ICS 25.040
N 10

中华人民共和国国家标准

GB/T 19582.1—2008
代替 GB/Z 19582.1—2004

基于 Modbus 协议的工业自动化网络规范
第 1 部分:Modbus 应用协议

Modbus industrial automation network specification—
Part 1:Modbus application protocol

2008-02-27 发布　　2008-09-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前　言

GB/T 19582—2008《基于 Modbus 协议的工业自动化网络规范》分为三部分：

——第 1 部分：Modbus 应用协议；

——第 2 部分：Modbus 协议在串行链路上的实现指南；

——第 3 部分：Modbus 协议在 TCP/IP 上的实现指南。

第 1 部分描述了 Modbus 事务处理；第 2 部分提供了有助于开发者在串行链路上实现 Modbus 应用层的参考信息；第 3 部分提供了有助于开发者在 TCP/IP 上实现 Modbus 应用层的参考信息。

GB/T 19582—2008 包括两个通信规程中使用的 Modbus 应用层协议和服务规范：

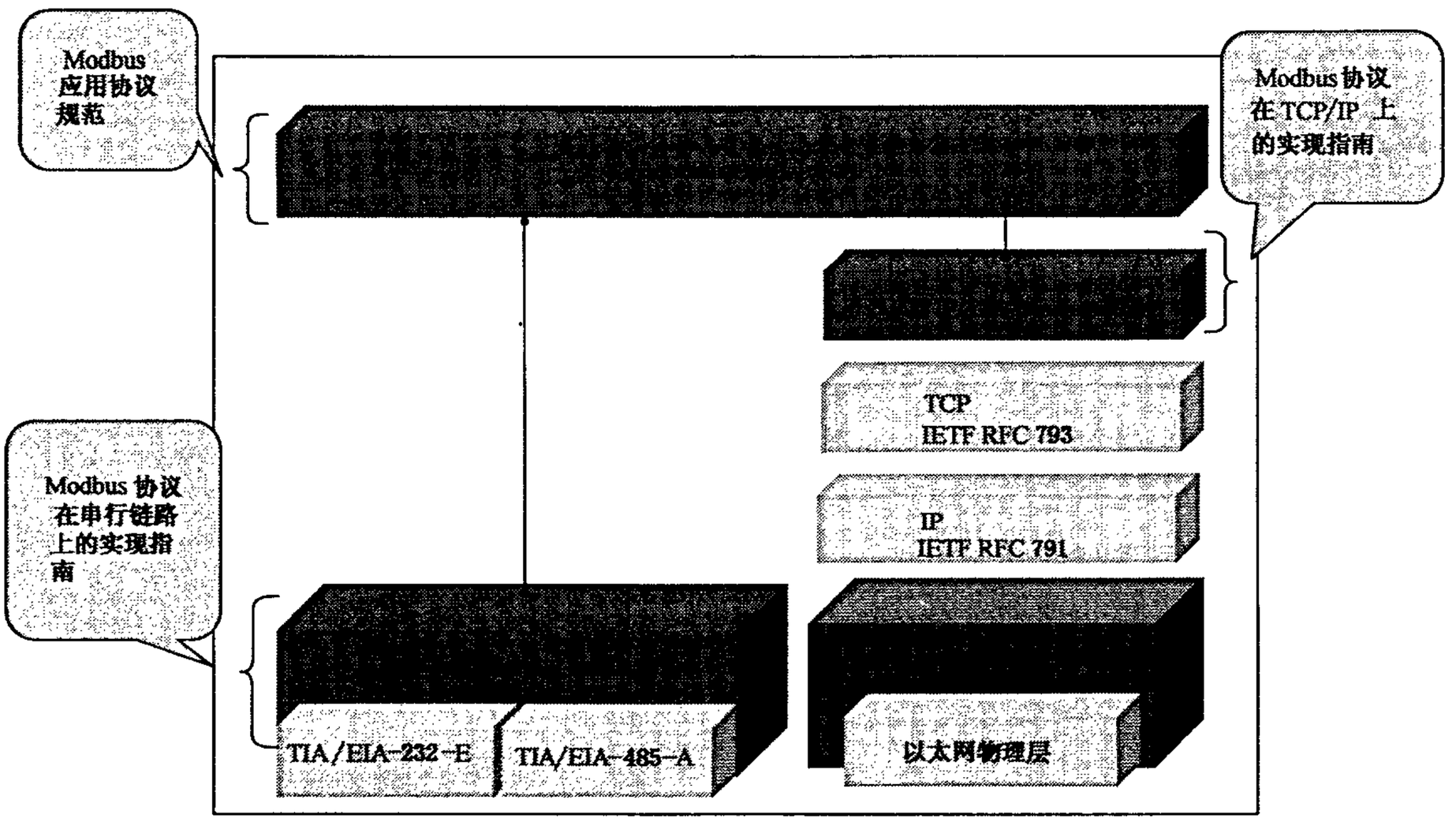
——串行链路上的 Modbus

Modbus 串行链路基于 TIA/EIA 标准：232-E 和 485-A。

——TCP/IP 上的 Modbus

Modbus TCP/IP 基于 IETF 标准：RFC793 和 RFC791。

串行链路和 TCP/IP 上的 Modbus 是根据相应 ISO 分层模型说明的两个通信规程。下图强调指出了 GB/T 19582—2008 的主要部分。深色方框表示规范，浅色方框表示已有的国际标准(TIA/EIA 和 IETF 标准)。



本部分从实施之日起代替 GB/Z 19582.1—2004；GB/Z 19582.1—2004 并于该日起予以废止。

本部分的附录 A、附录 B 为资料性附录。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量和控制标准化技术委员会第四分技术委员会归口。

本部分起草单位：机械工业仪器仪表综合技术经济研究所、西南大学、上海自动化仪表股份有限公司、北京交通大学现代通信研究所、北京机械工业自动化研究所、国家继电器质量监督检验中心、中国四联仪器仪表集团有限公司、中海石油研究中心、西北工业大学、施耐德电气(中国)投资有限公司。

本部分主要起草人：王玉敏、柳晓菁、刘枫、包伟华、孙昕、刘云男、唐济扬、贺春、刘渝新、徐伟华、欧阳劲松、何军红、华镕、王勇。

GB/Z 19582.1 首次发布时间为 2004 年 9 月 21 日，本部分第一次修订。

引　言

GB/T 19582—2008 是对 GB/Z 19582—2004《基于 Modbus 协议的工业自动化网络规范》的修订，修订的依据是 IEC 61158 CPF15(FDIS):2006 实时以太网 Modbus-RTPS。本部分的结构与GB/Z 19582.1—2004 基本一致，但在技术内容上对 GB/Z 19582.1—2004 进行了补充和完善。

基于 Modbus 协议的工业自动化网络规范
第 1 部分:Modbus 应用协议

1 范围

Modbus 是 OSI 模型第 7 层上的应用层报文传输协议,它在连接至不同类型总线或网络的设备之间提供客户机/服务器通信,见图 1。

从 1979 年开始,Modbus 作为工业串行链路的事实标准,Modbus 使成千上万的自动化设备能够通信。目前,对简单而精致的 Modbus 结构的支持仍在增长。互联网用户能够使用 TCP/IP 栈上的保留系统端口 502 访问 Modbus。

Modbus 是一个请求/应答协议,并且提供功能码规定的服务。Modbus 功能码是 Modbus 请求/应答 PDU 的元素。本部分描述了 Modbus 事务处理框架内使用的功能码。

Modbus 应用层报文传输协议用于在通过不同类型的总线或网络连接的设备之间的客户机/服务器通信。

目前,通过下列方式实现 Modbus 通信:

——以太网上的 TCP/IP,见 GB/T 19582.3。

——各种介质(有线:EIA/TIA-232-E、EIA-422、EIA/TIA-485-A;光纤、无线等等)上的异步串行传输。

——Modbus+,一种高速令牌传递网络。

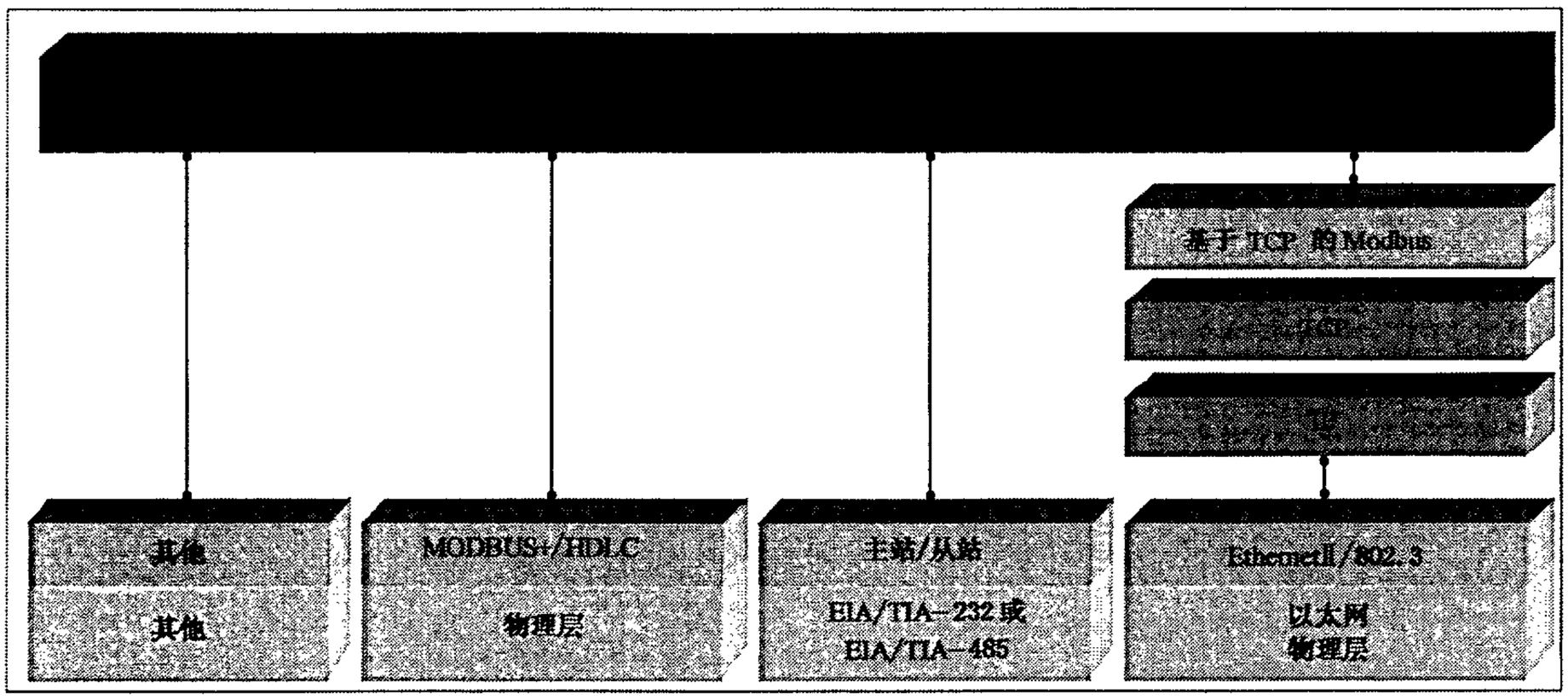


图 1 Modbus 通信栈

2 规范性引用文件

下列文件中的条款通过 GB/T 19582 的本部分的引用而成为本部分的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分,然而,鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本部分。

GB/T 15969 可编程序控制器

RFC 791 Internet Protocol,Sep81 DARPA

3 缩略语

ADU(Application Data Unit)	应用数据单元
HDLC(High Level Data Link Control)	高级数据链路控制
HMI(Human Machine Interface)	人机界面
IETF(Internet Engineering Task Force)	互联网工程工作组
I/O(Input/Output)	输入/输出
IP(Internet Protocol)	因特网协议
LSB(Least Significant Bit)	最低有效位
MAC(Medium Access Control)	介质访问控制
MB(Modbus Protocol)	Modbus 协议
MBAP(Modbus Application Protocol)	Modbus 应用协议
MEI(Modbus Encapsulated Interface)	Modbus 封装接口
MSB(Most Significant Bit)	最高有效位
NAK(Negative Acknowlegment)	否定确认
PDU(Protocol Data Unit)	协议数据单元
PLC(Programmable Logic Controller)	可编程序逻辑控制器
RFC(Request For Comment)	互联网“请求注解”文档
TCP(Transport Control Protocol)	传输控制协议

4 背景概要

Modbus 协议可以方便地在各种网络体系结构内进行通信，见图 2。

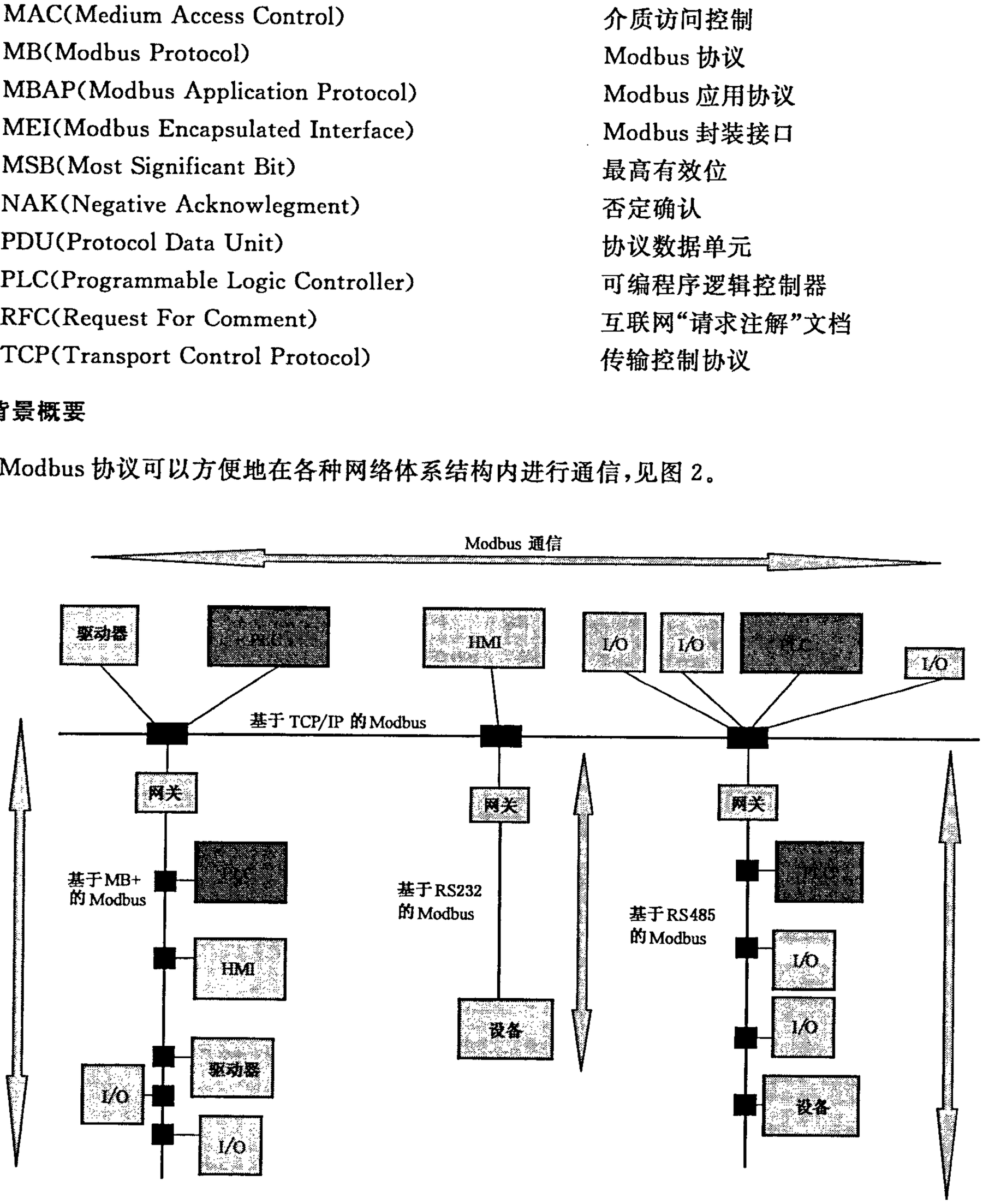


图 2 Modbus 网络体系结构的示例

每种设备(PLC、HMI、控制面板、驱动器、运动控制、I/O 设备……)都能使用 Modbus 协议来启动远程操作。

在基于串行链路和 TCP/IP 以太网上能够进行同样的通信。网关能够实现在各种使用 Modbus 协议的总线或网络之间的通信。

5 总体描述

5.1 协议描述

Modbus 协议定义了一个与基础通信层无关的简单协议数据单元(PDU)。特定总线或网络上的 Modbus 协议映射能够在应用数据单元(ADU)上引入一些附加字段,见图 3。

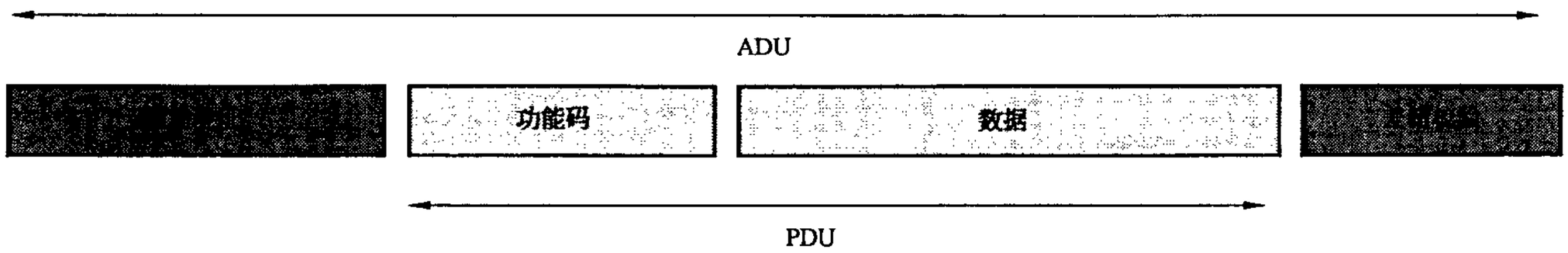


图 3 通用 Modbus 帧

Modbus 应用数据单元由启动 Modbus 事务处理的客户机创建。功能码向服务器指示将执行哪种操作。Modbus 应用协议建立了客户机启动的请求格式。

用一个字节编码 Modbus 数据单元的功能码字段。有效的码范围是十进制 1～255(128～255 保留用于异常响应)。当从客户机向服务器设备发送报文时,功能码字段通知服务器执行哪种操作。功能码“0”无效。

向一些功能码加入子功能码来定义多项操作。

从客户机向服务器设备发送的报文数据字段包括附加信息,服务器使用这个信息执行功能码定义的操作。这个字段还包括离散量和寄存器地址、处理的项目数量以及字段中的实际数据字节数。

在某种请求中,数据字段可以是不存在的(0 长度),在此情况下服务器不需要任何附加信息。功能码仅说明操作。

如果在一个正确接收的 Modbus ADU 中,不出现与所请求的 Modbus 功能有关的差错,那么服务器至客户机的响应数据字段包括所请求的数据。如果出现与所请求的 Modbus 功能有关的差错,那么该字段包括一个异常码,服务器应用能够使用这个字段确定下一个执行的操作。

例如,客户机能够读一组离散量输出或输入的开/关状态,或者客户机能够读/写一组寄存器的数据内容。

当服务器对客户机响应时,它使用功能码字段来指示正常(无差错)响应(见图 4)或者出现某种差错(称为异常响应),见图 5。对于正常响应,服务器仅复制原始功能码。

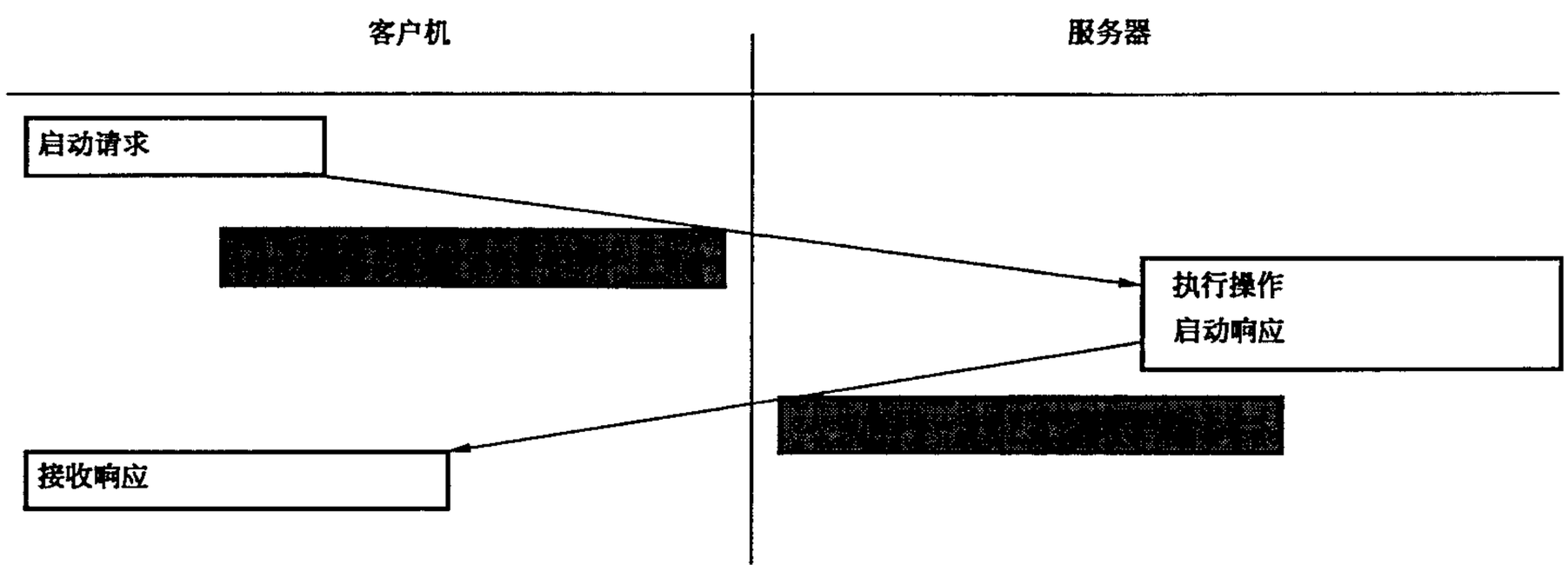


图 4 Modbus 事务处理(无差错)

