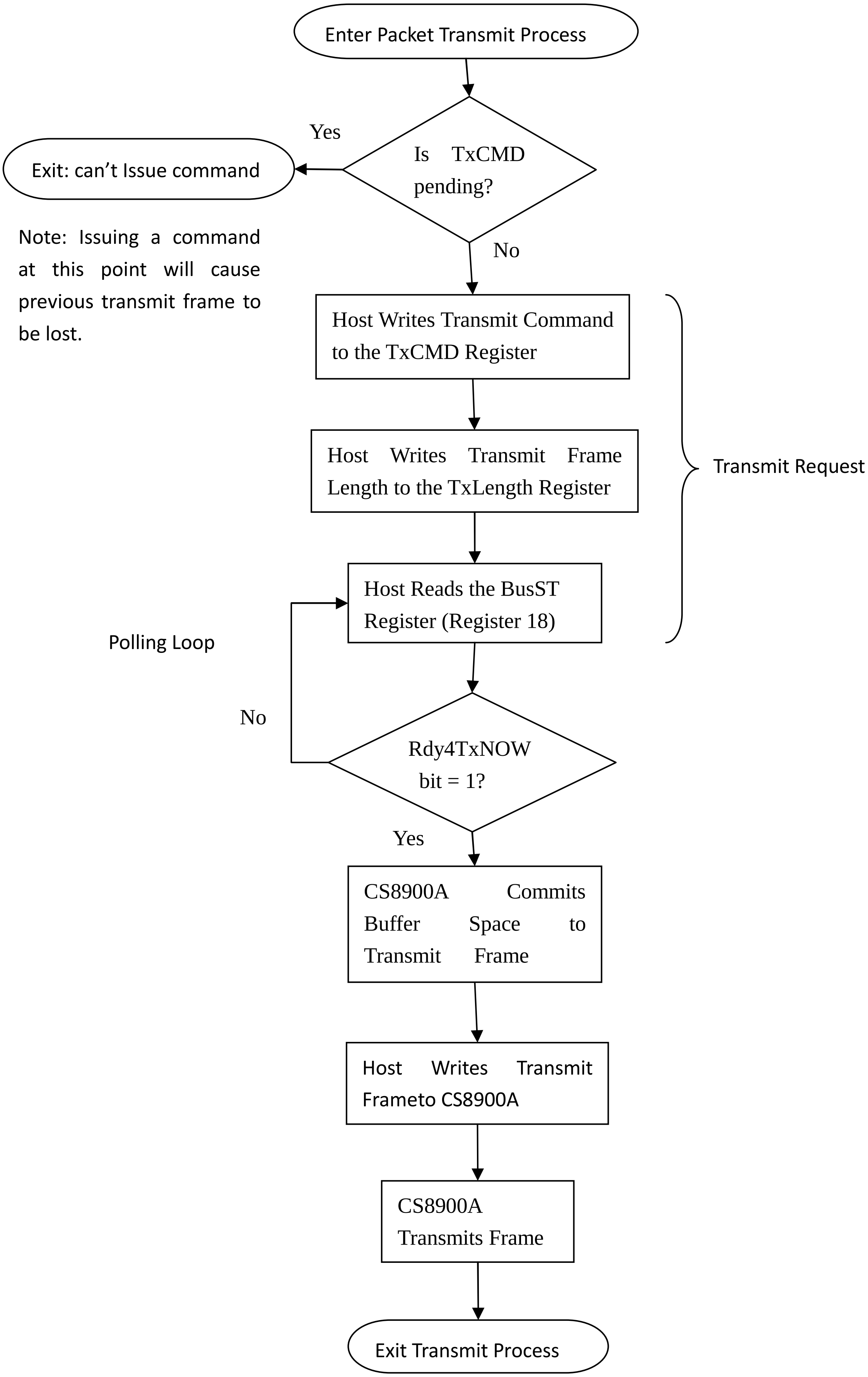
Enter Packet Transmit Process
Is TxCMD pending?
Yes
Exit: can't Issue command
Note: Issuing a command at this point will cause previous transmit frame to be lost.
No
Host Writes Transmit Command to the TxCMD Register
Host Writes Transmit Frame Length to the TxLength Register
Transmit Request
Host Reads the BusST Register (Register 18)
Polling Loop
No
Rdy4TxNOW bit = 1?
Yes
CS8900A Commits Buffer Space to Transmit Frame
Host Writes Transmit Frameto CS8900A
CS8900A Transmits Frame
Exit Transmit Process