对于异常响应，服务器将请求 PDU 中的原始功能码的最高有效位设置逻辑 1 后返回。

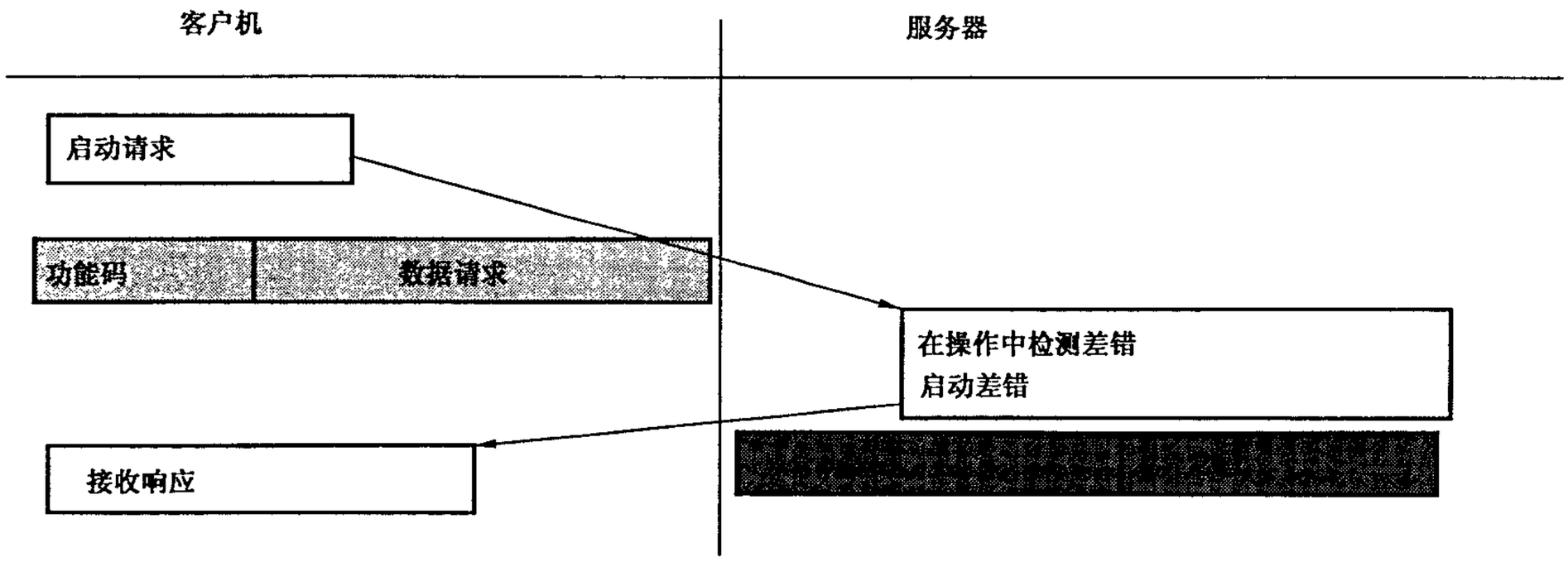


图 5　Modbus 事务处理(异常响应)

注：需要超时管理，以避免无限期地等待可能不会出现的应答。

Modbus 最初在串行链路上的实现(最大 RS485 ADU＝256 字节)限制了 Modbus PDU 的长度。

因此，对串行链路通信来说，Modbus PDU＝256－服务器地址(1 字节)－CRC(2 字节)＝253 字节。

从而：

RS232/RS485 ADU＝253 字节＋服务器地址(1 字节)＋CRC(2 字节)＝256 字节。

TCP Modbus ADU＝253 字节＋MBAP(7 字节)＝260 字节。

Modbus 协议定义了三种 PDU。它们是：

——Modbus 请求 PDU，mb_req_pdu；

——Modbus 响应 PDU，mb_rsp_pdu；

——Modbus 异常响应 PDU，mb_excep_rsp_pdu。

定义 mb_req_pdu 为：

mb_req_pdu＝{function_code，request_data}，其中：

function_code＝[1 字节] Modbus 功能码。

request_data＝[n 字节]，这个字段与功能码有关，通常包括诸如变量引用、变量计数、数据偏移量、子功能码等信息。

定义 mb_rsp_pdu 为：

mb_rsp_pdu＝{function_code，response_ data}，其中：

function_code＝[1 字节]Modbus 功能码。

response_data＝[n 字节]，这个字段与功能码有关，通常包括诸如变量引用、变量计数、数据偏移量、子功能码等信息。

定义 mb_excep_rsp_pdu 为：

mb_excep_rsp_pdu＝{exception-function_code，exception_code}，其中：

exception-function_code＝[1 字节] Modbus 功能码＋0x80。

exception_code＝[1 字节] Modbus 异常码，在表“Modbus 异常码”中定义(见第 8 章)。

5.2　数据编码

Modbus 使用最高有效字节在低地址存储的方式表示地址和数据项。这意味着当发送多个字节时，首先发送最高有效字节。例如：

寄存器大小　　值

16 位　　0x1234　　发送的第一字节为　0x12　　然后 0x34

注：更详细的信息参见参考文献[1]。

5.3 Modbus 数据模型

Modbus 的数据模型是以一组具有不同特征的表为基础建立的。4 个基本表见表 1。

表 1 Modbus 数据模型基本表

基本表	对象类型	访问类型	注 释
离散量输入	单个位	只读	I/O 系统可提供这种类型的数据
线圈	单个位	读写	通过应用程序可改变这种类型的数据
输入寄存器	16 位字	只读	I/O 系统可提供这种类型的数据
保持寄存器	16 位字	读写	通过应用程序可改变这种类型的数据

输入与输出之间以及位寻址的和字寻址的数据项之间的区别并不意味着应用特性的差别。如果所有 4 个表相互覆盖是对该目标机器最自然的解释，也是完全可接受的，而且很普遍。

对于每个基本表，协议都允许单个地选择 65 536 个数据项，而且其读写操作被设计为可以越过多个连续数据项直到数据大小规格限制，该限制与事务处理功能码有关。

很显然，必须将 Modbus 处理的所有数据(位，寄存器)放置在设备应用存储器中。但是，存储器的物理地址不应该与数据引用混淆。仅要求将数据引用与物理地址链接。

Modbus 功能码中使用的 Modbus 逻辑编号是以 0 开始的无符号整数。

——Modbus 模型实现的示例

下列示例表示了在设备中组织数据的两种方法。组织数据的方法有多种，在本部分中没有对所有方法进行描述。每个设备根据其应用都有它自己的组织数据的方法。

示例 1：含有 4 个独立块的设备

图 6 表示了含有数字量和模拟量、输入量和输出量的设备中的数据组织。由于不同块中的数据不相关，因此每个块是相互独立的。这样可通过不同 Modbus 功能码访问每个块。

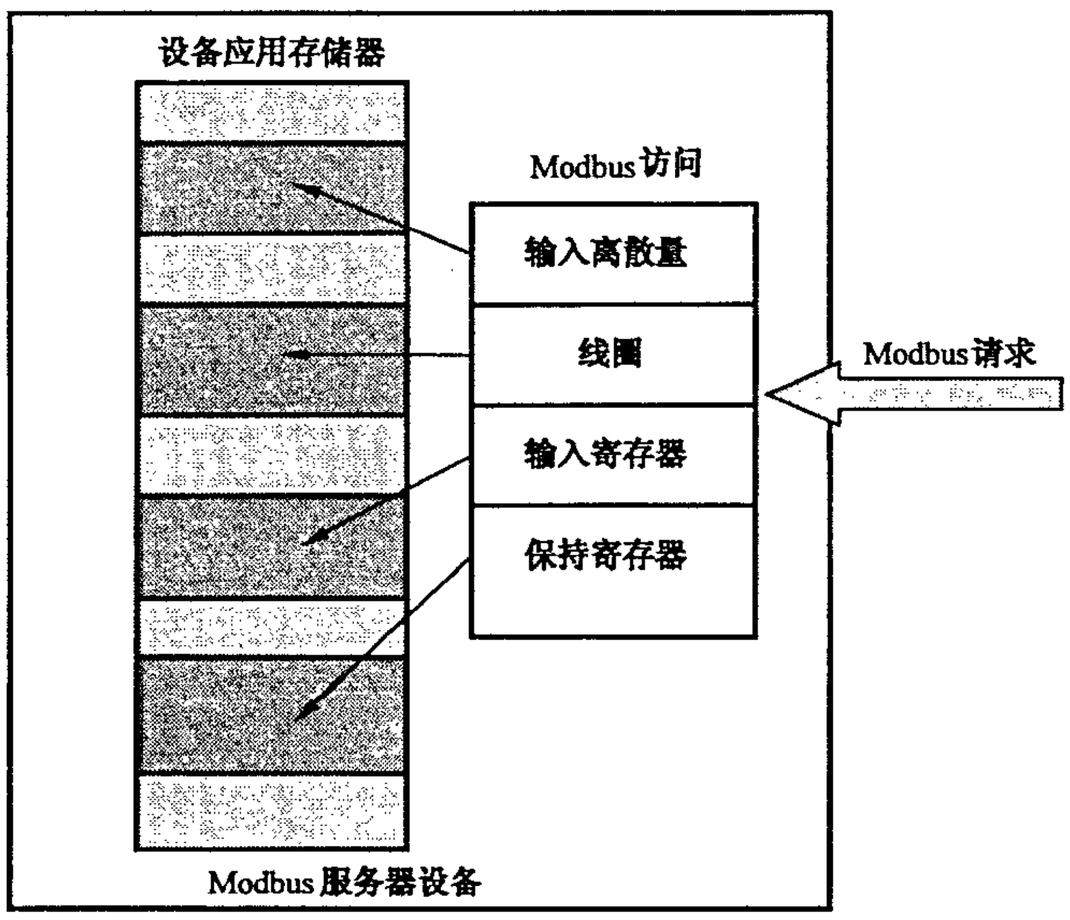


图 6 含有独立块的 Modbus 数据模型

示例 2：仅含有 1 个块的设备

在图 7 中，设备仅含有 1 个数据块。通过几个 Modbus 功能码能够得到相同数据，既可通过 16 位访问也可通过 1 位访问。

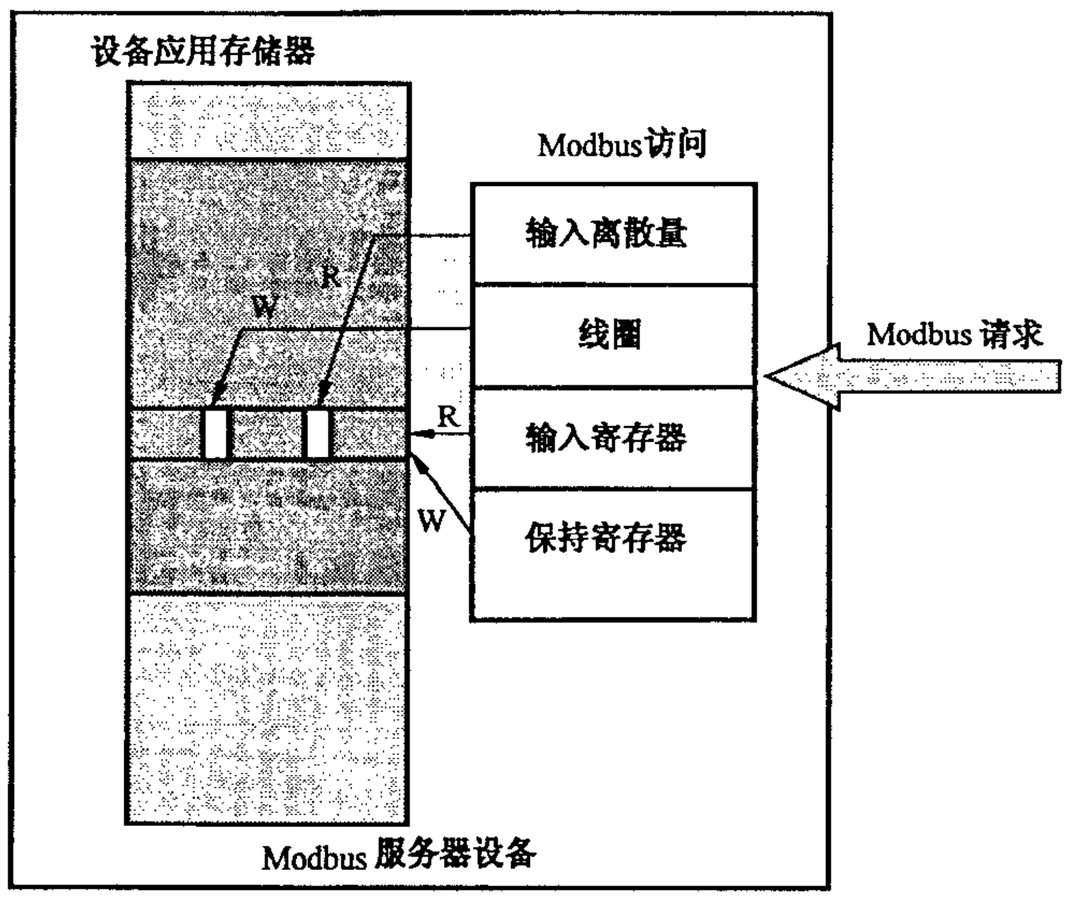


图 7　仅含有 1 个块的 Modbus 数据模型

5.4　Modbus 寻址模型

Modbus 应用协议精确地定义了 PDU 寻址规则。

在 Modbus PDU 中，从 0～65535 寻址每个数据。

Modbus 应用协议还明确地定义了由 4 个块构成的 Modbus 数据模型，每个块由几个编号为 1～n 的元素构成。

在 Modbus 数据模型中，从 1～n 来编号数据块中的每个元素。

然后，必须将 Modbus 数据模型与设备应用结合(GB/T 15969 对象，其他应用模型)。

Modbus 数据模型和设备应用之间的映射完全与特定设备相关。

图 8 表示了用 Modbus PDU 的 X－1 来寻址编号为 X 的 Modbus 数据。

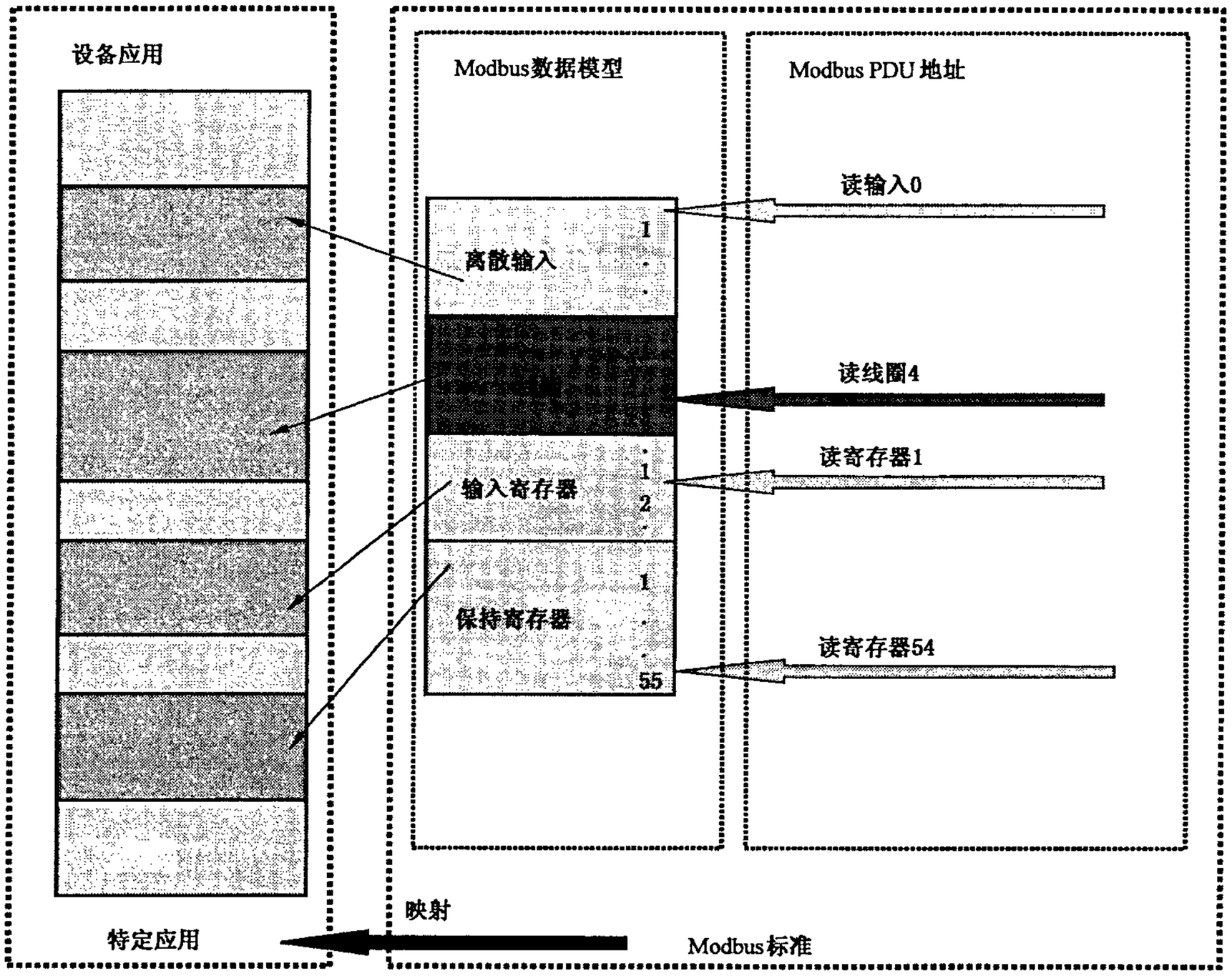


图 8　Modbus 寻址模型

5.5 Modbus 事务处理的定义

图 9 是 Modbus 事务处理状态图，描述了在服务器侧 Modbus 事务处理的一般处理过程。

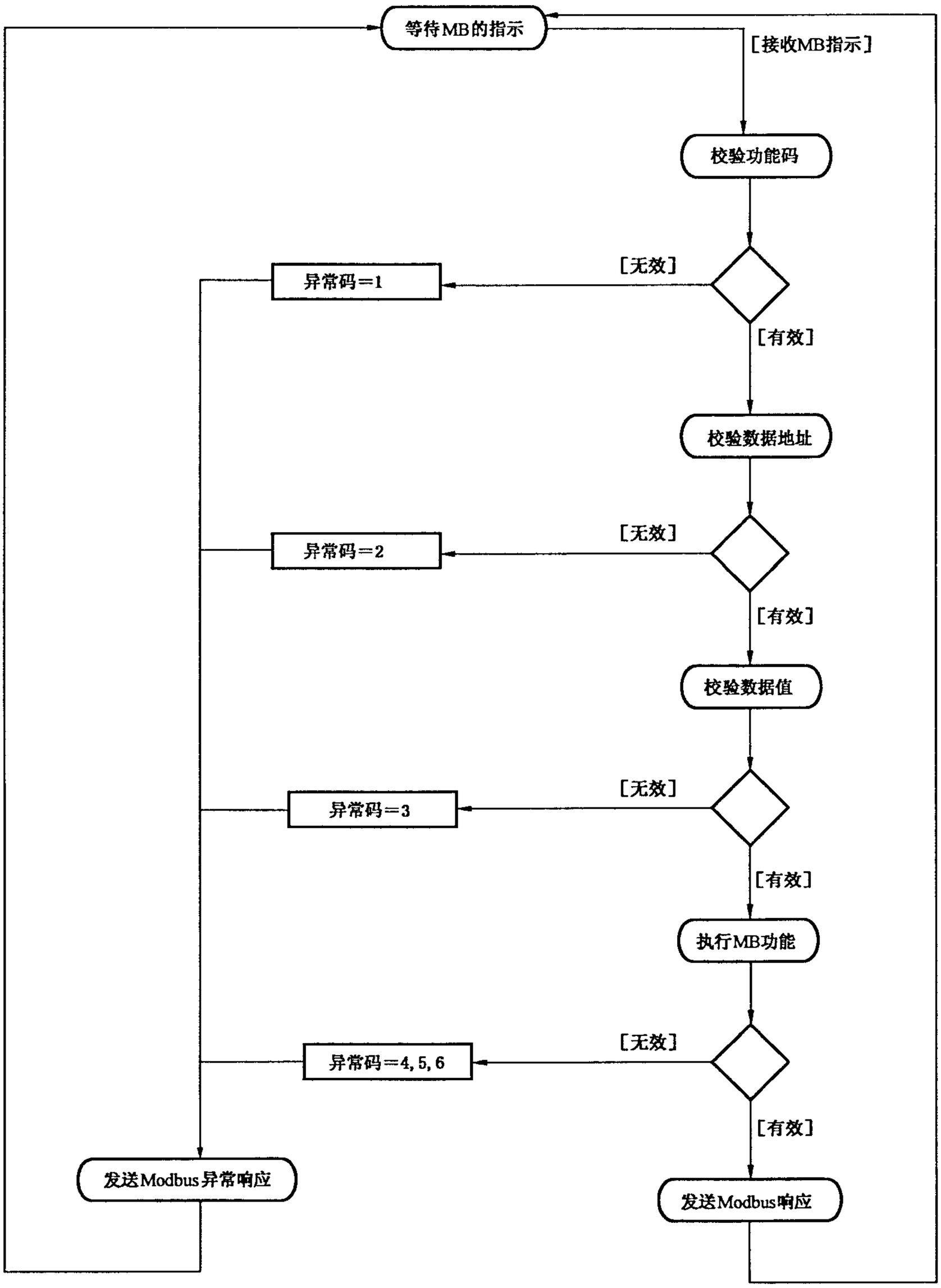


图 9 Modbus 事务处理的状态图

一旦服务器处理请求，就使用相应的 Modbus 服务器事务处理生成 Modbus 响应。

根据处理结果，可以建立两种类型响应：

——一个正常的 Modbus 响应：响应功能码＝请求功能码。

——一个异常的 Modbus 响应(见第 8 章)：

1) 用来为客户机提供处理过程中与所发现的差错相关的信息；

2） 异常功能码＝请求功能码＋0x80；

3） 提供一个异常码来指示差错原因。

6 功能码分类

有三类 Modbus 功能码，见图 10。它们是：

a） 公共功能码

1） 被确切定义的功能码；

2） 保证是唯一的；

3） 由 Modbus-IDA. org 确认的；

4） 公开的文档；

5） 可进行一致性测试；

6） 包含已被定义的公共功能码和保留给未来使用的功能码。

b） 用户定义功能码

1） 有两个用户定义功能码的区域，即十进制的 65～72 和 100～110；

2） 用户无需 Modbus 组织的任何批准就可以选择和实现一个功能码；

3） 不能保证被选功能码的使用是唯一的；

4） 如果用户希望将某种功能设置为一个公共功能码，那么用户必须启动 RFC，以便将这种变更引入公共分类中，并且指配一个新的公共功能码。

c） 保留功能码

某些公司在传统产品上现行使用的功能码，不作为公共使用。

注：参见附录 A。

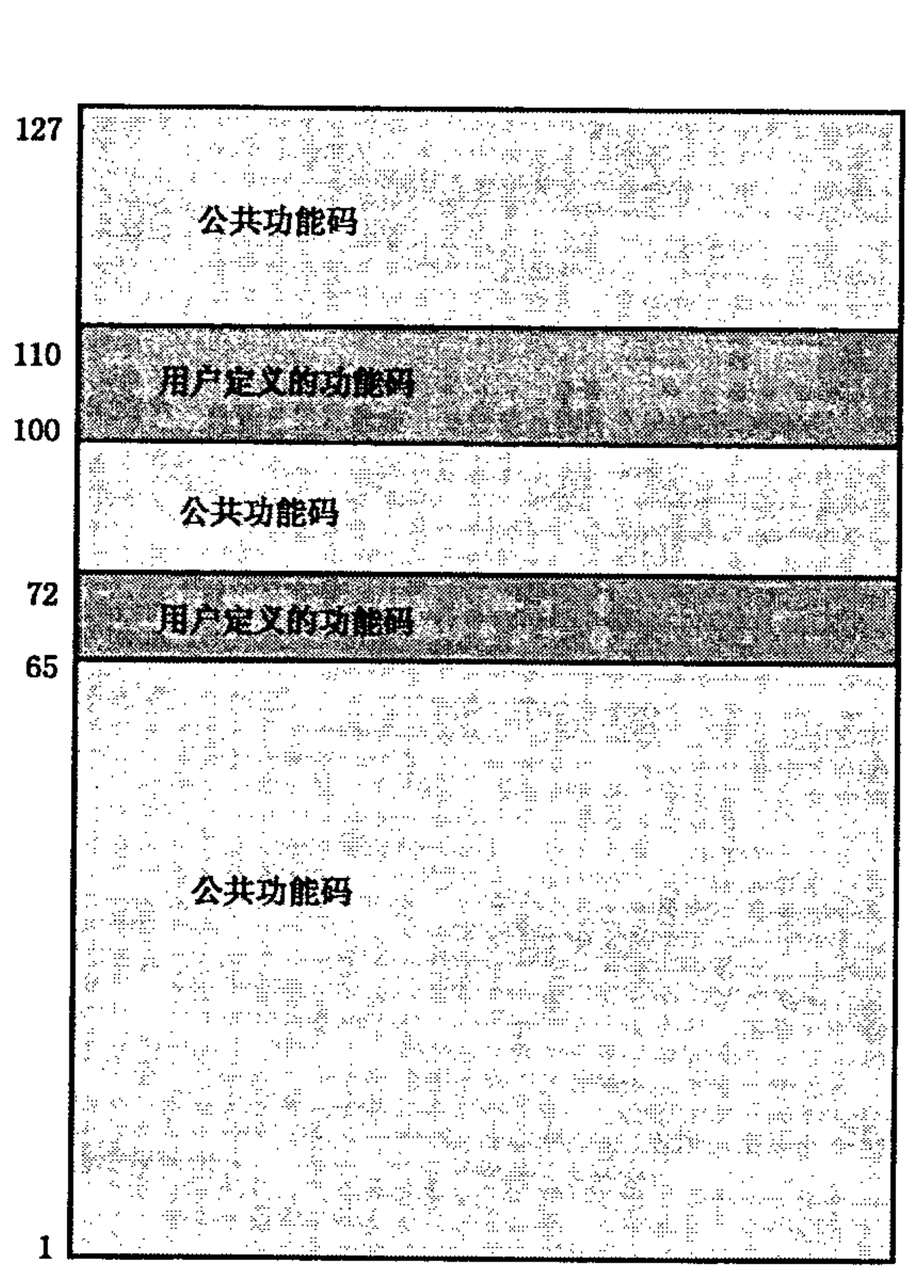


图 10 Modbus 功能码分类

6.1 公共功能码定义

见表 2。

表 2　公共功能码

				功能码			
				码	子码	十六进制	章节
数据访问	比特访问	物理离散量输入	读离散量输入	02		02	7.2
		内部比特或物理线圈	读线圈	01		01	7.1
			写单个线圈	05		05	7.5
			写多个线圈	15		0F	7.11
	16 比特访问	物理输入寄存器	读输入寄存器	04		04	7.4
		内部寄存器或物理输出器寄存器	读保持寄存器	03		03	7.3
			写单个寄存器	06		06	7.6
			写多个寄存器	16		10	7.12
			读/写多个寄存器	23		17	7.17
			屏蔽写寄存器	22		16	7.16
			读 FIFO 队列	24		18	7.18
	文件记录访问		读文件记录	20	6	14	7.14
			写文件记录	21	6	15	7.15
诊断			读异常状态	07		07	7.7
			诊断	08	00-18,20	08	7.8
			获得事件计数器	11		0B	7.9
			获得事件记录	12		0C	7.10
			报告从站 ID	17		11	7.13
			读设备标识码	43	14	2B	7.21
其他			封装接口传输	43	13,14	2B	7.19
			CANopen 通用引用	43	13	2B	7.20

7　功能码描述

7.1　01(0x01)读线圈

见图 11 和表 3～表 5。

使用该功能码从一个远程设备中读 1～2 000 个连续的线圈状态。请求 PDU 指定了起始地址，即指定了第一个线圈地址和线圈数目。在 PDU 中，从零开始寻址线圈。因此编号 1～16 的线圈寻址为 0～15。

响应报文中的线圈按数据字段的每位一个线圈进行打包。状态被表示成 1＝ON 和 0＝OFF。第一个数据字节的 LSB(最低有效位)包含询问中所寻址的输出。其他线圈依次类推，一直到这个字节的高位端为止，并在后续字节中按照从低位到高位的顺序排列。

如果返回的输出数量不是 8 的倍数，将用零填充最后数据字节中的剩余位(一直到字节的高位端)。字节计数字段指定了数据的全部字节数。

表 3　读线圈请求

功能码	1 字节	0x01
起始地址	2 字节	0x0000～0xFFFF
线圈数量	2 字节	1～2 000(0x7D0)

表 4　读线圈响应

功能码	1 字节	0x01
字节计数	1 字节	N[a]
线圈状态	n 字节	n=N 或 N+1
[a] N=输出数量/8,如果余数不等于 0,那么 N=N+1。		

表 5　读线圈错误响应

异常功能码	1 字节	功能码+0x80
异常码	1 字节	01 或 02 或 03 或 04

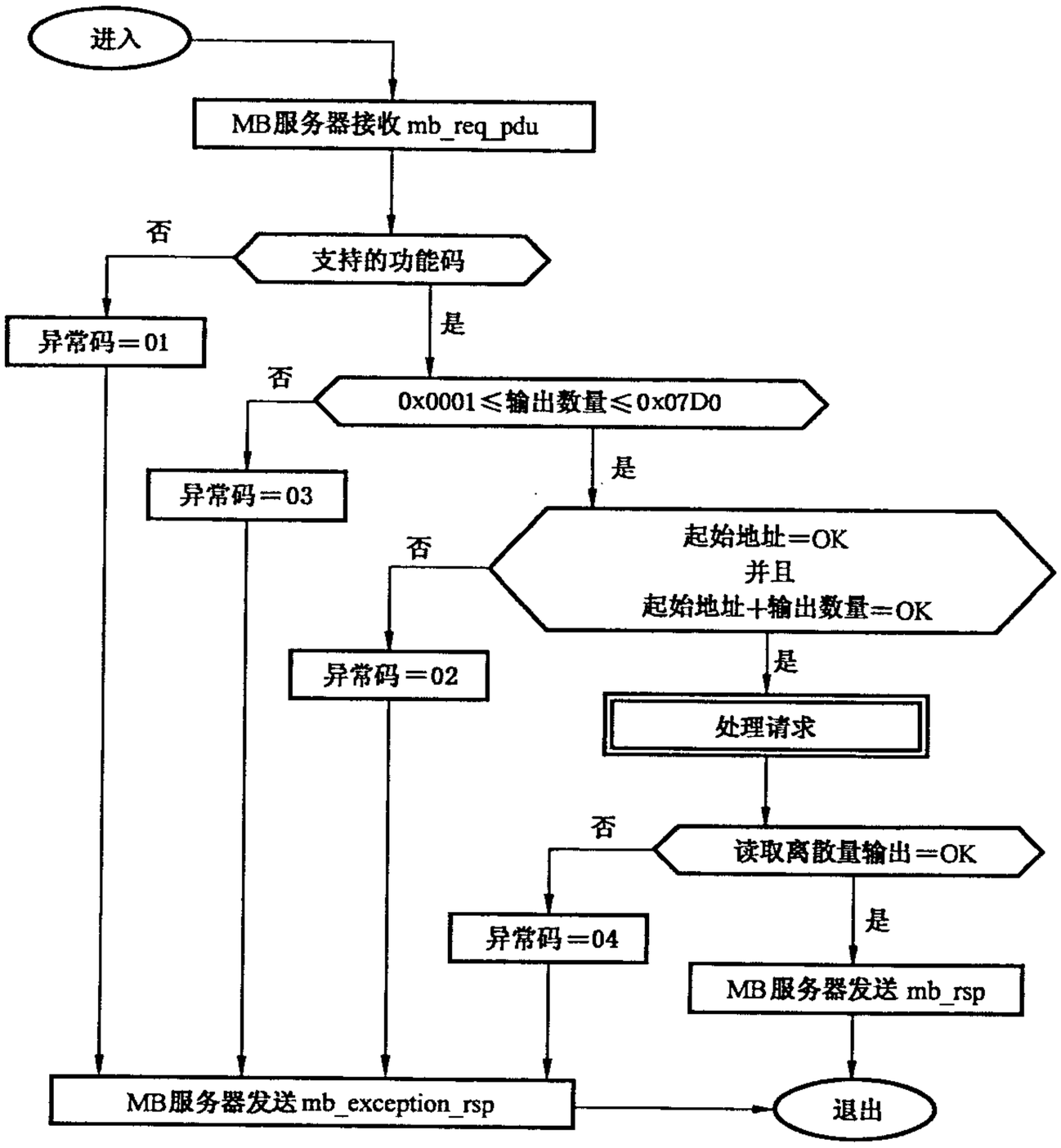


图 11　读线圈状态图

表 6 是一个请求读离散量输出 20～38 的示例。

表 6 读离散量输出

请求		响应	
字段名	十六进制	字段名	十六进制
功能	01	功能	01
起始地址 Hi	00	字节计数	03
起始地址 Lo	13	输出状态 27～20	CD
输出数量 Hi	00	输出状态 35～28	6B
输出数量 Lo	13	输出状态 38～36	05

将输出 27～20 的状态表示为十六进制字节值 CD，或二进制 1100 1101。输出 27 是这个字节的 MSB，输出 20 是 LSB。

通常，一个字节的 MSB 位于左侧，LSB 位于右侧。第一个字节的输出从左至右为 27～20。下一个字节的输出从左到右为 35～28。当串行发送这些位时，从 LSB 向 MSB 传输：20～27、28～35 等等。

在最后的数据字节中，将输出 38～36 的状态表示为十六进制字节值 05，或二进制 0000 0101。输出 38 是左侧第六个位位置，输出 36 是这个字节的 LSB。用零填充 5 个剩余高位位。

注：用零填充 5 个剩余位(一直到高位端)。

7.2 02(0x02)读离散量输入

见图 12 和表 7～表 9。

使用该功能码从一个远程设备中读 1～2 000 个连续的离散量输入状态。请求 PDU 指定了起始地址，即指定了第一个离散量输入地址和离散量输入数目。在 PDU 中，从零开始寻址离散量输入。因此编号 1～16 的离散量输入被寻址为 0～15。

响应报文中的离散量输入按数据字段的每位一个离散量输入进行打包。状态被表示成 1＝ON 和 0＝OFF。第一个数据字节的 LSB(最低有效位)包含询问中所寻址的输入。其他离散量输入依次类推，一直到这个字节的高位端为止，并在后续字节中按照从低位到高位的顺序排列。

如果返回的输入数量不是 8 的倍数，将用零填充最后数据字节中的剩余位(一直到字节的高位端)。字节计数字段指定了数据的全部字节数。

表 7 读离散量输入请求

功能码	1 字节	0x02
起始地址	2 字节	0x0000～0xFFFF
输入数量	2 字节	1～2 000(0x07D0)

表 8 读离散量输入响应

功能码	1 字节	0x02
字节计数	1 字节	N[a]
输入状态	N[a]×1 个字节	…
a N＝输入数量/8，如果余数不等于 0，那么 N＝N＋1。		

表 9 读离散量输入错误响应

异常功能码	1 字节	0x82
异常码	1 字节	01 或 02 或 03 或 04

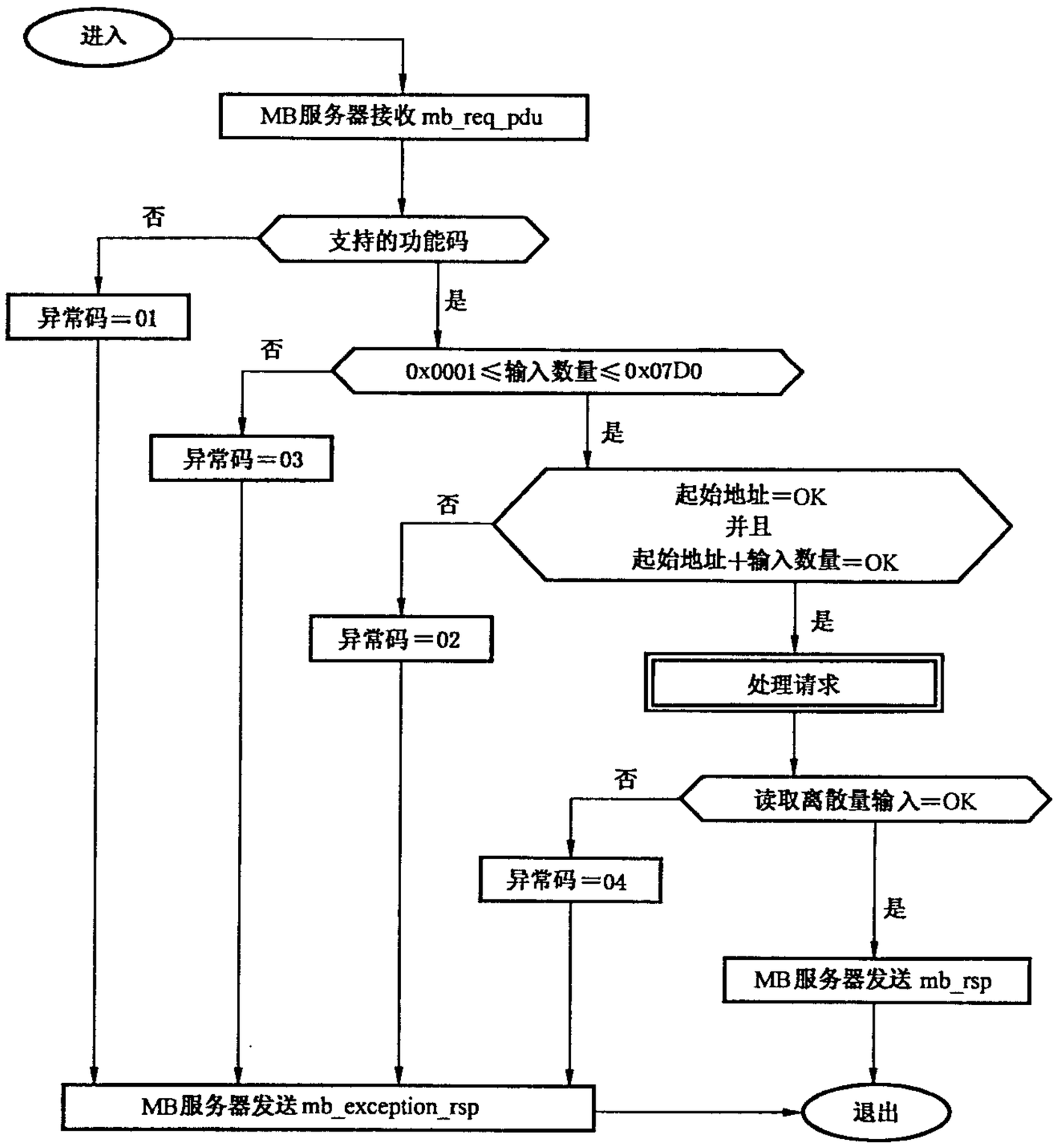


图 12　读离散量输入的状态图

表 10 是一个请求读离散量输入 197～218 的示例。

表 10　读离散量输入

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	02	功能	02
起始地址 Hi	00	字节计数	03
起始地址 Lo	C4	输入状态 204～197	AC
输入数量 Hi	00	输入状态 212～205	DB
输入数量 Lo	16	输入状态 218～213	35

将离散量输入 204～197 的状态表示为十六进制字节值 AC，或二进制 1010 1100。输入 204 是这个字节的 MSB，输入 197 是这个字节的 LSB。

将离散量输入 218～213 的状态表示为十六进制字节值 35，或二进制 0011 0101。输入 218 位于左侧第 3 位，输入 213 是 LSB。

注：用零填充 2 个剩余位(一直到高位端)。

7.3　03(0x03)读保持寄存器

见图 13 和表 11～表 13。

使用该功能码从远程设备中读保持寄存器连续块的内容。请求 PDU 指定了起始寄存器地址和寄

存器数量。在 PDU 中，从零开始寻址寄存器。因此编号 1～16 的寄存器被寻址为 0～15。

将响应报文中的寄存器数据按每个寄存器两字节进行打包，这个二进制内容正好填满每个字节。对于每个寄存器，第一个字节包括高位位，第二个字节包括低位位。

表 11　读保持寄存器请求

功能码	1 字节	0x03
起始地址	2 字节	0x0000～0xFFFF
寄存器数量	2 字节	1～125(0x007D)

表 12　读保持寄存器响应

功能码	1 字节	0x03
字节数	1 字节	2×N[a]
寄存器值	N[a]×2 字节	…
a　N=寄存器的数量。		

表 13　读保持寄存器错误响应

异常功能码	1 字节	0x83
异常码	1 字节	01 或 02 或 03 或 04

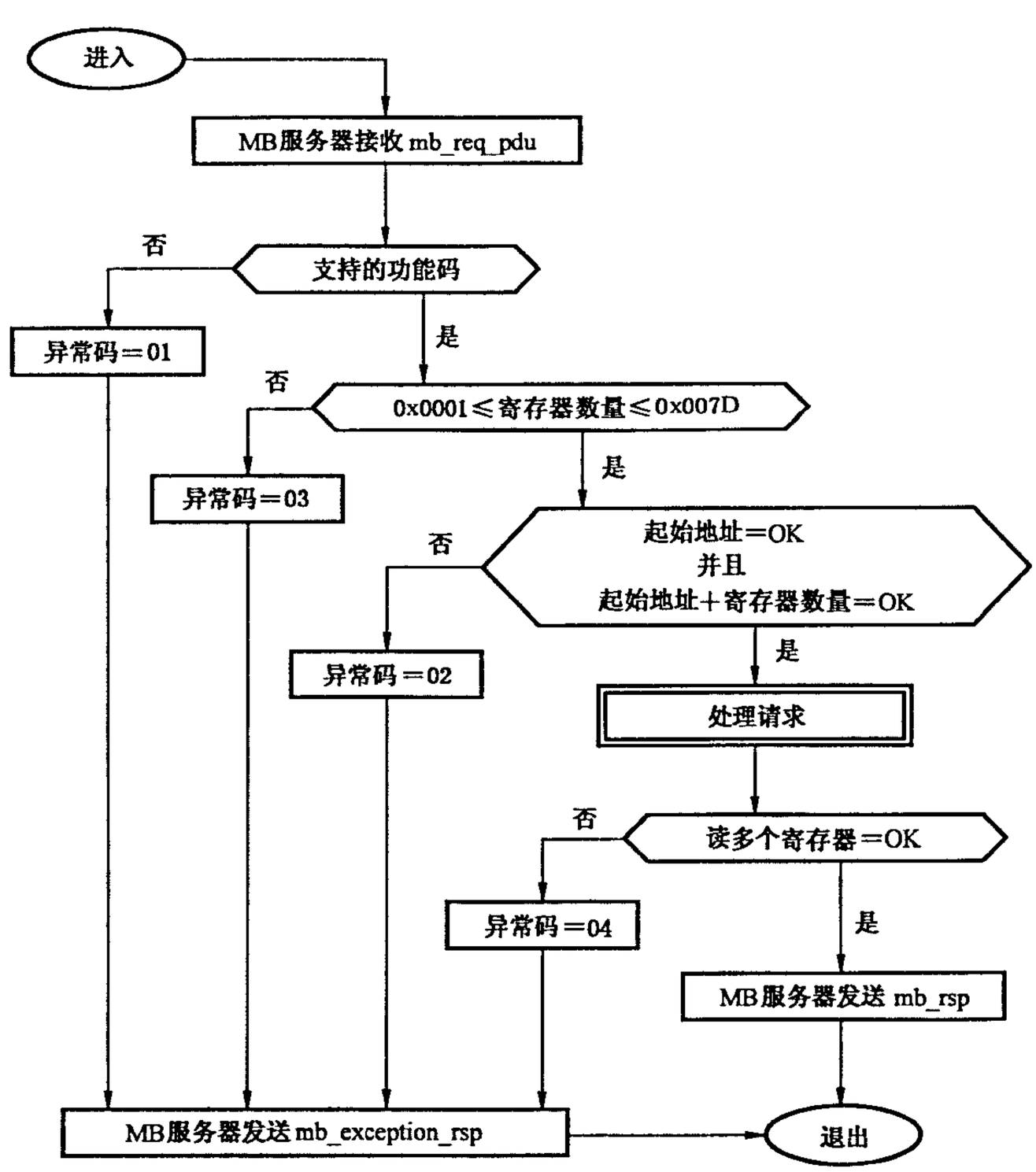


图 13　读保持寄存器的状态图

表 14 是一个请求读保持寄存器 108～110 的示例。

表 14 读保持寄存器

请求		响应	
字段名	十六进制	字段名	十六进制
功能	03	功能	03
起始地址 Hi	00	字节计数	06
起始地址 Lo	6B	寄存器值 Hi(108)	02
寄存器数量 Hi	00	寄存器值 Lo(108)	2B
寄存器数量 Lo	03	寄存器值 Hi(109)	00
		寄存器值 Lo(109)	00
		寄存器值 Hi(110)	00
		寄存器值 Lo(110)	64

将寄存器 108 的内容表示为两个十六进制字节值 02 2B,或十进制 555。将寄存器 109～110 的内容分别表示为十六进制 00 00 和 00 64,或十进制 0 和 100。

7.4 04(0x04)读输入寄存器

见图 14 和表 15～表 17。

使用该功能码从一个远程设备中读 1～125 个连续输入寄存器。请求 PDU 指定了起始地址和寄存器数量。在 PDU 中,从零开始寻址寄存器。因此,编号为 1～16 的输入寄存器被寻址为 0～15。