Figure 30. Transmit Operation in Polling Mode

2) The host writes the transmit frame length to the TxLength register (memory base + 0146h in memory mode and I/O base + 0006h in I/O mode). If the transmit length is erroneous, the command is discarded and the TxBidErr bit (Register 18, BusST, Bit 7) is set.

3) The host reads the BusST register. This read is performed in memory mode by reading Register 18, at memory base + 0138h. In I/O mode, the host must first set the PacketPage Pointer at the correct location by writing 0138h to the PacketPage Pointer Port (I/O base + 000Ah). The host can then read the BusST register from the PacketPage Data Port (I/O base + 000Ch).

4) After reading the register, the Rdy4TxNOW bit (Bit 8) is checked. If the bit is set, the frame can be written. If the bit is clear, the host must continue reading the BusST register (Register 18) and checking the Rdy4TxNOW bit (Bit 8) until the bit is set.

When the CS8900A is ready to accept the frame,the host transfers the entire frame from host memory to CS8900A memory using “REP” instruction (REP MOVS starting at memory base + 0A00h in memory mode, and REP OUT to Receive/Transmit Data Port (I/O base + 0000h) in I/O mode).

### 5.7.7 Transmit in Interrupt Mode

In interrupt mode, Rdy4TxiE bit (Register B,BufCFG, Bit 8) must be set for transmit operation.Transmit operation occurs in the following order and is shown in Figure 31.

1) The host bids for frame storage by writing the Transmit Command to the TxCMD register (memory base + 0144h in memory mode and I/O base + 0004h in I/O mode).

2) The host writes the transmit frame length to the TxLength register (memory base + 0146h in memory mode and I/O base + 0006h in I/O mode). If the transmit length is erroneous, the command is discarded and the TxBidErr, bit 7,in BusST register is set.