将响应报文中的寄存器数据按每个寄存器两字节进行打包,这个二进制内容正好填满每个字节。对于每个寄存器,第一个字节包括高位位,第二个字节包括低位位。

表 15 读输入寄存器请求

功能码	1 字节	0x04
起始地址	2 字节	0x0000～0xFFFF
输入寄存器数量	2 字节	0x0001～0x007D

表 16 读输入寄存器响应

功能码	1 字节	0x04
字节计数	1 字节	$2\times N^a$
输入寄存器	$N^a\times 2$ 字节	…
a N=输入寄存器的数量。		

表 17 读输入寄存器错误响应

异常功能码	1 字节	0x84
异常码	1 字节	01 或 02 或 03 或 04

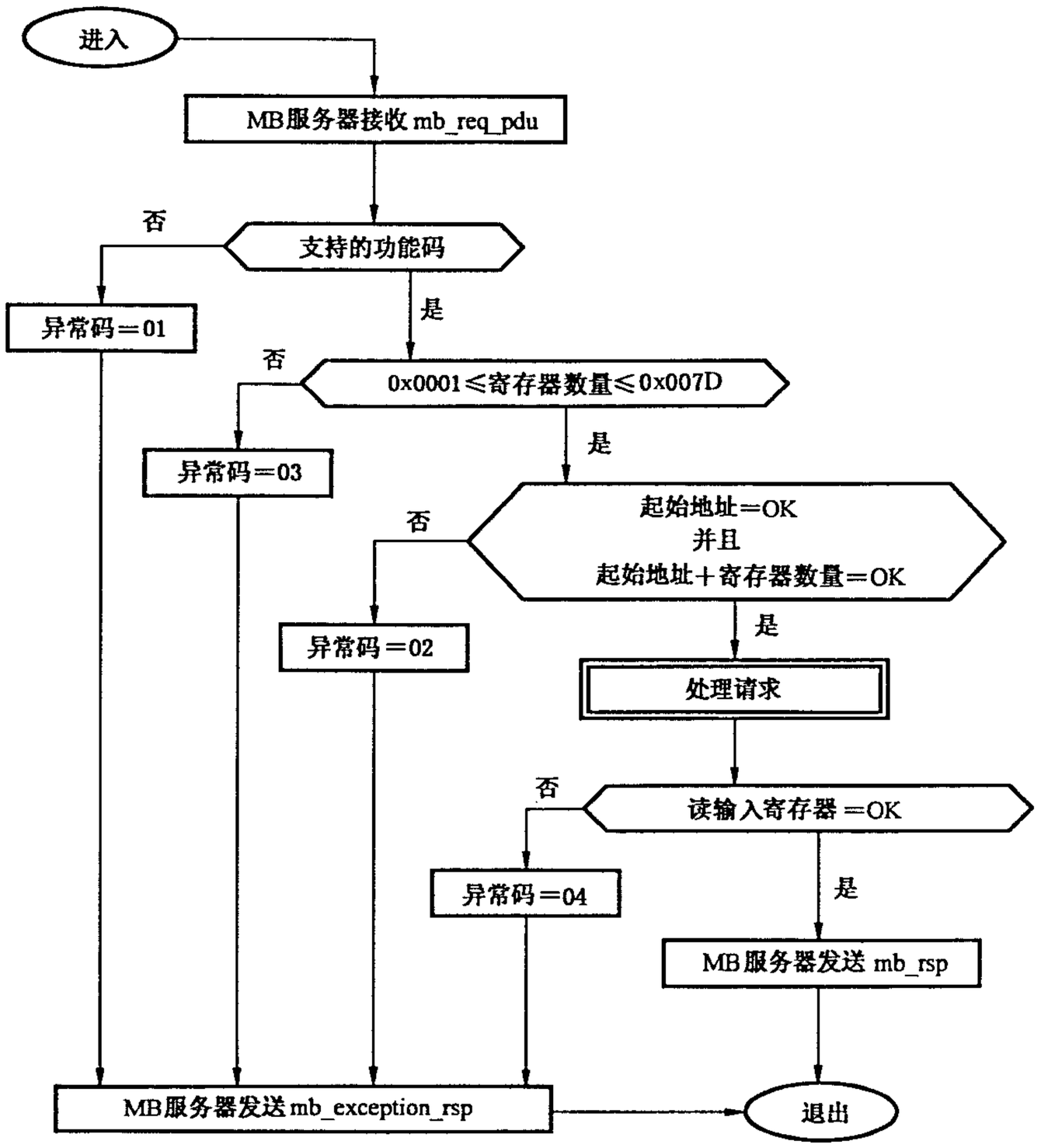


图 14　读输入寄存器的状态图

表 18 是一个请求读输入寄存器 9 的示例。

表 18　读输入寄存器

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	04	功能	04
起始地址 Hi	00	字节计数	02
起始地址 Lo	08	输入寄存器 9 Hi	00
输入寄存器数量 Hi	00	输入寄存器 9 Lo	0A
输入寄存器数量 Lo	01		

将输入寄存器 9 的内容表示为两个十六进制字节值 00 0A，或十进制 10。

7.5　05(0x05)写单个线圈

见图 15 和表 19～表 21。

使用该功能码将一个远程设备中的单个输出写为 ON 或 OFF。

所请求的 ON/OFF 状态由请求数据字段中的常数指定。十六进制值 FF 00 请求输出为 ON。十六进制值 00 00 请求输出为 OFF。其他所有值均是非法的，并且对输出不起作用。

请求 PDU 指定了被强制的线圈地址。从零开始寻址线圈。因此，编号为 1 的线圈被寻址为 0。所请求的 ON/OFF 状态由线圈值字段的常数指定。十六进制值 0xFF00 请求线圈为 ON。十六进制值 0x0000 请求线圈为 OFF。其他所有值均为非法的，并且对线圈不起作用。

正常的响应是请求的复制，在写入线圈状态之后被返回。

表 19　写单个线圈请求

功能码	1 字节	0x05
输出地址	2 字节	0x0000～0xFFFF
输出值	2 字节	0x0000 或 0xFF00

表 20　写单个线圈响应

功能码	1 字节	0x05
输出地址	2 字节	0x0000～0xFFFF
输出值	2 字节	0x0000 或 0xFF00

表 21　写单个线圈错误响应

异常功能码	1 字节	0x85
异常码	1 字节	01 或 02 或 03 或 04

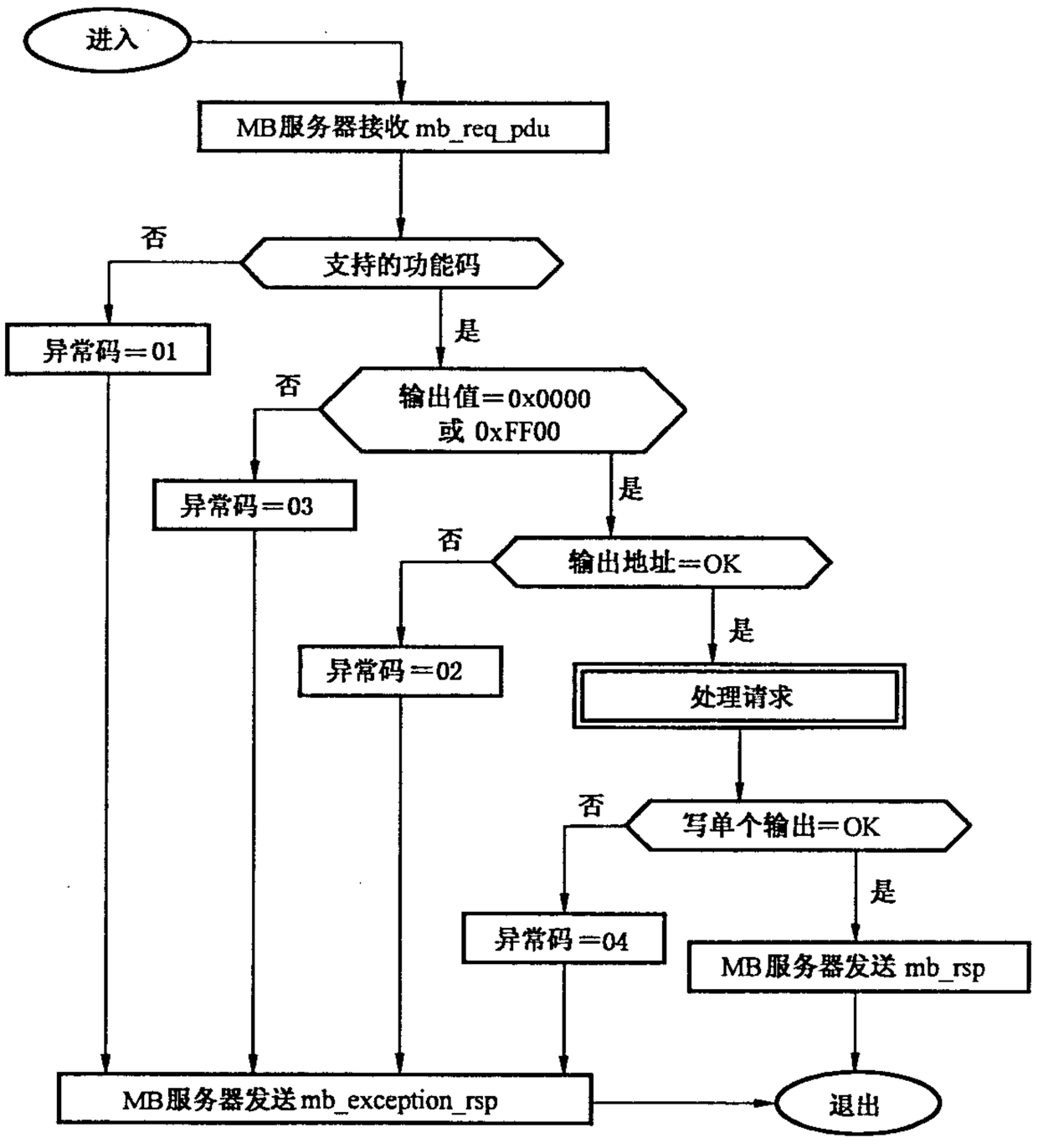


图 15　写单个线圈状态图

表 22 是一个请求写线圈 173 为 ON 的示例。

表 22　写单个线圈

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	05	功能	05
输出地址 Hi	00	输出地址 Hi	00
输出地址 Lo	AC	输出地址 Lo	AC
输出值 Hi	FF	输出值 Hi	FF
输出值 Lo	00	输出值 Lo	00

7.6 06(0x06)写单个寄存器

见图 16 和表 23～表 25。

使用该功能码在一个远程设备中写单个保持寄存器。

请求 PDU 指定了被写入寄存器的地址。从零开始寻址寄存器。因此,编号为 1 的寄存器被寻址为 0。

正常的响应是请求的复制,在写入寄存器内容之后被返回。

表 23 写单个寄存器请求

功能码	1 字节	0x06
寄存器地址	2 字节	0x0000～0xFFFF
寄存器值	2 字节	0x0000～0xFFFF

表 24 写单个寄存器响应

功能码	1 字节	0x06
寄存器地址	2 字节	0x0000～0xFFFF
寄存器值	2 字节	0x0000～0xFFFF

表 25 写单个寄存器错误响应

异常功能码	1 字节	0x86
异常码	1 字节	01 或 02 或 03 或 04

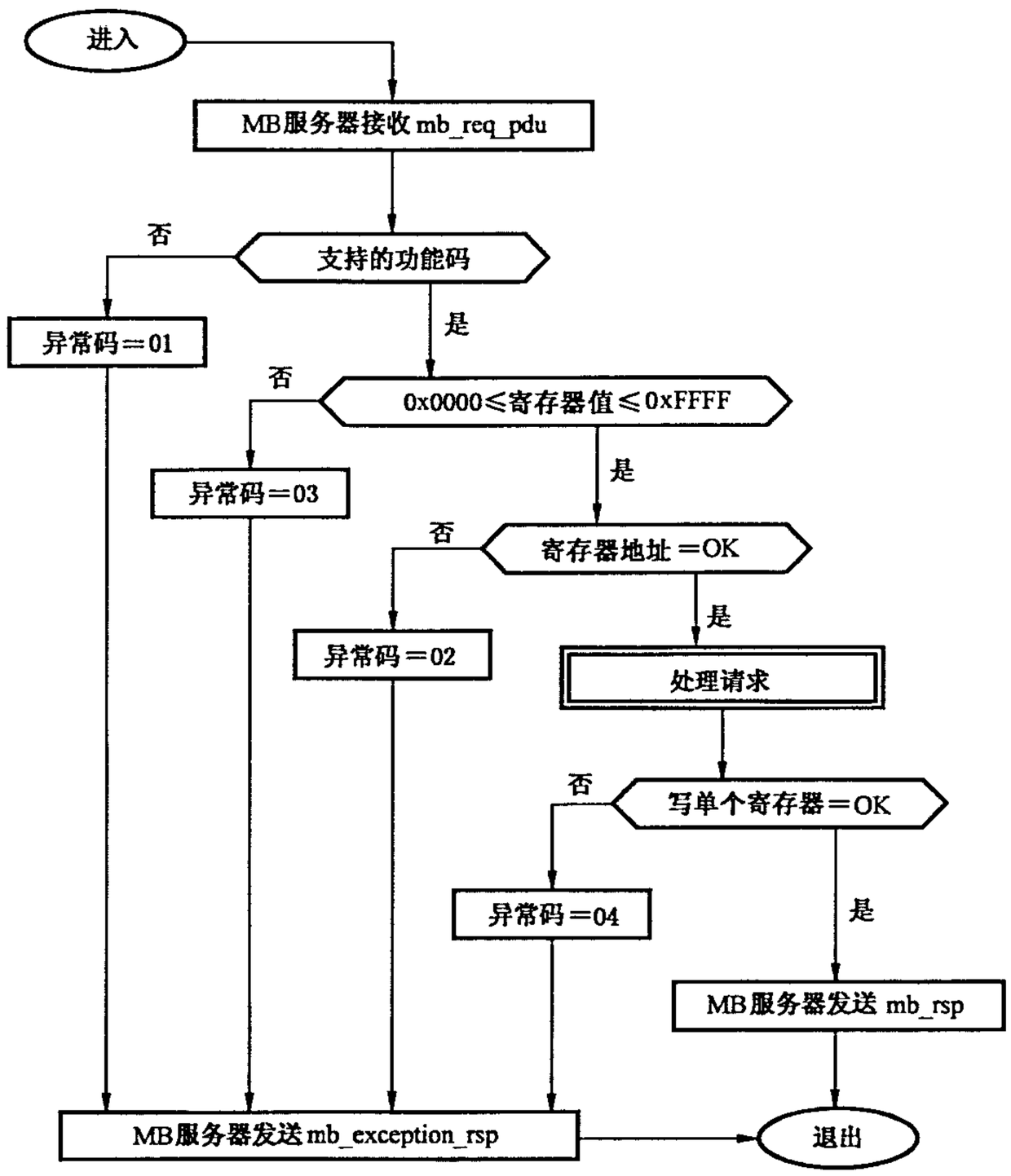


图 16 写单个寄存器状态图

表 26 是一个请求将十六进制 00 03 写入寄存器 2 的示例。

表 26　写入单个寄存器

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	06	功能	06
寄存器地址 Hi	00	输出地址 Hi	00
寄存器地址 Lo	01	输出地址 Lo	01
寄存器值 Hi	00	输出值 Hi	00
寄存器值 Lo	03	输出值 Lo	03

7.7　07(0x07)读异常状态(仅用于串行链路)

见图 17 和表 27～表 29。

使用这个功能码从一个远程设备中读 8 个异常状态输出的内容。

因为异常码输出引用是已知的(在这个功能中不需要输出引用),该功能提供访问这种信息的一种简单方法。

正常的响应包括 8 个异常状态输出的状态。将这些输出按每个输出一个位打包成一个数据字节。在该字节的最低有效位中包含最低位输出引用的状态。

8 个异常状态输出的内容是与设备相关的。

表 27　读异常状态请求

功能码	1 字节	0x07

表 28　读异常状态响应

功能码	1 字节	0x07
输出数据	1 字节	0x00～0xFF

表 29　读异常状态错误响应

异常功能码	1 字节	0x87
异常码	1 字节	01 或 04

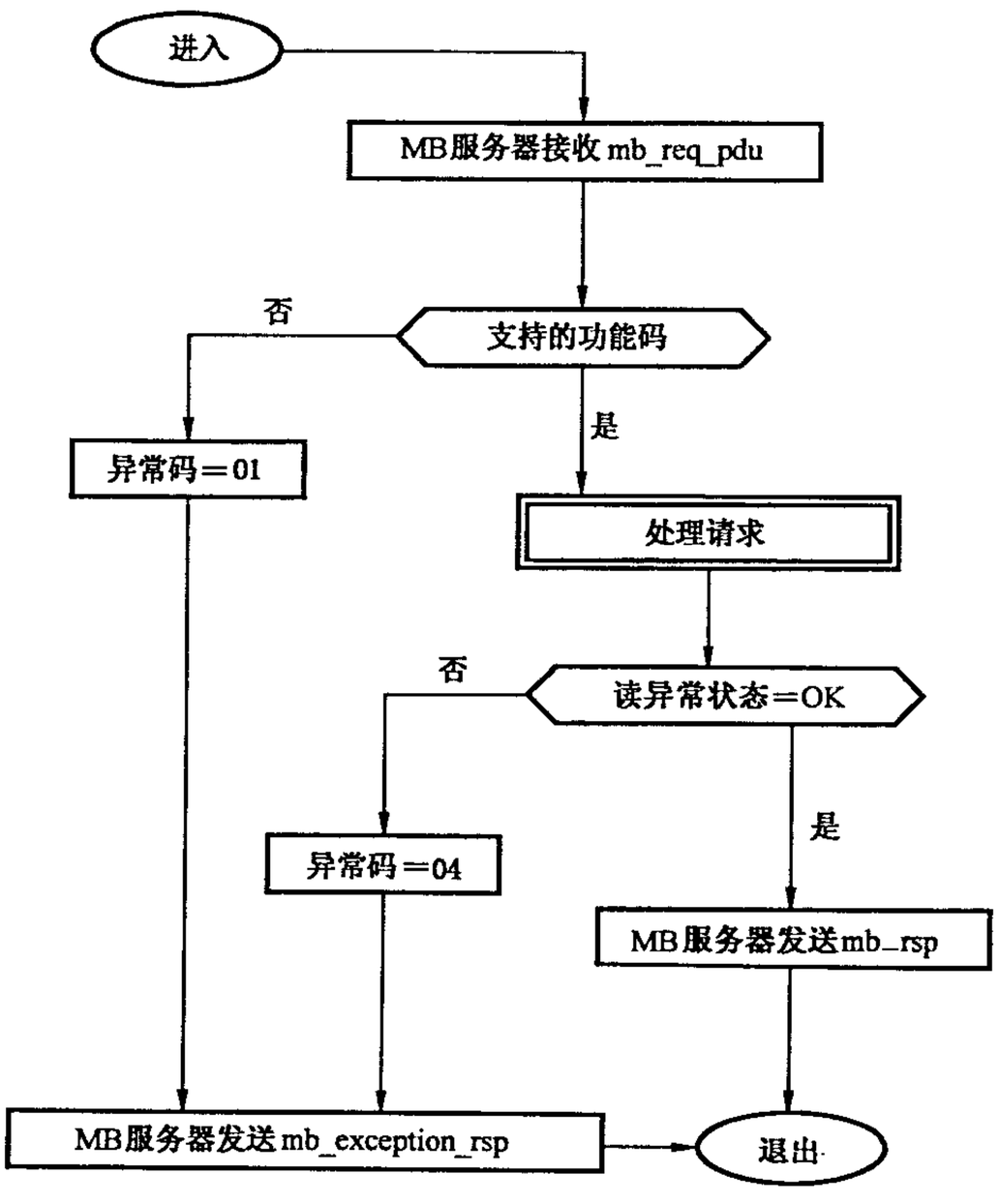


图 17　读异常状态的状态图

表 30 是一个请求读异常状态的示例。

表 30 读异常状态

请求		响应	
字段名	十六进制	字段名	十六进制
功能	07	功能	07
		输出数据	6D

在这个示例中，输出数据是十六进制 6D(二进制 0110 1101)。从左至右，输出为 OFF-ON-ON-OFF-ON-ON-OFF-ON。按从最高到最低寻址输出的方式显示状态。

7.8 08(0x08)诊断(仅用于串行链路)

见表 31～表 33。

Modbus 功能码 08 提供了一系列测试，用于检查客户机(主站)设备与服务器(从站)之间的通信系统，或检查服务器中的各种内部差错状态。

这个功能使用询问中的 2 个字节的子功能码字段来定义所要执行的测试类型。服务器在正常的响应中复制功能码和子功能码。某些诊断会导致远程设备通过正常响应的数据字段返回相应数据。

通常，向远程设备发送诊断功能并不影响远程设备中用户程序的运行。诊断不能访问用户逻辑，例如：离散量和寄存器。某些功能可以随意地复位远程设备中的差错计数器。

但是，可以强制服务器设备进入“只听模式”，在这种模式中服务器设备监视通信系统中的报文而不对报文进行响应。如果应用程序与远程设备更多的数据交换有关，就可能影响应用程序的结果。一般情况下，这种模式被强制用来从通信系统中去除有故障的远程设备。

下列诊断功能专门用于串行链路设备。

对返回询问数据请求的正常响应是回送相同的数据。同时还复制功能码和子功能码。

表 31 诊断请求

功能码	1 字节	0x08
子功能	2 字节	…
数据	N×2 字节	…

表 32 诊断响应

功能码	1 字节	0x08
子功能	2 字节	…
数据	N×2 字节	…

表 33 诊断错误响应

异常功能码	1 字节	0x88
异常码	1 字节	01 或 03 或 04

7.8.1 串行链路设备支持的子功能码

表 34 是由串行链路设备所支持的子功能码列表。所列出的每个子功能码均附带用于诊断的数据字段内容的示例。

表 34 串行链路设备支持的子功能码

子功能码		名称
十六进制	十进制	
00	00	返回询问数据
01	01	重新启动通信选项
02	02	返回诊断寄存器

表 34(续)

子功能码		名称
十六进制	十进制	
03	03	改变 ASCII 输入分隔符
04	04	强制只听模式
	05…09	保留
0A	10	清除计数器和诊断寄存器
0B	11	返回总线报文计数
0C	12	返回总线通信差错计数
0D	13	返回总线异常差错计数
0E	14	返回从站报文计数
0F	15	返回从站无响应计数
10	16	返回从站 NAK 计数
11	17	返回从站忙计数
12	18	返回总线字符超限计数
13	19	保留
14	20	清除超限计数器和标志
N. A.	21… 65535	保留

a) 00 返回询问数据

在响应中返回(回送)请求数据字段中传递的数据。全部响应报文应该与请求报文一致。

子功能	数据字段(请求)	数据字段(响应)
00 00	任意	复制请求数据

b) 01 重新启动通信选项

必须初始化和重新启动远程设备串行链路端口,并且清除所有通信事件计数器。如果端口正在只听模式下,不返回响应。它是唯一的能将端口退出只听模式的功能。如果端口不是处于只听模式,在进行重新启动之前返回正常的响应。

当远程设备接收到请求时,它设法进行重新启动,并进行加电确认测试。成功地完成测试后,端口进入联机状态。

请求数据字段内容是十六进制 FF 00 时,能够清除端口通信事件记录。内容是 00 00 时,使记录保持在与重新启动之前相同。

子功能	数据字段(请求)	数据字段(响应)
00 01	00 00	复制请求数据
00 01	FF 00	复制请求数据

c) 02 返回诊断寄存器

在响应中返回远程设备的 16 位诊断寄存器内容。

子功能	数据字段(请求)	数据字段(响应)
00 02	00 00	诊断寄存器内容

d) 03 改变 ASCII 输入分隔符

请求数据字段中传递的字符“CHAR”作为报文结束分隔符(替代默认的 LF 字符),供以后报文使用。在 ASCII 报文结束不要求换行的情况下,使用这种功能。

子功能	数据字段(请求)	数据字段(响应)
00 03	CHAR 00	复制请求数据

e) 04 强制只听模式

强制被寻址的远程设备进入 Modbus 通信的只听模式。这就使远程设备与网络中的其他设备断开，允许网络中的其他设备继续通信，而没有来自被寻址远程设备的中断。不返回响应。

当远程设备进入只听模式时，关闭所有激活的通信控制。允许就绪看门狗定时器超时，锁定控制关断。当设备在这种模式下，监视对设备寻址的 Modbus 报文或广播，但不进行任何操作，也不发送任何响应。

进入这种模式后唯一可处理的功能是重新启动通信选项功能(功能码 8，子功能 1)。

子功能	数据字段(请求)	数据字段(响应)
00 04	00 00	没有返回的响应

f) 10(十六进制 0A)清除计数器和诊断寄存器

目的是清除所有计数器和诊断寄存器。加电时也会清除计数器。

子功能	数据字段(请求)	数据字段(响应)
00 0A	00 00	复制响应数据

g) 11(十六进制 0B)返回总线报文计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后远程设备在通信系统中检测到的报文数量。

子功能	数据字段(请求)	数据字段(响应)
00 0B	00 00	全部报文计数

h) 12(十六进制 0C)返回总线通信差错计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后远程设备遇到的 CRC 出错数量。

子功能	数据字段(请求)	数据字段(响应)
00 0C	00 00	CRC 差错计数

i) 13(十六进制 0D)返回总线异常差错计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后远程设备返回的 Modbus 异常响应数量。

在第 8 章中详细说明和列出了异常响应。

子功能	数据字段(请求)	数据字段(响应)
00 0D	00 00	异常差错计数

j) 14(十六进制 0E)返回从站报文计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后对远程设备寻址的报文数量或远程设备处理的广播报文数量。

子功能	数据字段(请求)	数据字段(响应)
00 0E	00 00	从站报文计数

k) 15(十六进制 0F)返回从站无响应计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后对没有返回响应(既没有正常响应也没有异常响应)的远程设备寻址的报文数量。

子功能	数据字段(请求)	数据字段(响应)
00 0F	00 00	从站无响应计数

l) 16(十六进制 10)返回从站 NAK 计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后对返回 NAK 异常响应的远程设备寻址的报文数量。在第 8 章中详细说明并列出了异常响应。

子功能	数据字段(请求)	数据字段(响应)
00 10	00 00	从站 NAK 计数

m) 17(十六进制 11)返回从站忙计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后对返回从站设备忙异常响应的远程设备寻址的报文数量。

子功能	数据字段(请求)	数据字段(响应)
00 11	00 00	从站设备忙计数

n) 18(十六进制 12)返回总线字符超限计数

响应数据字段返回上一次重启动、清除计数器操作或加电之后,对由于字符超限状况而无法处理的远程设备寻址的报文数量。字符抵达端口的速度高于存储字符的速度或由于硬件故障而丢失字符,均产生字符超限。

子功能	数据字段(请求)	数据字段(响应)
00 12	00 00	从站字符超限计数

o) 20(十六进制 14)清除超限计数器和标志

清除超限差错计数器,并复位出错标志。

子功能	数据字段(请求)	数据字段(响应)
00 14	00 00	返回请求数据

7.8.2 示例和状态图

见图 18 和表 35。

这是一个请求远程设备返回询问数据的示例。它使用子功能码 0(两个字节字段的十六进制 00 00)。用两个字节数据字段(十六进制 A5 37)发送需要返回的数据。

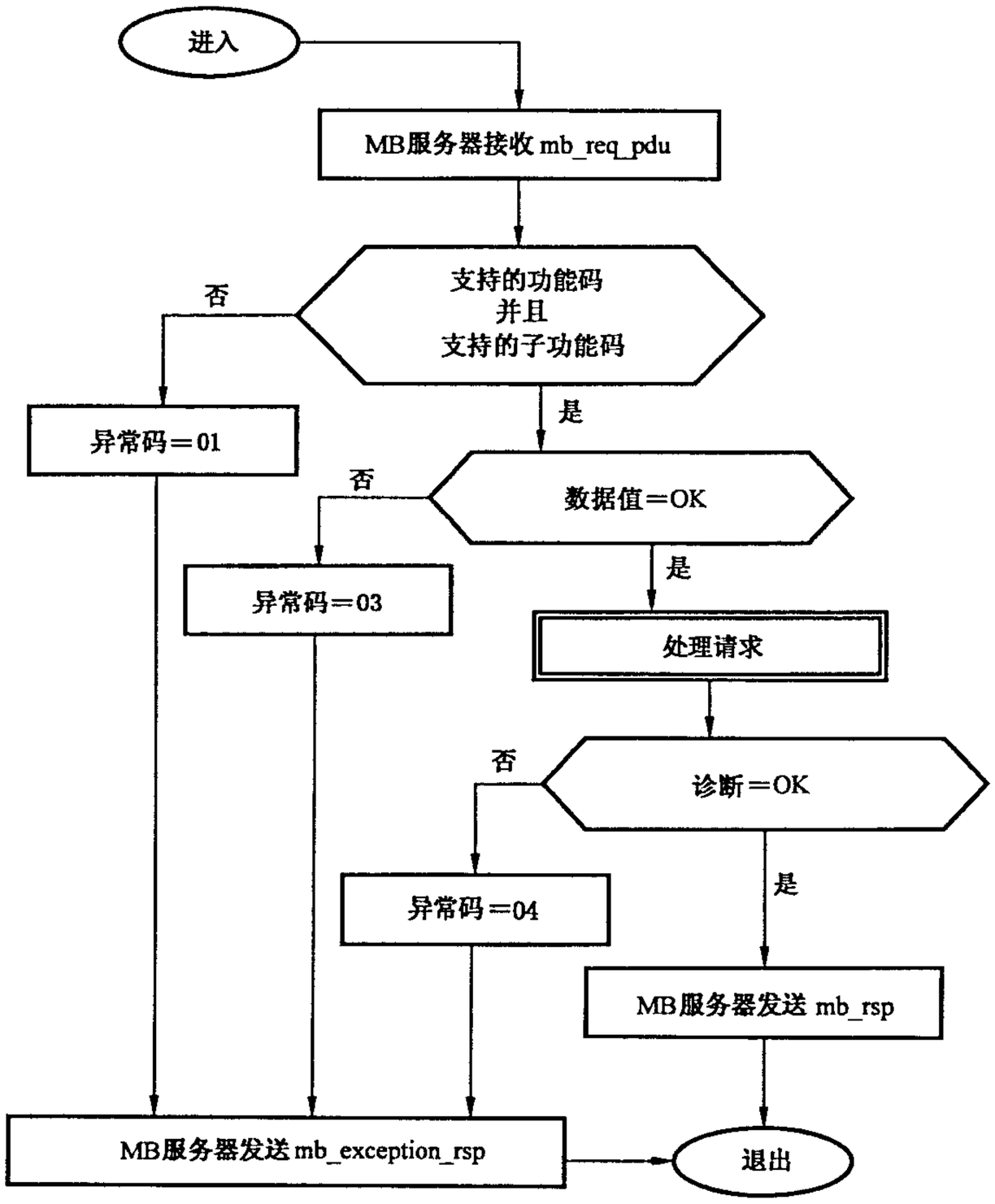


图 18 诊断状态图

表 35　请求远程设备返回询问数据

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	08	功能	08
子功能 Hi	00	子功能 Hi	00
子功能 Lo	00	子功能 Lo	00
数据 Hi	A5	数据 Hi	A5
数据 Lo	37	数据 Lo	37

对其他类型询问响应的数据字段可能包含差错计数或子功能码所请求的其他数据。

7.9　11(0x0B)获得通信事件计数器(仅用于串行链路)

见图 19 和表 36～表 38。

使用这个功能码从远程设备通信事件计数器中获取状态字和事件计数的值。

通过在一系列报文之前和之后获取的当前计数,客户机可以确定远程设备是否正常地处理报文。

对于每次成功地完成报文传输,就将远程事件计数器增加 1。对于异常响应、轮询命令或读事件计数器命令,不改变计数器。

通过附带子功能重新启动通信选项(码 00 01)或清除计数器和诊断寄存器(码 00 0A)的诊断功能(码 08),可以复位事件计数器。

正常响应包括两字节状态字和两字节事件计数。如果远程设备一直在处理前面发出的程序命令(处在忙状态),那么状态字将全为 1(十六进制 FF FF)。否则,状态字全为 0。

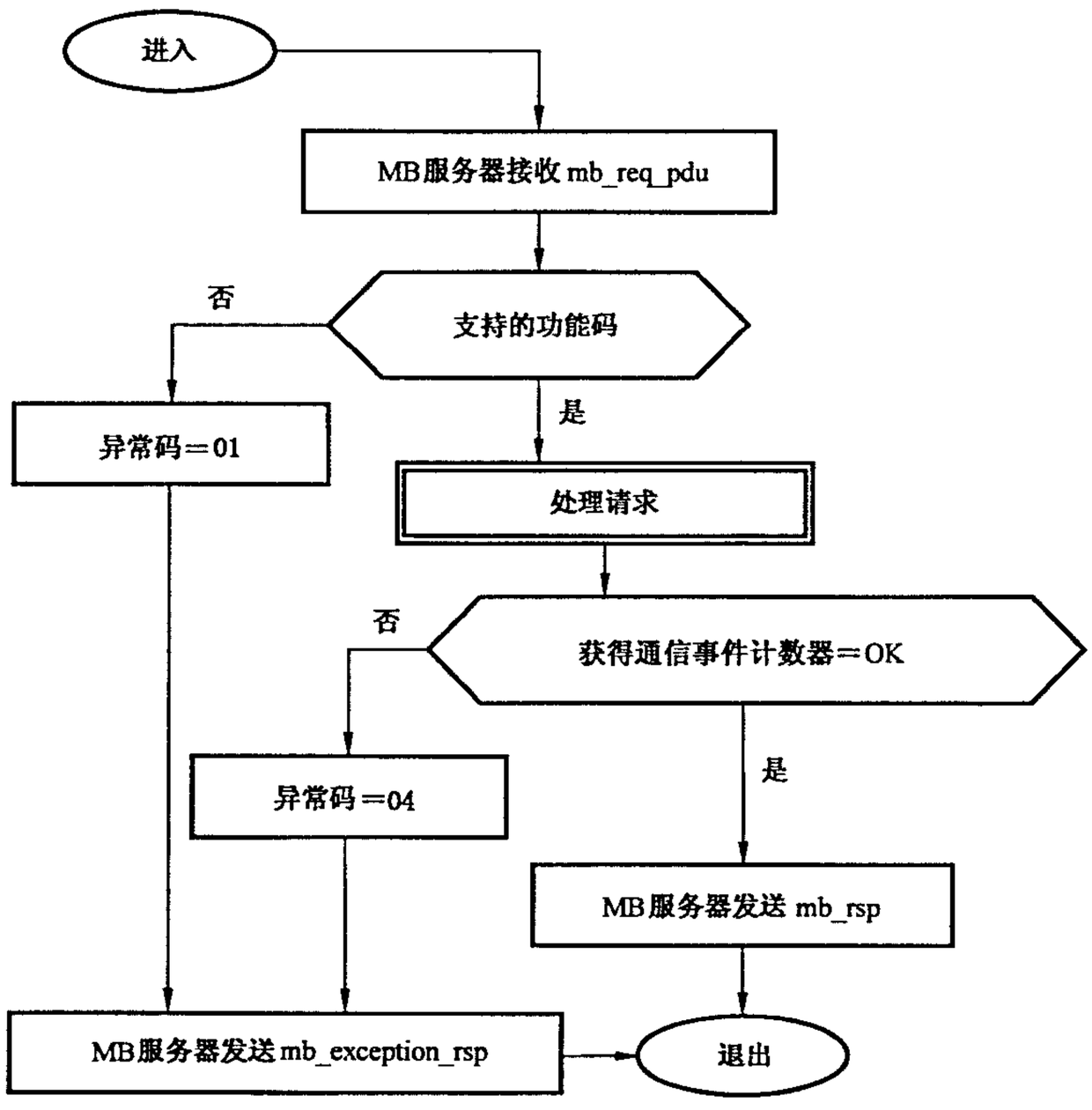


图 19　获得通信事件计数器状态图

表 36　获得通信事件计数器请求

功能码	1 字节	0x0B

表 37　获得通信事件计数器响应

功能码	1 字节	0x0B
状态	2 字节	0x0000～0xFFFF
事件计数	2 字节	0x0000～0xFFFF

表 38　获得通信事件计数器错误响应

异常功能码	1 字节	0x8B
异常码	1 字节	01 或 04

表 39 是一个请求获得远程设备中通信事件计数器的示例。

表 39　获得远程设备中通信事件计数器

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	0B	功能	0B
		状态 Hi	FF
		状态 Lo	FF
		事件计数 Hi	01
		事件计数 Lo	08

在该示例中，状态字是十六进制 FF FF，表示在远程设备中仍然执行程序功能。事件计数表示设备已经记录了 264(十六进制 01 08)个事件。

7.10　12(0x0C)获得通信事件记录(仅用于串行链路)

见图 20 和表 40～表 42。

使用这个功能码从远程设备中获得状态字、事件计数、报文计数以及一个事件字节字段。

该状态字和事件计数与获得通信事件计数器功能码(十六进制 11 0B)返回的信息一致。

报文计数器包含上一次重启动、清除计数器操作或加电之后远程设备处理的报文数量。报文计数与诊断功能(码 08)、子功能返回总线报文计数(十六进制码 11 0B)返回的报文计数一致。

事件字节字段包含 0～64 个字节，每个字节与远程设备的一个 Modbus 发送或接收操作的状态对应。远程设备按时间顺序将事件放入字段中。字节 0 是最新事件。每个新字节替代字段中的最旧字节。

正常响应包含一个 2 字节状态字字段、一个 2 字节事件计数字段、一个 2 字节报文计数字段以及一个含有 0～64 个字节事件的字段。一个字节计数字段定义了这四个字段中的数据总长度。

表 40　获得通信事件记录请求

功能码	1 字节	0x0C

表 41　获得通信事件记录响应

功能码	1 字节	0x0C
字节计数	1 字节	N[a]
状态	2 字节	0x0000～0xFFFF
事件计数	2 字节	0x0000～0xFFFF
报文计数	2 字节	0x0000～0xFFFF
事件	(N－6)×1 字节	…
[a] N＝事件数量＋3×2 个字节(状态、事件计数和报文计数的长度)。		

表 42　获得通信事件记录错误响应

异常功能码	1 字节	0x8C
异常码	1 字节	01 或 04

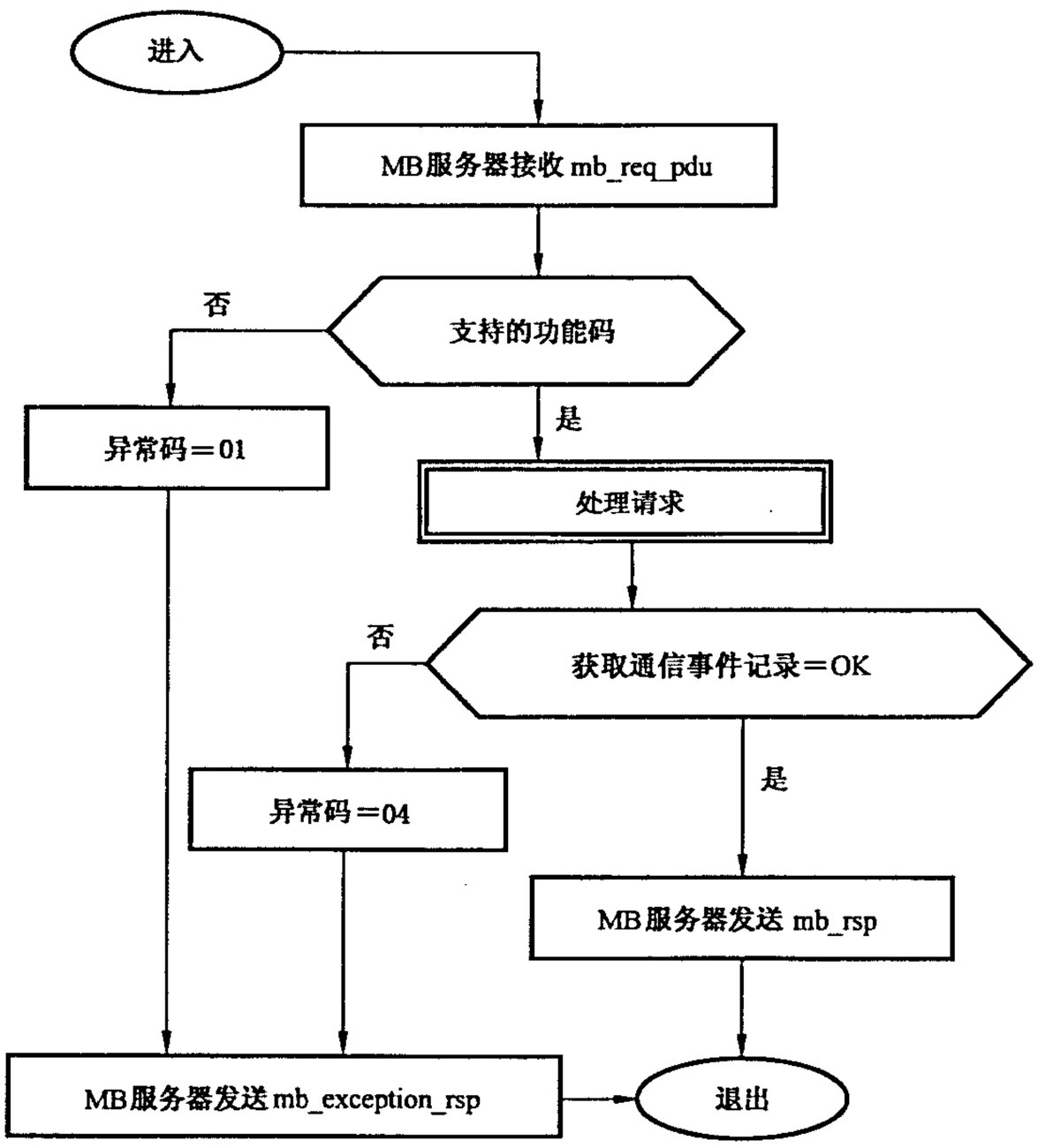


图 20　获取通信事件记录状态图

表 43 是一个请求获得远程设备中通信事件记录的示例。

表 43　获得远程设备中通信事件记录

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	0C	功能	0C
		字节计数	08
		状态 Hi	00
		状态 Lo	00
		事件计数 Hi	01
		事件计数 Lo	08
		报文计数 Hi	01
		报文计数 Lo	21
		事件 0	20
		事件 1	00

在这个示例中，状态字是十六进制 00 00，表示远程设备没有正在处理程序功能。事件计数表示远程设备已经计数 264(十六进制 01 08)个事件。报文计数表示已经处理了 289(十六进制 01 21)个报文。

用事件 0 字节表示最新通信事件，其内容(十六进制 20)表示远程设备最近已进入只听模式。

用事件 1 字节表示前一个事件，其内容(十六进制 00)表示远程设备曾接收一个通信重启动。

下面说明响应事件字节的格式。

事件字节包含的内容

由获得通信事件记录功能返回的事件字节可以是四种类型中的任何一种。类型由每个字节中的位 7(高位位)定义，还可以由位 6 进一步定义。下面详细解释这四种类型。

a)　远程设备 Modbus 接收事件

当接收询问报文时，远程设备存储这种类型的事件字节。在远程设备处理报文之前存储。通过将位 7 设置为逻辑“1”来定义这种事件。其他位在相应状态为“真”时被设置为逻辑“1”。字节的格式为：

位	内容
0	未使用
1	通信错误
2	未使用
3	未使用
4	字符超限
5	当前在只听模式下
6	收到广播
7	1

b)　远程设备 Modbus 发送事件

当处理完请求报文时，远程设备存储这种类型的事件字节。在远程设备返回正常响应/异常响应或没有响应后存储。通过将位 7 设置为逻辑“0”、将位 6 设置为逻辑“1”来定义这种事件。其他位在相应状态为“真”时被设置为逻辑“1”。字节的格式为：

位	内容
0	读异常发送(异常码 1～3)
1	从站中断异常发送(异常码 4)
2	从站忙异常发送(异常码 5～6)
3	从站程序 NAK 异常发送(异常码 7)
4	出现的写超时错误
5	当前在只听模式下
6	1
7	0

c) 远程设备进入只听模式

当进入只听模式时,远程设备存储这种类型的事件字节。该事件被定义为十六进制 04。

d) 远程设备开始重启通信

当重新启动通信端口时,远程设备存储这种类型的事件字节。通过附带子功能重新启动通信选项(码 00 01)的诊断功能(码 08)可以重新启动远程设备。

该功能还能将远程设备置于“错误状态下继续”或“错误状态下停止”模式。如果将远程设备置于“错误状态下继续”模式,则将事件字节加入现存事件记录中。如果将远程设备置于“错误状态下停止”模式,则将事件字节加入事件记录中,并且将其余记录清 0。

位 0 的内容定义了该事件。

7.11 15(0x0F)写多个线圈

见图 21 和表 44～表 46。

使用该功能码将一个远程设备中的一个线圈序列的每个线圈强制为 ON 或 OFF。请求 PDU 指定了被强制的线圈引用。从零开始寻址线圈。因此,编号为 1 的线圈 1 被寻址为 0。