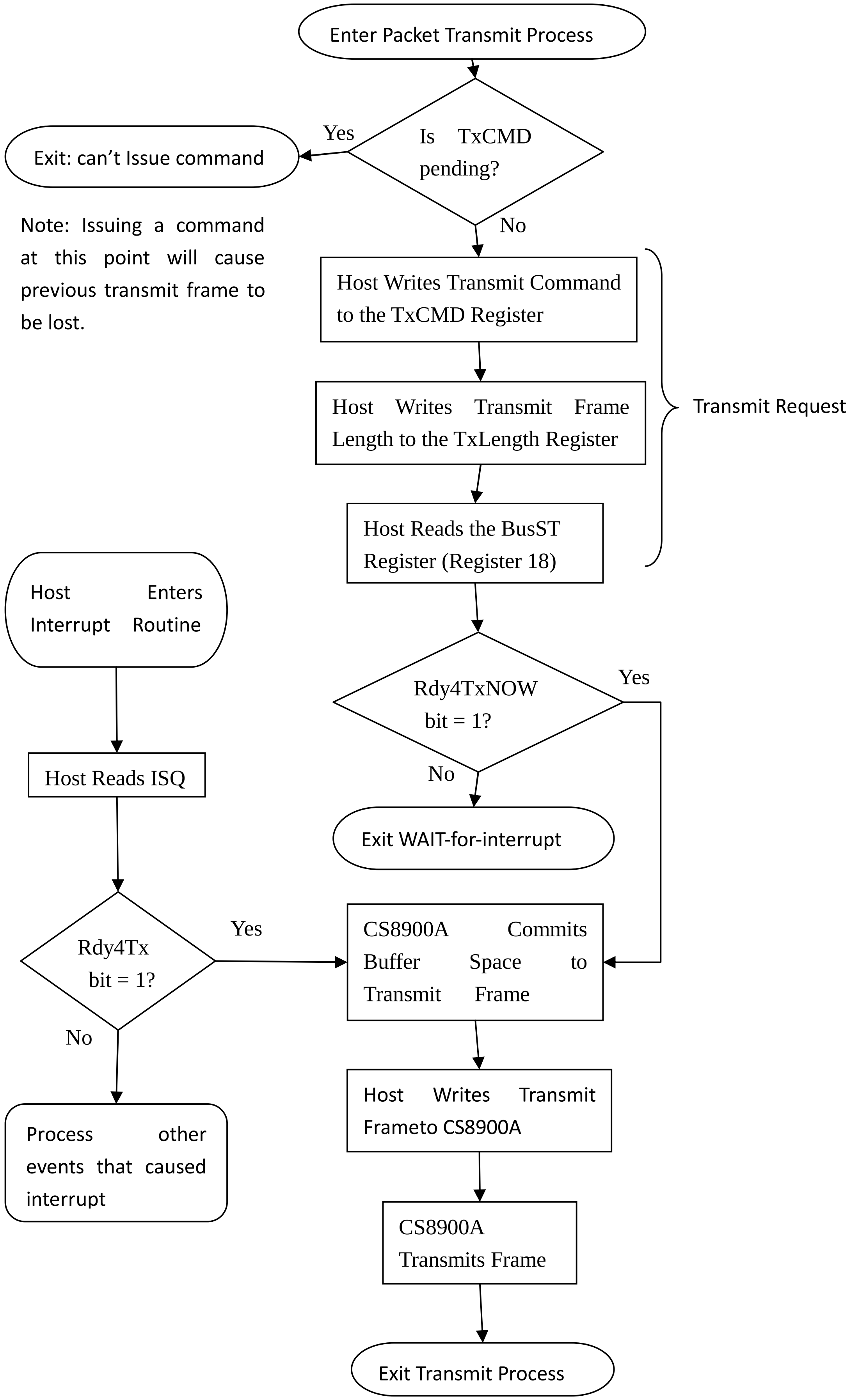
Enter Packet Transmit Process
Is TxCMD pending?
Yes
Exit: can’t Issue command
Note: Issuing a command at this point will cause previous transmit frame to be lost.
No
Host Writes Transmit Command to the TxCMD Register
Host Writes Transmit Frame Length to the TxLength Register
Host Reads the BusST Register (Register 18)
Transmit Request
Host Enters Interrupt Routine
Rdy4TxNOW bit = 1?
Yes
No
Exit WAIT-for-interrupt
Host Reads ISQ
Rdy4Tx bit = 1?
Yes
No
CS8900A Commits Buffer Space to Transmit Frame
Host Writes Transmit Frameto CS8900A
Process other events that caused interrupt
CS8900A Transmits Frame
Exit Transmit Process

Figure 31. Transmit Operation in Interrupt Mode

3) The host reads the BusST register. This read is performed in memory mode by reading Register 18, at memory base + 0138h. In I/O mode,the host must first set the PacketPage Pointer at the correct location by writing 0138h to the PacketPage Pointer Port (I/O base + 000Ah), it than can read the BusST register from the PacketPage Data Port (I/O base + 000Ch).After reading the register, the Rdy4TxNOW bit is checked. If the bit is set, the frame can be written to CS8900A memory. If Rdy4TxNOW is clear, the host will have to wait for the CS8900A buffer memory to become available at which time the host will be interrupted. On interrupt, the host enters the interrupt service routine and reads ISQ register (Memory base + 0120h in memory mode and I/O base + 0008h in I/O) and checks the Rdy4Tx bit (bit 8). If Rdy4Tx is clear then the CS8900A waits for the next interrupt. If Rdy4Tx is set, then the CS8900A is ready to accept the frame.

4) When the CS8900A is ready to accept the frame, the host transfers the entire frame from host memory to CS8900A memory using REP instruction (REP MOVS to memory base + 0A00h in memory mode, and REP OUT to Receive/Transmit Data Port (I/O base + 0000h) in I/O mode).

### 5.7.8 Completing Transmission

When the CS8900A successfully completes transmitting a frame, it sets the TxOK bit (Register 8,TxEvent, Bit 8). If the TxOKiE bit (Register 7, TxCFG, bit 8) is set, the CS8900A generates a corresponding interrupt.

### 5.7.9 Rdy4TxNOW vs. Rdy4Tx

The Rdy4TxNOW bit (Register 18, BusST, bit 8) is used to tell the host that the CS8900A is ready to accept a frame for transmission. This bit is used during the Transmit Request process or after the Transmit Request process to signal the host that space has become available when interrupts are not being used (i.e. the Rdy4TxiE bit (Register B,BufCFG, Bit 8) is not set). Also, the Rdy4Tx bit is used with interrupts and requires the Rdy4TxiE bit be set.

Figure 30 provides a diagram of error free transmission without collision.

### 5.7.10 Committing Buffer Space to a Transmit Frame

When the host issues a transmit request, the CS8900A checks the length of the transmit frame to see if there is sufficient on-chip buffer space. If there is, the CS8900A sets the Rdy4TxNOW bit. If not, and the Rdy4TxiE bit is set, the CS8900A waits for buffer space to free up and then sets the Rdy4Tx bit. If Rdy4TxiE is not set, the CS8900A sets the Rdy4TxNOW bit when space becomes available.

Even though transmit buffer space may be available, the CS8900A does not commit buffer space to a transmit frame until all of the following are true:

1) The host must issues a Transmit Request;

2) The Transmit Request must be successful; and,

3) Either the host reads that the Rdy4TxNOW bit(Register 18, BusST, Bit 8) is set, or the host reads that the Rdy4Tx bit (Register C, BufEvent, bit 8) is set.

If the CS8900A commits buffer space to a particular transmit frame, it will not allow subsequent frames to be written to that buffer space as long as the transmit frame is committed.

After buffer space is committed, the frame is subsequently transmitted unless any of the following

occur:

1) The host completely writes the frame data, but transmission failed on the Ethernet line. There are three such failures, and these are indicated by three transmit error bits in the TxEvent register (Register 8): 16coll, Jabber, or Out-of-

Window.

Or:

2) The host aborts the transmission by setting the Force (Register 9, TxCMD, bit 8) bit. In this case, the committed transmit frame, as well as any yet-to-be-transmitted frames queued in the on-chip memory, are cleared and not transmitted. The host should make TxLength = 0 when using the Force bit.

Or:

3) There is a transmit under-run, and the TxUnderrun bit (Register C, BufEvent, Bit 9) is set.

Successful transmission is indicated when the TxOK bit (Register 8, TxEvent, Bit 8) is set.

## 5.7.11 Transmit Frame Length

The length of the frame transmitted is determined by the value written into the TxLength register(PacketPage base + 0146h) during the Transmit Request. The length of the transmit frame may be modified by the configuration of the TxPadDis bit (Register 9, TxCMD, Bit D) and the InhibitCRC bit (Register 9, TxCMD, Bit C). Table 35 defines how these bits affect the length of the transmit frame. In addition, it shows which frames the CS8900A will send.

| Register 9, TxCMD | | Host specified transmit length at 0146h (in bytes) | | | |
|---|---|---|---|---|---|
| TxPadDis (Bit D) | InhibitCRC (Bit C) | 3<TxLength <60 | 60<TxLength<1514 | 1514<TxLength< 1518 | TxLength >1518 |
| 0 | 0 | Pad to 60 and add CRC | Send frame and add CRC [Normal Mode] | Will not send | Will not send |
| 0 | 1 | Pad to 60 and send without CRC | Send frame without CRC | Send frame without CRC | Will not send |
| 1 | 0 | Send without pads, and add CRC | Send frame and add CRC | Will not send | Will not send |
| 1 | 1 | Send without pads and without CRC | Send frame without CRC | Send frame without CRC | Will not send |

Notes: 8. If the TxPadDis bit is clear and InhibitCRC is set and the CS8900A is commanded to send a frame of length less than 60 bytes, the CS8900A pads.

9. The CS8900A will not send a frame with TxLength less than 3 bytes.

Table 35. Transmit Frame Length