请求数据字段的内容指定了被请求的 ON/OFF 状态。数据字段中为逻辑“1”的位请求相应输出为 ON;为逻辑“0”的位请求相应输出为 OFF。

正常的响应返回功能码、起始地址以及被强制的线圈数量。

表 44 写多个线圈请求 PDU

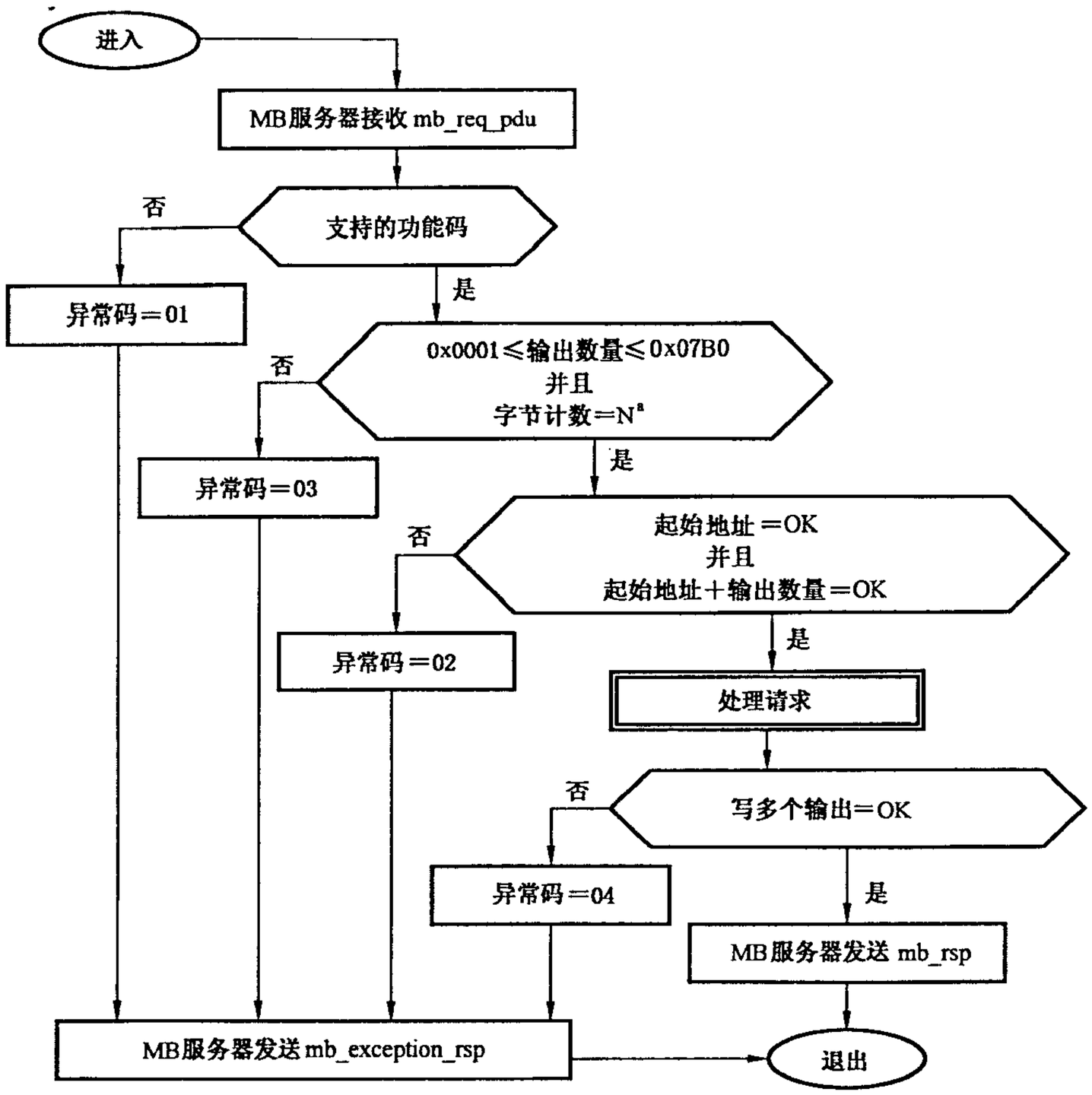
功能码	1 字节	0x0F
起始地址	2 字节	0x0000～0xFFFF
输出数量	2 字节	0x0001～0x07B0
字节计数	1 字节	N[a]
输出值	N[a]×1 字节	
[a] N=输出数量/8,如果余数不等于 0,那么 N=N+1。		

表 45 写多个线圈响应 PDU

功能码	1 字节	0x0F
起始地址	2 字节	0x0000～0xFFFF
输出数量	2 字节	0x0001～0x07B0

表 46 写多个线圈错误响应

异常功能码	1 字节	0x8F
异常码	1 字节	01 或 02 或 03 或 04



[a] N＝输出数量/8，如果余数不等于 0，那么 N＝N＋1。

图 21　写多个输出的状态图

这是一个请求从线圈 20 开始写入 10 个线圈的示例。

请求的数据内容为两个字节：十六进制 CD 01(二进制 1100 1101 0000 0001)。二进制位按如下方式对应于输出。

位：	1	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1
输出：	27	26	25	24	23	22	21	20	—	—	—	—	—	—	29	28

传输的第一字节(十六进制 CD)对应于输出 27～20，最低有效位对应于最低输出(20)。

传输的下一字节(十六进制 01)对应于输出 29～28，最低有效位对应于最低输出(28)。应该用零填充最后数据字节中的未使用位，见表 47。

表 47　用零填充最后数据字节中的未使用位

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	0F	功能	0F
起始地址 Hi	00	起始地址 Hi	00
起始地址 Lo	13	起始地址 Lo	13
输出数量 Hi	00	输出数量 Hi	00
输出数量 Lo	0A	输出数量 Lo	0A
字节计数	02		
输出值 Hi	CD		
输出值 Lo	01		

7.12 16(0x10) 写多个寄存器

见图 22 和表 48～表 50。

使用该功能码在一个远程设备中写连续寄存器块(1～123 个寄存器)。

在请求数据字段中指定了请求写入的值。将数据按每个寄存器两字节打包。

正常的响应返回功能码、起始地址以及被写入寄存器的数量。

表 48 写多个寄存器请求

功能码	1 字节	0x10
起始地址	2 字节	0x0000～0xFFFF
寄存器数量	2 字节	0x0001～0x007B
字节计数	1 字节	2×N[a]
寄存器值	N[a]×2 字节	值
[a] N=寄存器数量。		

表 49 写多个寄存器响应

功能码	1 字节	0x10
起始地址	2 字节	0x0000～0xFFFF
寄存器数量	2 字节	1～123(0x7B)

表 50 写多个寄存器错误响应

异常功能码	1 字节	0x90
异常码	1 字节	01 或 02 或 03 或 04

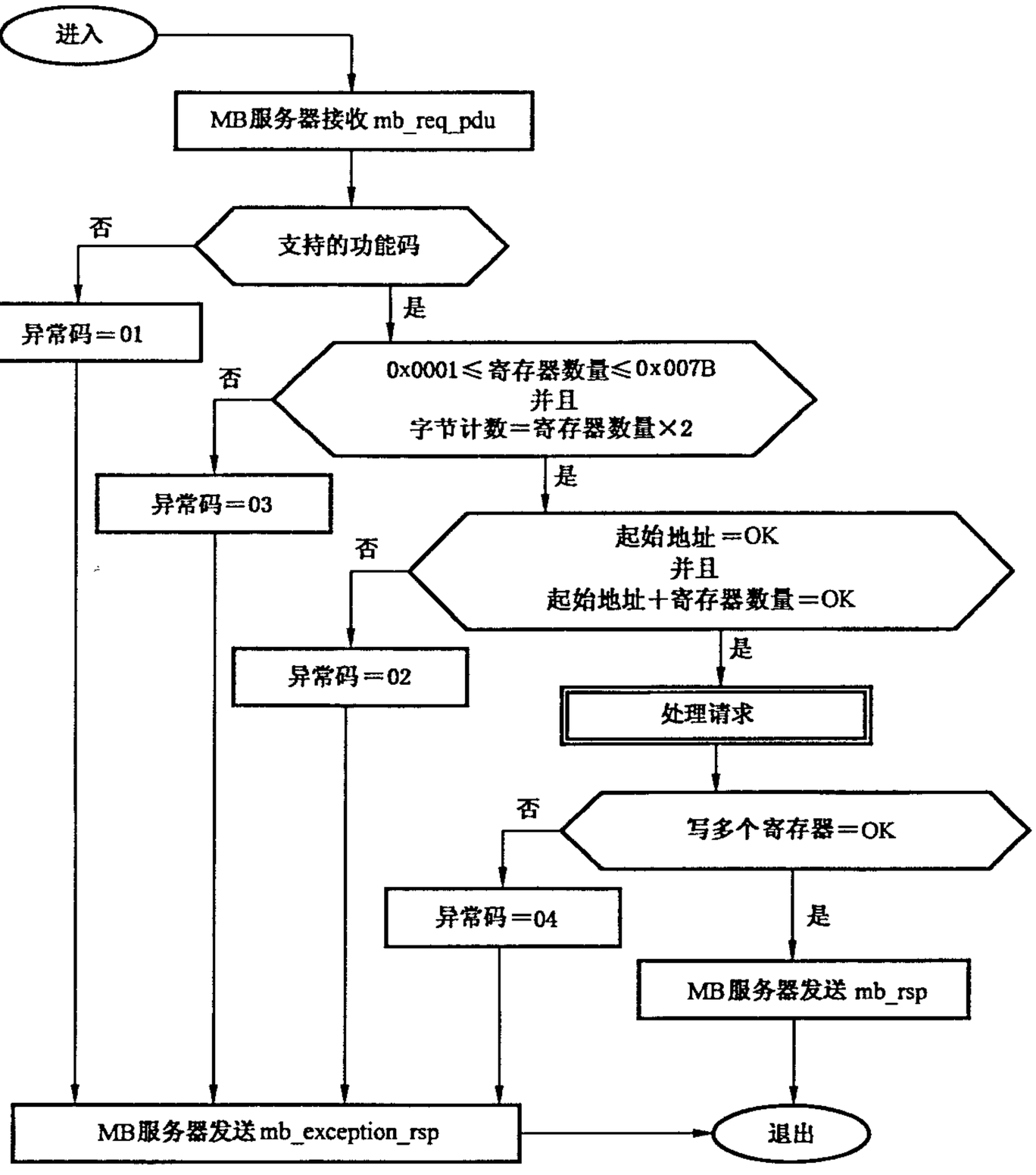


图 22 写多个寄存器状态图

表 51 是一个请求将十六进制 00 0A 和 01 02 写入以第 2 个寄存器开始的两个寄存器的示例。

表 51　写入连续两个寄存器

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	10	功能	10
起始地址 Hi	00	起始地址 Hi	00
起始地址 Lo	01	起始地址 Lo	01
寄存器数量 Hi	00	寄存器数量 Hi	00
寄存器数量 Lo	02	寄存器数量 Lo	02
字节计数	04		
寄存器值 Hi	00		
寄存器值 Lo	0A		
寄存器值 Hi	01		
寄存器值 Lo	02		

7.13　17(0x11)报告从站 ID(仅用于串行链路)

见图 23 和表 52～表 54。

使用这个功能码读远程设备特定的类型描述、当前状态以及其他信息。

下列示例表示了正常响应的格式。每种类型设备有其特定的数据内容。

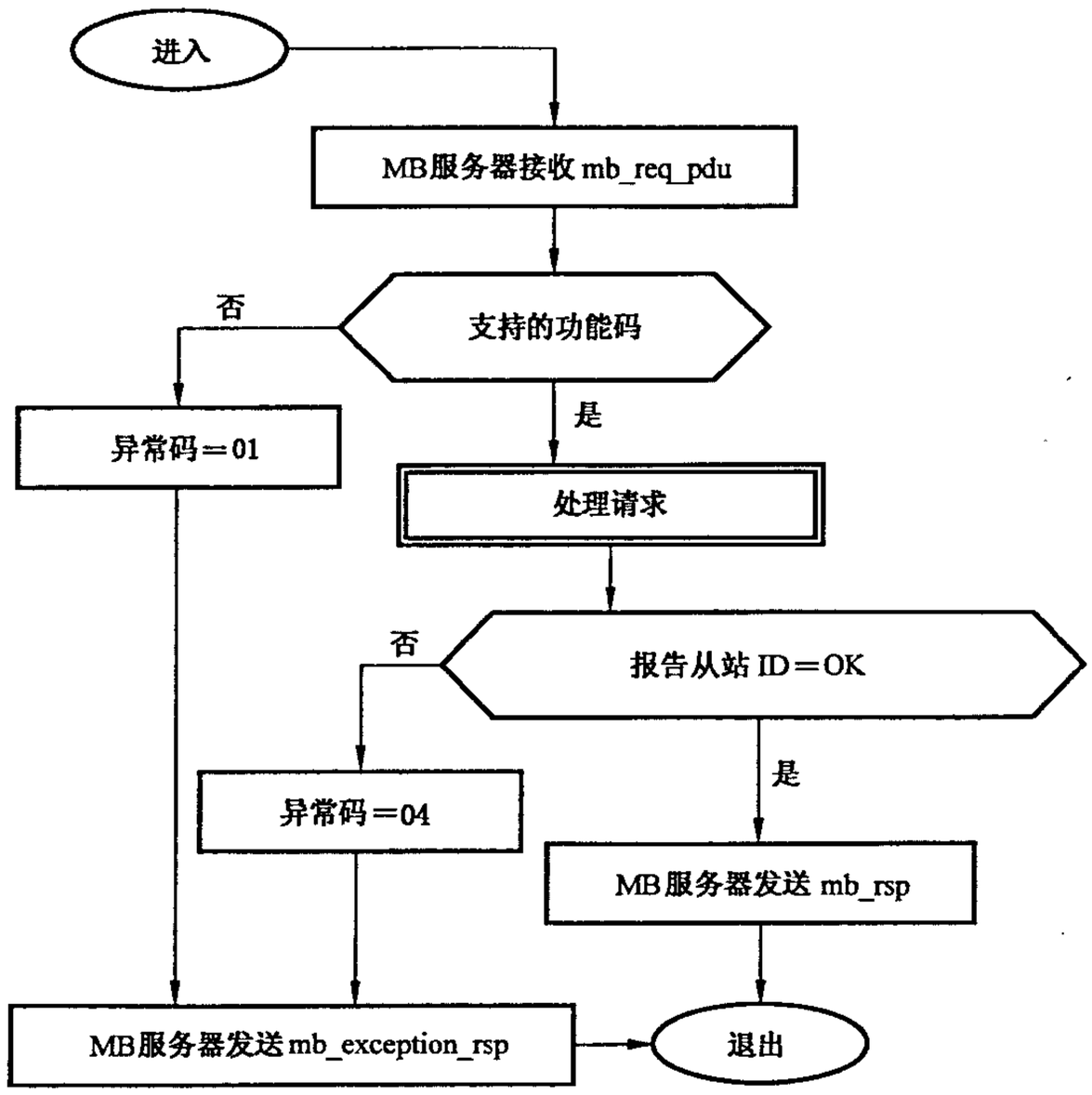


图 23　报告从站 ID 状态图

表 52　报告从站 ID 请求

功能码	1 字节	0x11

表 53　报告从站 ID 响应

功能码	1 字节	0x11
字节计数	1 字节	…
从站 ID	与特定设备相关	…
运行指示状态	1 字节	0x00＝OFF,0xFF＝ON
附加数据	…	…

表 54　报告从站 ID 错误响应

异常功能码	1 字节	0x91
异常码	1 字节	01 或 04

表 55 是一个请求报告 ID 和状态的示例。

表 55　请求报告 ID 和状态

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	11	功能	11
		字节计数	与特定设备相关
		从站 ID	与特定设备相关
		运行指示器状态	0x00 或 0xFF
		附加数据	与特定设备相关

7.14　20(0x14) 读文件记录

见图 24 和表 56～表 58。

使用该功能码读文件记录。根据字节数量提供所有请求数据长度，并且根据寄存器提供所有记录长度。

文件采取记录的结构。每个文件包括 10 000 个记录，寻址这些记录为十进制 0000～9999 或十六进制 0x0000～0x270F，例如：记录 12 被寻址为 12。

该功能可以读多个引用组。这些组可以是分散的(不连续的)，但每组中的引用必须是连续的。

用含有 7 个字节的单独的“子请求”字段定义每个组：

——引用类型：1 个字节(必须指定为 6)；

——文件号：2 个字节；

——文件中的起始记录号：2 个字节；

——所读记录的长度：2 个字节。

被读取的寄存器数量加上预期响应中的所有其他字段不能超过 Modbus 报文 PDU 允许的长度：253 个字节。

正常的响应是一系列“子响应”，它们与“子请求”一一对应。字节计数字段是所有“子响应”中全部字节数。另外，每个“子响应”都包括一个表示其自身字节计数的字段。

表 56　读文件记录请求

功能码	1 字节	0x14
字节计数	1 字节	0x07～0xF5 字节
子请求 x,引用类型	1 字节	06
子请求 x,文件号	2 字节	0x0001～0xFFFF
子请求 x,记录号	2 字节	0x0000～0x270F
子请求 x,记录长度	2 字节	N
子请求 x+1…	…	…

表 57　读文件记录响应

功能码	1 字节	0x14
响应数据长度	1 字节	0x07～0xF5
子请求 x,文件响应长度	1 字节	0x05～0xF5(RUDY)
子请求 x,引用类型	1 字节	06
子请求 x,记录数据	N×2 字节	…
子请求 x+1…	…	…

表 58　读文件记录错误响应

异常功能码	1 字节	0x94
异常码	1 字节	01 或 02 或 03 或 04 或 08

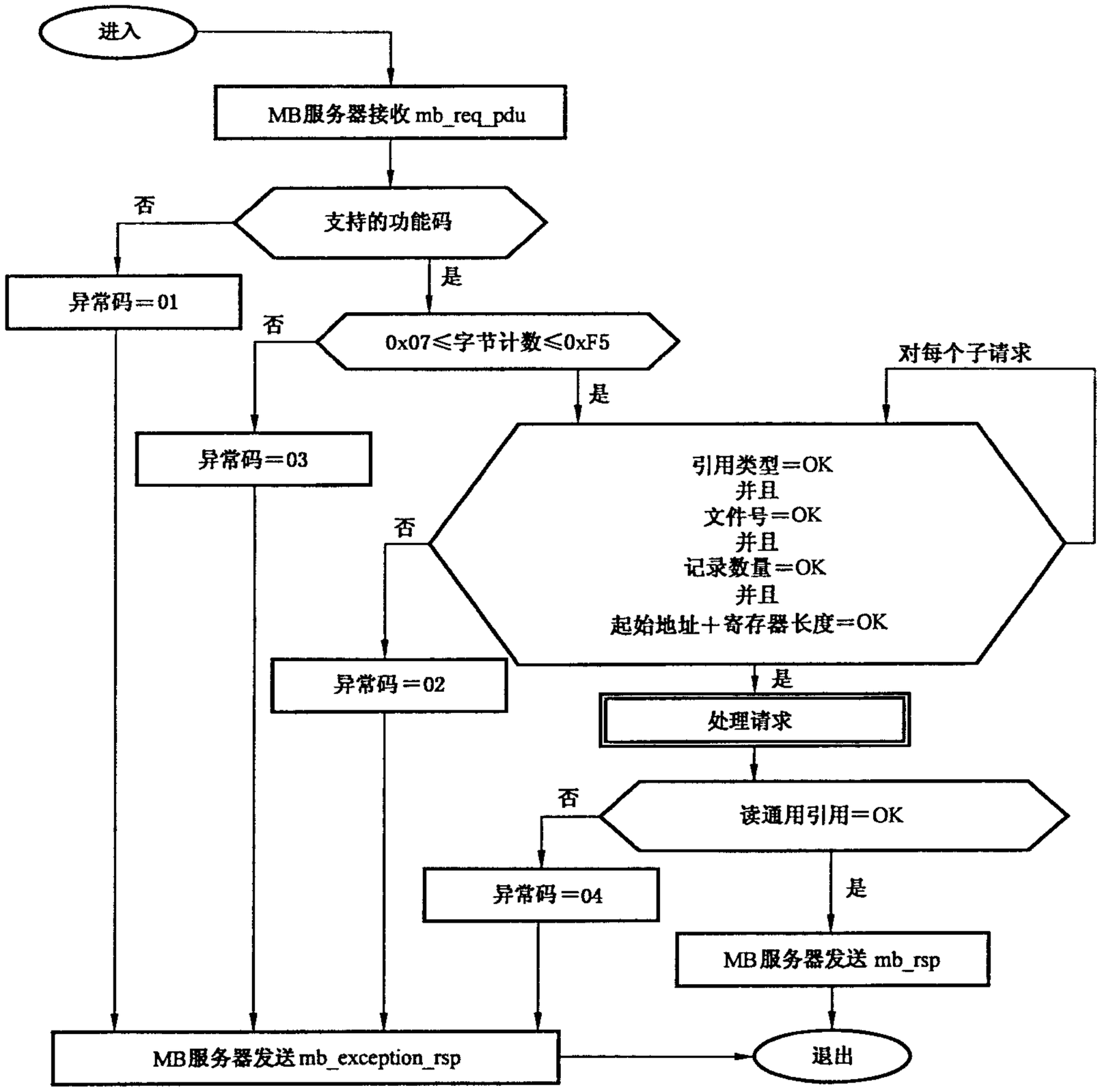


图 24　读文件记录状态图

表 59 是一个请求从远程设备读两个引用组的示例：

组 1 包括文件 4 中的 2 个寄存器，以寄存器 1 开始(地址 0001)。

组 2 包括文件 3 中的 2 个寄存器，以寄存器 9 开始(地址 0009)。

表 59 请求从远程设备读两个引用组

请求		响应	
字段名	十六进制	字段名	十六进制
功能	14	功能	14
字节计数	0E	响应数据长度	0C
子请求 1,引用类型	06	子请求 1,文件响应长度	05
子请求 1,文件号 Hi	00	子请求 1,引用类型	06
子请求 1,文件号 Lo	04	子请求 1,寄存器数据 Hi	0D
子请求 1,记录号 Hi	00	子请求 1,寄存器数据 Lo	FE
子请求 1,记录号 Lo	01	子请求 1,寄存器数据 Hi	00
子请求 1,记录长度 Hi	00	子请求 1,寄存器数据 Lo	20
子请求 1,记录长度 Lo	02	子请求 2,文件响应长度	05
子请求 2,引用类型	06	子请求 2,引用类型	06
子请求 2,文件号 Hi	00	子请求 2,寄存器数据 Hi	33
子请求 2,文件号 Lo	03	子请求 2,寄存器数据 Lo	CD
子请求 2,记录号 Hi	00	子请求 2,寄存器数据 Hi	00
子请求 2,记录号 Lo	09	子请求 2,寄存器数据 Lo	40
子请求 2,记录长度 Hi	00		
子请求 2,记录长度 Lo	02		

7.15 21(0x15) 写文件记录

见图 25 和表 60～表 62。

使用该功能码进行文件记录写入。根据字节数量提供所有请求数据长度，并且根据 16 位字的数量提供所有记录长度。

文件采取记录的结构。每个文件包括 10 000 个记录，寻址这些记录为十进制 0000～9999 或十六进制 0x0000～0x270F，例如：记录 12 被寻址为 12。

该功能可以写多个引用组。这些组可以是分散的，即不连续的，但每组内的引用必须是连续的。

用含有 7 个字节和数据的单独“子请求”字段定义每个组：

引用类型：1 个字节(必须指定为 6)。

文件号：2 个字节。

文件中的起始记录号：2 个字节。

所写入记录的长度：2 个字节。

被写入的数据：每个寄存器为 2 字节。

被写入的寄存器数量加上请求中的所有其他字段不能超过 Modbus 报文 PDU 允许的长度：253 个字节。

正常的响应是请求的复制。

表 60　写文件记录请求

功能码	1 字节	0x15
请求数据长度	1 字节	0x09～0xFB
子请求 x,引用类型	1 字节	06
子请求 x,文件号	2 字节	0x0001～0xFFFF
子请求 x,记录号	2 字节	0x0000～0x270F
子请求 x,记录长度	2 字节	N
子请求 x,记录数据	N×2 字节	…
子请求 x+1…	…	…

表 61　写文件记录响应

功能码	1 字节	0x15
响应数据长度	1 字节	0x09～0xFB
子请求 x,引用类型	1 字节	06
子请求 x,文件号	2 字节	0x0001～0xFFFF
子请求 x,记录号	2 字节	0x0000～0x270F
子请求 x,记录长度	2 字节	N
子请求 x,记录数据	N×2 字节	…
子请求 x+1…	…	…

表 62　写文件记录错误响应

异常功能码	1 字节	0x95
异常码	1 字节	01 或 02 或 03 或 04 或 08

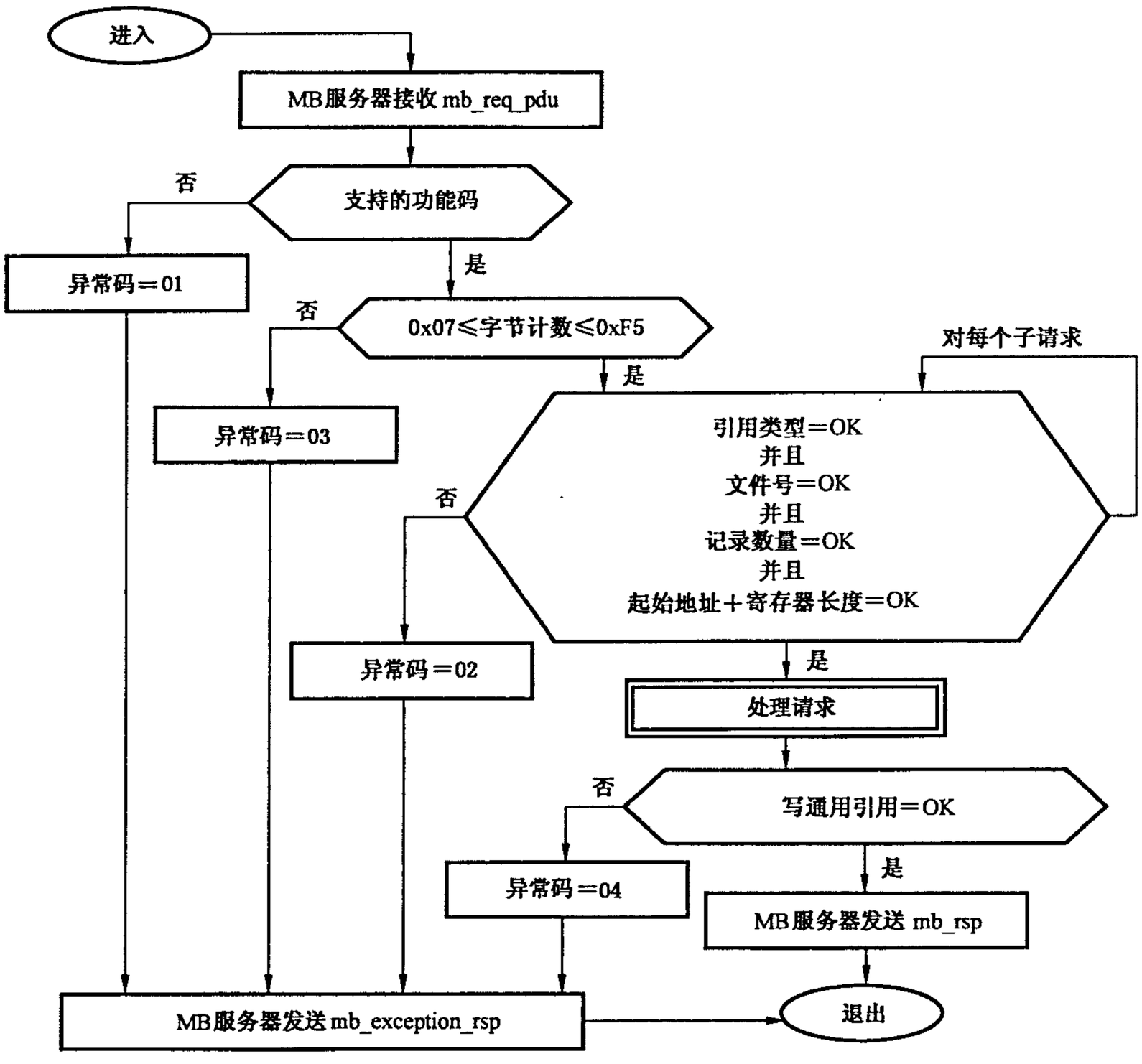


图 25　写文件记录状态图

表 63 是一个请求将一个引用组写入远程设备的示例。

组包括文件 4 中的 3 个寄存器，以寄存器 7 开始(地址 0007)。

表 63 一个引用组写入远程设备

请求		响应	
字段名	十六进制	字段名	十六进制
功能	15	功能	15
请求数据长度	0D	请求数据长度	0D
子请求 1,引用类型	06	子请求 1,引用类型	06
子请求 1,文件号 Hi	00	子请求 1,文件号 Hi	00
子请求 1,文件号 Lo	04	子请求 1,文件号 Lo	04
子请求 1,记录号 Hi	00	子请求 1,记录号 Hi	00
子请求 1,记录号 Lo	07	子请求 1,记录号 Lo	07
子请求 1,寄存器长度 Hi	00	子请求 1,寄存器长度 Hi	00
子请求 1,寄存器长度 Lo	03	子请求 1,寄存器长度 Lo	03
子请求 1,寄存器数据 Hi	06	子请求 1,寄存器数据 Hi	06
子请求 1,寄存器数据 Lo	AF	子请求 1,寄存器数据 Lo	AF
子请求 1,寄存器数据 Hi	04	子请求 1,寄存器数据 Hi	04
子请求 1,寄存器数据 Lo	BE	子请求 1,寄存器数据 Lo	BE
子请求 1,寄存器数据 Hi	10	子请求 1,寄存器数据 Hi	10
子请求 1,寄存器数据 Lo	0D	子请求 1,寄存器数据 Lo	0D

7.16 22(0x16) 屏蔽写寄存器

见图 26 和表 64～表 66。

该功能码用于通过利用“AND_Mask”、“OR_Mask”以及当前寄存器内容的组合来修改指定的保持寄存器的内容。这个功能可用来设置或清除寄存器中的不同位。

请求指定了被写入的保持寄存器、被用于“AND_Mask”的数据以及被用于“OR_Mask”的数据。从 0 开始寻址寄存器。因此，寄存器 1～16 被寻址为 0～15。

功能的算法为：

结果＝(当前内容 AND And_Mask) OR(Or_Mask AND(NOT And_Mask))

例如：

	十六进制	二进制
当前内容＝	12	0001 0010
And_Mask＝	F2	1111 0010
Or_Mask＝	25	0010 0101
NOT And_Mask＝	0D	0000 1101
结果＝	17	0001 0111

注 1：如果 Or_Mask 值为零，那么结果是当前内容和 And_Mask 的简单逻辑 AND(与)。如果 And_Mask 值为零，则结果等于 Or_Mask 值。

注 2：使用读保持寄存器功能(功能码 03)可读出寄存器的内容。但是，当控制器扫描它的用户逻辑程序时，随之可以改变寄存器的内容。

正常的响应是请求的复制。在寄存器被写入之后，返回响应。

表 64　屏蔽写寄存器请求

功能码	1 字节	0x16
引用地址	2 字节	0x0000～0xFFFF
And_Mask	2 字节	0x0000～0xFFFF
Or_Mask	2 字节	0x0000～0xFFFF

表 65　屏蔽写寄存器响应

功能码	1 字节	0x16
引用地址	2 字节	0x0000～0xFFFF
And_Mask	2 字节	0x0000～0xFFFF
Or_Mask	2 字节	0x0000～0xFFFF

表 66　屏蔽写寄存器错误响应

异常功能码	1 字节	0x96
异常码	1 字节	01 或 02 或 03 或 04

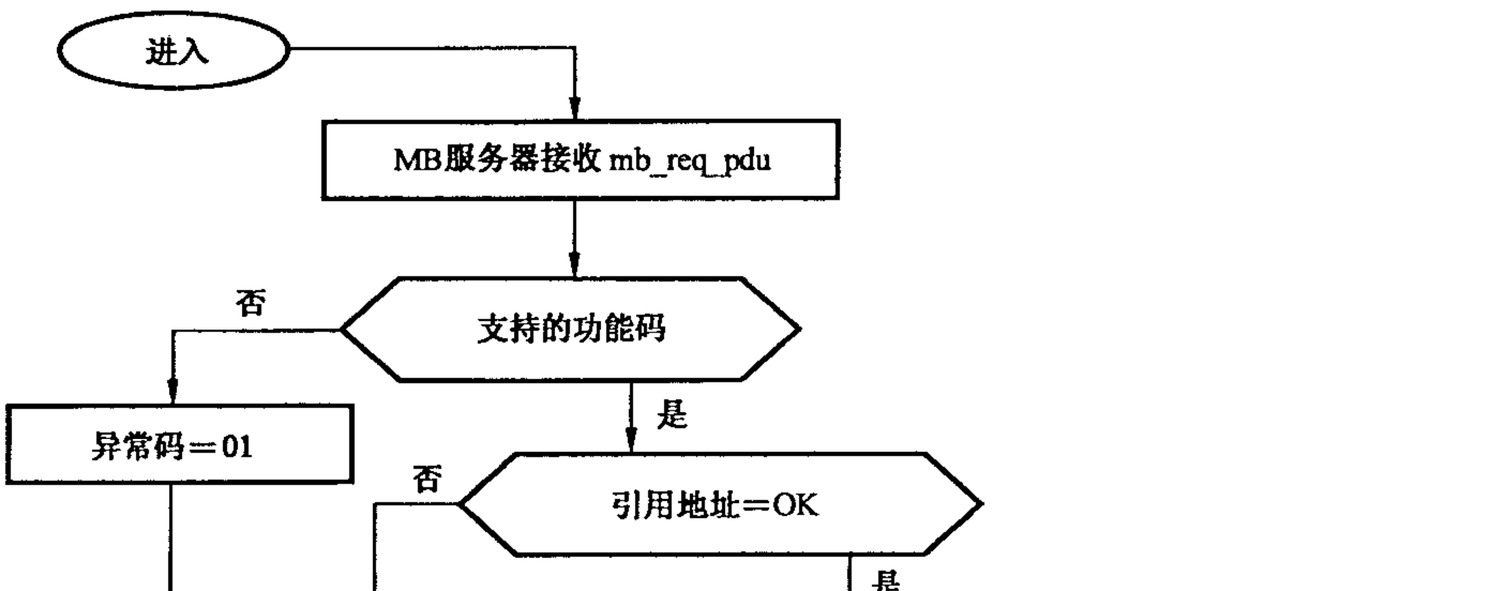


图 26　屏蔽写保持寄存器状态图

表 67 是一个利用上述屏蔽值在远程设备中对第 5 个寄存器的屏蔽写入的示例。

表 67　远程设备中对第 5 个寄存器的屏蔽写入

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	16	功能	16
引用地址 Hi	00	引用地址 Hi	00
引用地址 Lo	04	引用地址 Lo	04
And_Mask Hi	00	And_Mask Hi	00
And_Mask Lo	F2	And_Mask Lo	F2
Or_Mask Hi	00	Or_Mask Hi	00
Or_Mask Lo	25	Or_Mask Lo	25

7.17　23(0x17) 读/写多个寄存器

见图 27 和表 68～表 70。

这个功能码实现了在一个单独 Modbus 事务处理中一个读操作和一个写操作的组合。在读之前进行写操作。

从零开始寻址保持寄存器。因此，在 PDU 中保持寄存器 1～16 被寻址为 0～15。

请求指定了被读取的保持寄存器的起始地址和数目、被写入的保持寄存器的起始地址和数目以及数据。在写数据字段中，字节计数指定随后的字节数目。

正常的响应包含所读寄存器组中的数据。在读数据字段中，字节计数字段指定随后的字节数目。

表 68　读/写多个寄存器请求

功能码	1 字节	0x17
读起始地址	2 字节	0x0000～0xFFFF
读的数量	2 字节	0x0001～0x007D
写起始地址	2 字节	0x0000～0xFFFF
写的数量	2 字节	0x0001～0x0079
写字节计数	1 字节	$2\times N^a$
写寄存器值	$N^a\times 2$ 字节	
[a] N=写的数量。		

表 69　读/写多个寄存器响应

功能码	1 字节	0x17
字节计数	1 字节	$2\times N^a$
读寄存器值	$N^a\times 2$ 字节	…
[a] N=读的数量。		

表 70　读/写多个寄存器错误响应

异常功能码	1 字节	0x97
异常码	1 字节	01 或 02 或 03 或 04

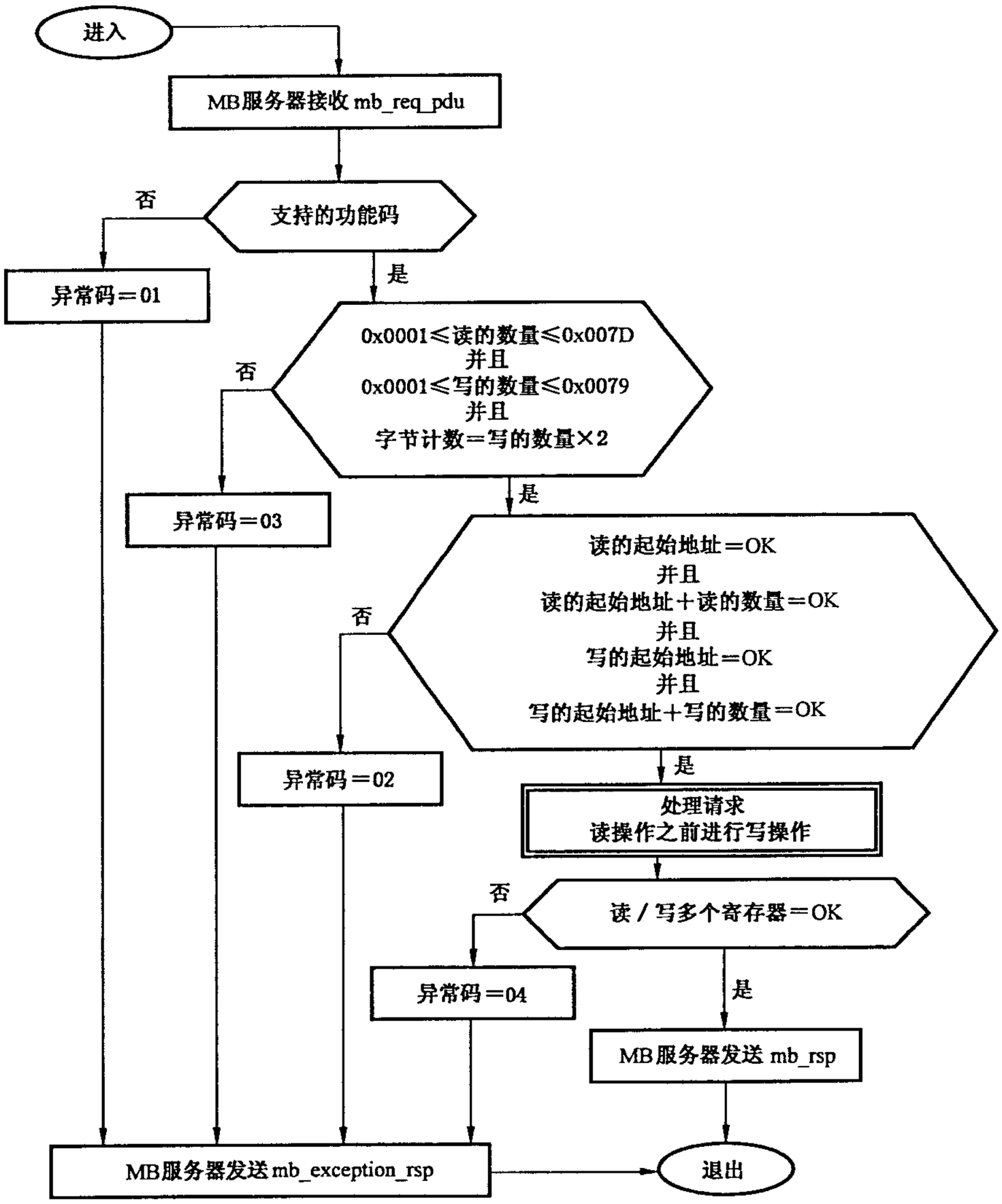


图 27　读/写多个寄存器状态图

表 71 是请求从第 4 个寄存器开始读 6 个寄存器并且从第 15 个寄存器开始写 3 个寄存器的示例。

表 71　从寄存器读和从寄存器写

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	17	功能	17
读起始地址 Hi	00	字节计数	0C
读起始地址 Lo	03	读寄存器值 Hi	00
读的数量 Hi	00	读寄存器值 Lo	FE
读的数量 Lo	06	读寄存器值 Hi	0A
写起始地址 Hi	00	读寄存器值 Lo	CD

表 71(续)

请　求		响　应	
字段名	十六进制	字段名	十六进制
写起始地址 Lo	0E	读寄存器值 Hi	00
写的数量 Hi	00	读寄存器值 Lo	01
写的数量 Lo	03	读寄存器值 Hi	00
写字节计数	06	读寄存器值 Lo	03
写寄存器值 Hi	00	读寄存器值 Hi	00
写寄存器值 Lo	FF	读寄存器值 Lo	0D
写寄存器值 Hi	00	读寄存器值 Hi	00
写寄存器值 Lo	FF	读寄存器值 Lo	FF
写寄存器值 Hi	00		
写寄存器值 Lo	FF		

7.18　24(0x18)读 FIFO 队列

见图 28 和表 72～表 74。

这个功能码允许读远程设备中的先入先出(FIFO)寄存器队列内容。这个功能码返回队列中寄存器的计数,随后为队列的数据。最多可以读 32 个寄存器:计数加上最多 31 个队列数据寄存器。首先返回队列计数寄存器,随后为队列数据寄存器。

这个功能码读队列内容,但不能清除队列内容。

在正常的响应中,字节计数表示随后的字节数量,包括队列计数字节和数值寄存器字节(不包括差错校验字段)。

队列计数是队列中数据寄存器的数量(不包括计数寄存器)。

如果队列计数大于 31,则返回一个含有异常码 03(非法数据值)的异常响应。

表 72　读 FIFO 队列请求

功能码	1 字节	0x18
FIFO 指针地址	2 字节	0x0000～0xFFFF

表 73　读 FIFO 队列响应

功能码	1 字节	0x18
字节计数	2 字节	…
FIFO 计数	2 字节	≤31
FIFO 数值计数	N[a]×2 字节	…
[a] N=FIFO 计数。		

表 74　读 FIFO 队列错误响应

异常功能码	1 字节	0x98
异常码	1 字节	01 或 02 或 03 或 04

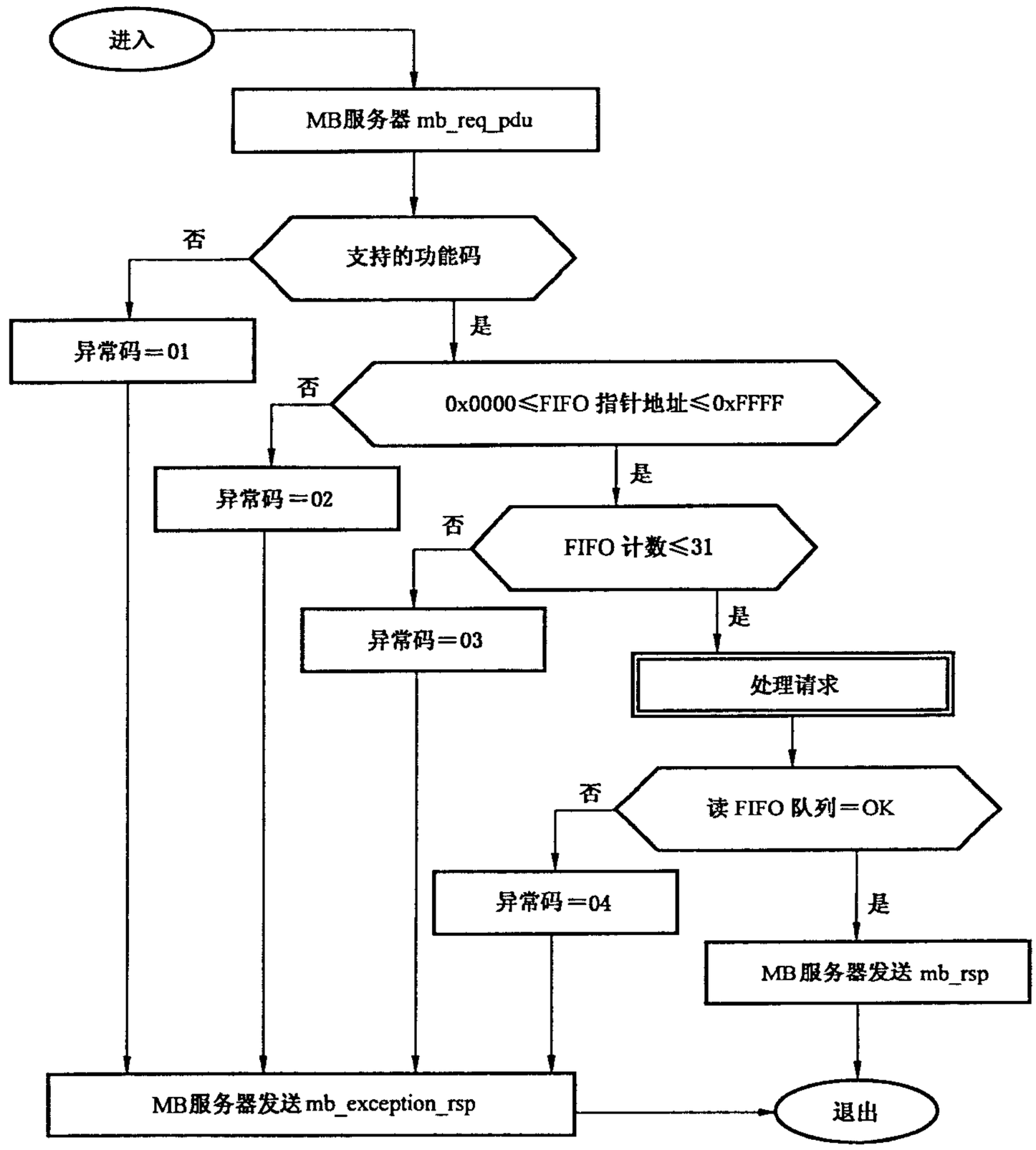


图 28　读 FIFO 队列状态图

表 75 是对远程设备读 FIFO 队列请求的示例。请求是读取以指针寄存器 1246(0x04DE)开始的队列。

表 75　对远程设备读 FIFO 队列请求

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	18	功能	18
FIFO 指针地址 Hi	04	字节计数 Hi	00
FIFO 指针地址 Lo	DE	字节计数 Lo	06
		FIFO 计数 Hi	00
		FIFO 计数 Lo	02
		FIFO 数值寄存器 Hi	01
		FIFO 数值寄存器 Lo	B8
		FIFO 数值寄存器 Hi	12
		FIFO 数值寄存器 Lo	84

在这个示例中，FIFO 指针寄存器(请求中的 1246)返回的内容为队列计数值 2。队列计数后面是两个数据寄存器。它们是 1247(十进制内容 440—0x01B8)和 1248(十进制内容 4740—0x1284)。

7.19　43(0x2B) 封装接口传输

见图 29 和表 76～表 78。

注：见附录 A。

功能码 43 及其用于设备标识的 MEI 类型 14 是本部分中当前可提供的两个封装接口传输方法之一。下面的功能码和 MEI 类型不属本部分的内容，这些功能码和 MEI 类型是专门保留的：43/0-12 和 43/15-255。

Modbus 封装接口(MEI)传输是一个在 Modbus PDU 内部将服务请求、方法调用和相关返回隧道化的机制。

MEI 传输的主要特征是封装属于已定义的接口组成部分的方法调用或服务请求，以及方法调用返回或服务响应。

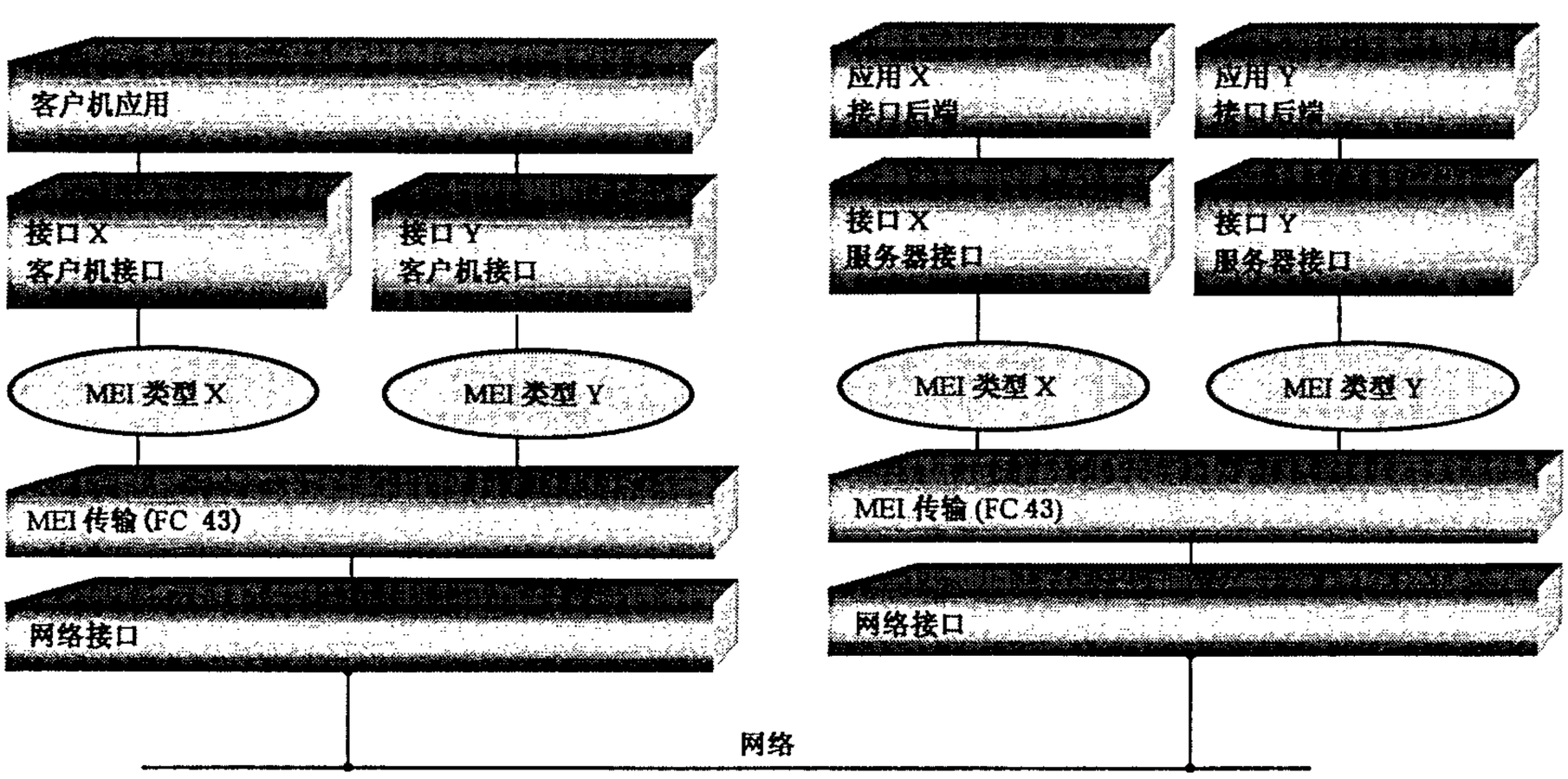


图 29　Modbus 封装接口传输

网络接口可以是用于发送 Modbus PDU 的任何通信栈，如 TCP/IP 或串行链路。

MEI 类型是 Modbus 指配的编号，因此，它是唯一的。根据附录 A，值 0～255 保留，MEI 类型 13 和 MEI 类型 14 除外。

MEI 传输实现使用 MEI 类型来分配对所指定接口的方法调用。

由于 MEI 传输服务是接口未知的，因此接口所需的任何特性或策略必须由该接口提供，例如：MEI 事务处理和 MEI 接口差错处理等。

表 76　封装接口传输请求

功能码	1 字节	0x2B
MEI 类型	1 字节	0x0D 或 0x0E
MEI 类型规定的数据	N 字节	…

表 77　封装接口传输响应

功能码	1 字节	0x2B
MEI 类型	1 字节	请求中的 MEI 类型的复制
MEI 类型规定的数据	N 字节	…

表 78　封装接口传输错误响应

异常功能码	1 字节	0xAB Fc 0x2B+0x80
异常码	1 字节	01 或 02 或 03 或 04

示例见读设备标识请求。

7.20 43/13(0x2B/0x0D) CANopen 通用引用请求和响应 PDU

CANopen 通用引用命令是服务的一种封装，这些服务被用来访问(读或写)CAN-Open 设备对象字典(CAN-Open Device Object Dictionary)的条目以及控制和监视 CANopen 系统和设备。

MEI 类型 13(0x0D)是 Modbus 分配给 CiA(CAN in Automation)，许可 CiA 用于 CANopen 通用引用的指定号码。

系统将工作于现有 Modbus 网络的限制内。因此，询问或者修改系统内对象字典的所需信息被映射到 Modbus 报文的格式中。在请求报文和响应报文中，PDU 被限制为 253 个字节。

注：关于 MEI 类型 13 的信息见附录 B。

7.21 43/14(0x2B/0x0E)读设备标识

见图 30 和表 79～表 82。

这个功能码允许读取与远程设备的物理和功能描述相关的标识和附加信息。

读设备标识接口由包含一组可寻址数据元素组成的地址空间构成。数据元素被称作对象，由对象 ID 识别它们。

接口由 3 类对象组成：

——基本设备标识。该类别的所有对象都是强制的：厂商名称、产品代码和修订版本号。

——常规设备标识。除基本数据对象以外，设备提供了附加的和可选择的标识以及数据对象描述。本部分定义了该类别的所有对象，但是它们的实现是可选的。

——扩展设备标识。除常规数据对象以外，设备提供了附加的和可选的标识以及关于物理设备本身的专用数据描述。所有这些数据都与设备有关。

表 79 读设备标识

对象 ID	对象名称/描述	类 型	强制的/可选的	类 别
0x00	厂商名称	ASCII 字符串	强制的	基本的
0x01	产品代码	ASCII 字符串	强制的	
0x02	主次版本号	ASCII 字符串	强制的	
0x03	厂商网址	ASCII 字符串	可选的	常规的
0x04	产品名称	ASCII 字符串	可选的	
0x05	型号名称	ASCII 字符串	可选的	
0x06	用户应用名称	ASCII 字符串	可选的	
0x07 … 0x7F	保留		可选的	
0x80 … 0xFF	可选择地定义专用对象范围(0x80～0xFF)与产品有关	与设备相关	可选的	扩展的

表 80 读设备标识请求

功能码	1 字节	0x2B
MEI 类型	1 字节	0x0E
读设备 ID 码	1 字节	01/02/03/04
对象 ID	1 字节	0x00～0xFF

表 81　读设备标识响应

功能码	1 字节	0x2B
MEI 类型	1 字节	0x0E
读设备 ID 码	1 字节	01/02/03/04
一致性等级	1 字节	0x01 或 0x02 或 0x03 或 0x81 或 0x82 或 0x83
接续标识	1 字节	00/FF
下一个对象 ID	1 字节	对象 ID 号
对象数量	1 字节	…
列表		
对象 ID	1 字节	…
对象长度	1 字节	…
对象值	对象长度	与对象 ID 有关

表 82　读设备标识错误响应

功能码	1 字节	0xAB： Fc0x2B+0x80
异常码	1 字节	01 或 02 或 03 或 04

请求参数描述：

Modbus 封装接口指配的编号 14 表示读标识请求。

参数“读设备 ID 码”允许定义四种访问类型：

01：请求获得基本设备标识(流访问)

02：请求获得常规设备标识(流访问)

03：请求获得扩展设备标识(流访问)

04：请求获得特定标识对象(单个访问)

如果读设备 ID 码是无效的，则在响应中返回异常码 03。

在设备标识不适用单独响应的情况下，需要进行多个事务处理(请求/响应)。对象 ID 字节给出了所要获得的第一个对象的标识。对于第一个事务处理来说，客户机必须设置对象 ID 为 0，以便获得设备标识数据的起始信息。对于随后事务处理来说，客户机必须将对象 ID 设置为服务器在此前响应中返回的值。

注：对象是不可分割的，因此任何对象的大小与事务处理响应大小一致。

如果对象 ID 不匹配任何已知对象，那么服务器的响应如同指向对象 0(从头开始)。

在单个访问的情况下：读设备 ID 码 04，请求中的对象 ID 给出了所要获得的对象标识。

如果在单个访问中对象 ID 不匹配任何已知对象，那么服务器返回一个异常码＝02(非法数据地址)的异常响应。

如果要求服务器设备高于其一致性等级的描述，则服务器设备必须根据其对应的实际等级响应。

响应参数描述：

功能码：　功能码 43(十进制)0x2B(十六进制)

MEI 类型：　为设备标识接口指配的编号 MEI 类型 14(0x0E)

读设备 ID 码：　与请求读设备 ID 码相同：01、02、03 或 04

一致性等级：　设备的标识一致性等级和支持的访问类型

0X01：基本标识(仅流访问)

0X02：常规标识(仅流访问)

0X03：扩展标识(仅流访问)

0X81：基本标识(流访问和单个访问)

	0X82:常规标识(流访问和单个访问)
	0X83:扩展标识(流访问和单个访问)
接续标识:	在读设备 ID 码 01、02 或 03(流访问)的情况下,
	在标识数据不适用单独响应的情况下,需要进行多个事务处理(请求/响应)。
	00:没有后续对象
	FF:有后续标识对象,并且需要进一步的 Modbus 事务处理
	在读设备 ID 码 04(单个访问)的情况下,
	必须设置这个字段为 00。
下一个对象 ID:	如果“接续标识=FF”,那么请求下一个对象 ID
	如果“接续标识=00”,那么必须设置为 00(无用的)
对象数目	在响应中返回的标识对象的数目(对于单个访问,对象数目=1)
对象 0. ID	PDU 中返回的第一个对象的标识(流访问)或请求的对象的标识(单个访问)
对象 0. 长度	第一个对象的字节长度
对象 0. 值	第一个对象的值(对象 0. 长度字节)
...	
对象 N. ID	最后对象的标识(在响应中)
对象 N. 长度	最后对象的字节长度
对象 N. 值	最后对象的值(对象 N. 长度字节)

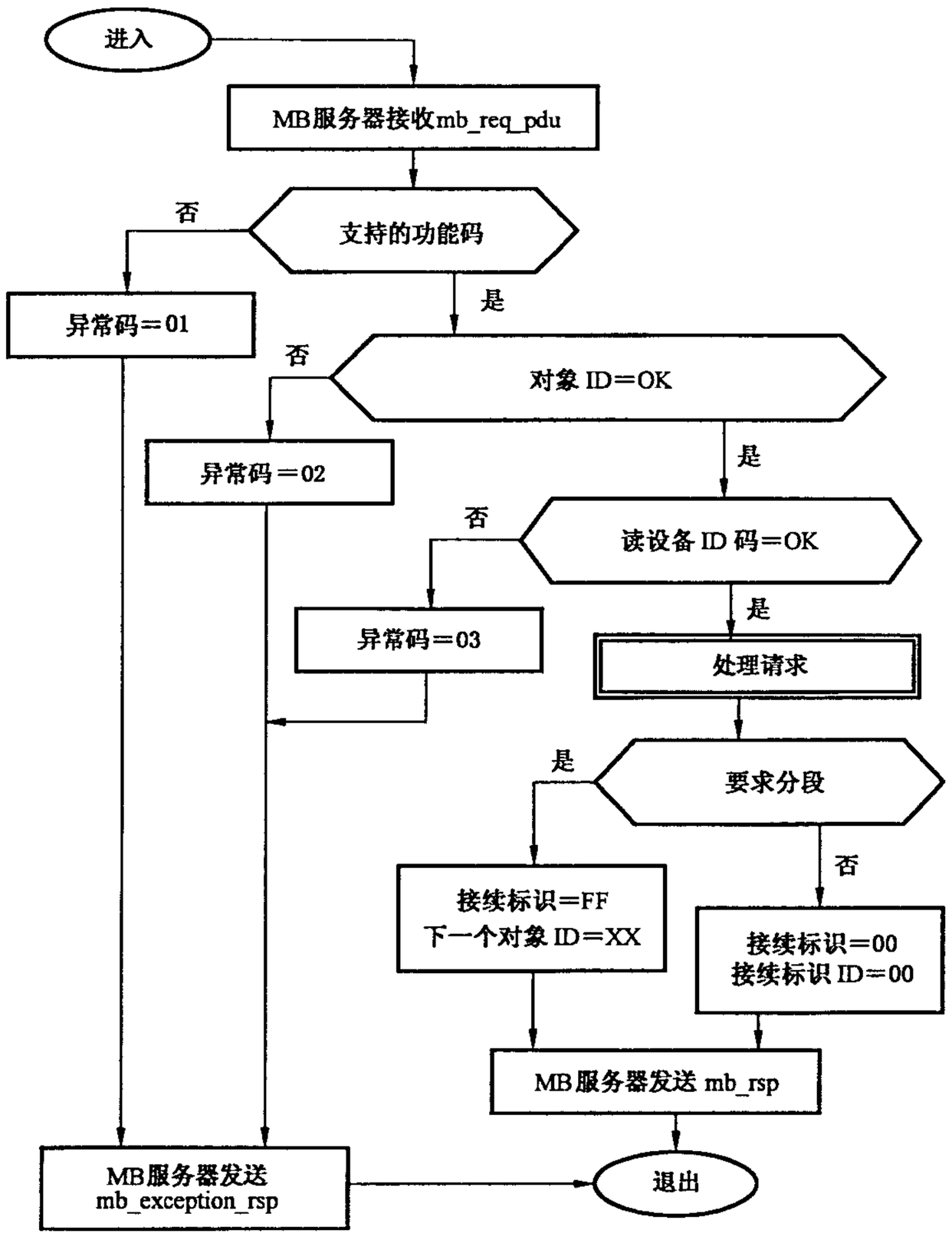


图 30 读设备标识状态图

“基本设备标识”的读设备标识请求的示例见表 83。在这个示例中，所有信息在一个响应 PDU 中发送。

表 83　读“基本设备标识”请求

请　求		响　应	
字段名	值	字段名	值
功能	2B	功能	2B
MEI 类型	0E	MEI 类型	0E
读设备 ID 码	01	读设备 ID 码	01
对象 ID	00	一致性等级	01
		接续标识	00
		下一个对象 ID	00
		对象数量	03
		对象 ID	00
		对象长度	16
		对象值	“Company identification”
		对象 ID	01
		对象长度	0D
		对象值	“product code XX”
		对象 ID	02
		对象长度	05
		对象值	“V2.11”

如果一个设备需要多个事务处理发送响应，那么启动随后事务处理。

第一个事务处理见表 84。

表 84　事务处理 1

请　求		响　应	
字段名	值	字段名	值
功能	2B	功能	2B
MEI 类型	0E	MEI 类型	0E
读设备 ID 码	01	读设备 ID 码	01
对象 ID	00	一致性等级	01
		接续标识	FF
		下一个对象 ID	02
		对象数量	03
		对象 ID	00
		对象长度	16
		对象值	“Company identification”
		对象 ID	01
		对象长度	1C
		对象值	“Product code ××××××××” (0x1C 个字符)

第二个事务处理见表 85。

表 85　事务处理 2

请　求		响　应	
字段名	值	字段名	值
功能	2B	功能	2B
MEI 类型	0E	MEI 类型	0E
读设备 ID 码	01	读设备 ID 码	01
对象 ID	02	一致性等级	01
		接续标识	00
		下一个对象 ID	00
		对象数量	03
		对象 ID	02
		对象长度	05
		对象值	“V2.11”

8　Modbus 异常响应

当客户机设备向服务器设备发送请求时，客户机希望得到一个正常的响应。主站的询问可能导致下列四种事件之一：

——如果服务器设备接收到无通信错误的请求，并且可以正常地处理询问，那么服务器设备将返回一个正常响应。

——如果由于通信错误服务器没有接收到请求，那么不能返回响应。客户机程序将按超时处理。

——如果服务器接收到请求，但是检测到一个通信错误（奇偶校验、LRC、CRC...），那么不能返回响应。客户机程序将按超时处理。

——如果服务器接收到无通信错误的请求，但不能处理这个请求（例如，如果请求读一个不存在的输出或寄存器），那么服务器将返回一个异常响应，通知客户机出错误的原因。

异常响应报文有两个与正常响应不同的字段：

功能码字段：在正常响应中，服务器在响应的功能码字段复制原始请求的功能码。所有功能码的 MSB 都为 0（它们的值都低于十六进制 80）。在异常响应中，服务器设置功能码的 MSB 为 1。这使得异常响应中的功能码值比正常响应中的功能码值高十六进制 80。

通过设置功能码的 MSB，客户机的应用程序能够识别异常响应，并且能够检测异常码的数据字段。

数据字段：在正常的响应中，服务器可以在数据字段中返回数据或统计值（请求中要求的任何信息）。在异常响应中，服务器在数据字段中返回异常码。这说明了服务器产生异常的原因。

表 86 为客户机请求和服务器异常响应的示例。

表 86　客户机请求和服务器异常响应

请　求		响　应	
字段名	十六进制	字段名	十六进制
功能	01	功能	81
起始地址 Hi	04	异常码	02
起始地址 Lo	A1		
输出数量 Hi	00		
输出数量 Lo	01		

在这个示例中，客户机向服务器设备发出一个请求。功能码（01）用于读输出状态操作。它请求地

址为 1185(十六进制 04A1)的输出状态。根据输出字段(0001) 所指定的数量,只读出一个输出。

如果在服务器设备中不存在该输出地址,那么服务器将返回带有异常码(02)的异常响应。这就说明客户机指定的是非法的从站数据地址。

表 87 列出了异常码。

表 87 异常码

Modbus 异常码		
代码	名　　称	含　　义
01	非法功能	对于服务器(或从站)来说,询问中接收到的功能码是不允许的操作。这也许是因为功能码仅适用于新设备而在被选单元中没有实现;还可能表示服务器(或从站)在错误状态中处理这种请求,例如:因为它是未配置的,并且正被要求返回寄存器值
02	非法数据地址	对于服务器(或从站)来说,询问中接收到的数据地址是不允许的地址。特别是,寄存器编号和传输长度的组合是无效的。对于带有 100 个寄存器的控制器来说,PDU 赋值第一个寄存器为 0,最后一个为 99。如果起始寄存器编号 96 和寄存器数量为 4 的请求被处理,那么这个请求会成功操作于寄存器 96、97、98 和 99;而如果起始寄存器编号 96 和寄存器数量为 5 的请求被处理,那么将产生异常码 02"非法数据地址",因为它试图作用于寄存器 96、97、98、99 和 100,而寄存器 100 是不存在的
03	非法数据值	对于服务器(或从站)来说,询问数据字段中包含的是不允许的值。它表示组合请求中剩余部分结构方面的错误,例如:隐含长度不正确。它决不表示寄存器中被提交存储的数据项有一个应用程序期望之外的值,因为 Modbus 协议并不知道任何特殊寄存器的任何特殊值的具体含义
04	从站设备故障	当服务器(或从站)正在试图执行所请求的操作时,产生不可恢复的差错
05	确认	与编程命令一起使用。 服务器(或从站)已经接受请求,并且正在进行处理,但是需要较长的处理时间。返回这个响应以防止在客户机(或主站)中发生超时错误。客户机(或主站)可以继续发送轮询程序完成报文来确定是否完成处理
06	从站设备忙	与编程命令一起使用。 服务器(或从站)正在处理较长时间的程序命令。当服务器(或从站)空闲时,客户机(或主站)应该稍后重新传送报文
08	存储奇偶性差错	与功能码 20、21 和引用类型 6 一起使用,以指示扩展文件区不能通过一致性校验; 服务器(或从站)试图读记录文件,但是在存储器中发现一个奇偶校验错误。客户机(或主站)可以重新发送请求,但可能需要在服务器(或从站)设备上提供这种服务
0A	网关路径不可用	与网关一起使用。它表示网关不能为处理请求分配输入端口至输出端口的内部通信路径。通常表示网关配置错误或过载
0B	网关目标设备响应失败	与网关一起使用。它表示没有从目标设备中获得响应。通常表示设备未在网络中

附 录 A
（资料性附录）
Modbus 保留的功能码、子码及 MEI 类型

下列功能码和子码不属本部分的内容，这些功能码和子码是特殊保留的。格式是功能码/子码，或仅有功能码，其所有子码(0～255)均被保留：8/19、8/21～65 535、9、10、13、14、41、42、90、91、125、126和 127。

功能码 43 中用于设备标识的 MEI 类型 14，以及用于 CANopen 通用引用请求和响应 PDU 的 MEI 类型 13 都是本部分中目前可用的封装接口传输。

下列功能码和 MEI 类型不属本部分的内容，这些功能码和 MEI 类型是特殊保留的：43/0～12 和 43/15～255。在本部分中，不支持具有与封装接口传输相同或类似结果的用户定义的功能码。

附 录 B
（资料性附录）
CANopen 通用引用命令

关于功能码 43 和 MEI 类型 13 的用法，请访问 MODBUS-IDA 网站或 CiA(CAN in Automation)网站。

参 考 文 献

[1] ANSI/EIA/TIA-232-E:1997 interface between data terminal equipment and data circuit-terminating equipment employing serial binary data interchang.

[2] ANSI/EIA/TIA-485-A:1998 electrical characteristics of generators and recievers for use in balanced digital multipoint systems.

[3] MODBUS-IDA. org Modbus application protocol specification.

中 华 人 民 共 和 国
国 家 标 准
基于 Modbus 协议的工业自动化网络规范
第 1 部分：Modbus 应用协议
GB/T 19582.1—2008
*
中国标准出版社出版发行
北京复兴门外三里河北街 16 号
邮政编码：100045
网址 www.spc.net.cn
电话：68523946 68517548
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 3.5 字数 97 千字
2008 年 5 月第一版 2008 年 5 月第一次印刷
*
书号：155066·1-31328

GB/T 19582.1-2008