

# 第一部分 卡片基本规范

## 目 次

1 主要内容	3
2 参考资料	3
3 定义	3
4 缩略语和符号表示	
5 物理特性、卡上信息记录方法和物理接口要求	
6 电特性	
7 卡的操作过程	
8 复位应答	

## 1 主要内容

本规范的这一部分规定了 ID-1 型带触点集成电路卡的基本技术要求，主要包括以下内容：

——物理特性、记录方法、物理接口要求，主要定义了该类卡的基本物理特性。

——电气信号和传输协议，规定了该类卡和终端间的电源、电气信号协议和信息交换协议，涉及卡的信号频率、电压值、电流值、校验、操作规程和传输与通信协议。

本部分适用于中国范围内发行或应用的 IC 卡，其使用对象主要是与 IC 卡应用相关的卡片设计、制造、管理、发行以及应用系统的研制、开发、集成和维护等部门或单位。

## 2 参考资料

- GB/T 14916 – 1994 识别卡- 物理特性
- GB/T 16649.1 – 1996 识别卡- 带触点的集成电路卡- 第1部分: 物理特性
- GB/T 16649.2 – 1996 识别卡- 带触点的集成电路卡- 第2部分: 触点尺寸和位置
- GB/T 16649.3 – 1996 识别卡- 带触点的集成电路卡- 第3部分: 电信号和传输协议
- ISO/IEC 7816 – 4: 1995 识别卡- 带触点的集成电路卡- 第4部分: 交换用行业间指令
- ISO/IEC 7816 – 5: 1995 识别卡- 带触点的集成电路卡- 第5部分: 应用标识符的编号体系和注册程序。
- 《集成电路卡注册管理办法》

## 3 定义

### 3.1 识别卡 identification card

一种可识别其持卡人和发卡方的卡，卡上载有其预期应用及有关交易所要求输入的数据。

### 3.2 集成电路(IC) Integrated circuit(s)

将处理和/或存储功能集成在一个芯片上的电子器件。

### 3.3 集成电路卡(IC卡) integrated circuit(s) card (IC card)

内部封装一个或多个集成电路的ID-1型卡(如ISO 7810、ISO 7811第1至第5部分、ISO 7812和ISO 7813中描述的)。

### 3.4 触点 contact

在集成电路和外部接口设备之间保持电流连续性的导电元件。

### 3.5 凸印 embossing

使字符从卡的正面显著地凸起。

### 3.6 接口设备 Interface device

在操作中同IC卡电连接的终端、通信设备或机器。

### 3.7 状态H State H

高状态逻辑电平。

### 3.8 状态L State L

低状态逻辑电平。

### 3.9 状态Z State Z

标记(如ISO 1177中定义)。

### 3.10 状态A State A

空位(如ISO 1177中定义).

### 3.11 'XY'

十六进制记数法,等于相对于基数16的XY.

### 3.12 块 block

由起始域、信息域和终止域组成的连续字符. 其中起始域和终止域是必需的,信息域是可选的.

### 3.13 目的节点地址 destination node address(DNA)

节点地址子域(DNA)的一部分,用于标识一个块的未来接收者.

### 3.14 终止域 epilogue field

块的最后一个域. 包括差错检测编码(EDC)字节.

### 3.15 差错检测编码 error detection code(EDC)

差错检测的方法之一,检测起始域和信息域的所有字符. 它在终止域中被传送.

### 3.16 域 field

定义为起始域、信息域或终止域.

### 3.17 信息块 information block(I-block)

主要用于传输应用层信息的块.

### 3.18 信息域 information field(INF)

含有数据(一般为应用数据)的块中的一个域.

### 3.19 长度 length(LEN)

起始域中的一个子域. 它指出在块的信息域中被传输的字节个数.

### 3.20 节点地址 node address (NAD)

起始域中的一个子域. 它指明某个块的目的和源节点地址以及VPP状态控制.

### 3.21 起始域 prologue field

块的第1个域. 它包含节点地址(NAD)子域、协议控制字节(PCB)和长度(LEN).

### 3.22 协议控制字节 protocol control byte(PCB)

起始域中的一个子域. 它包含传输控制信息.

### 3.23 接收准备块 receive ready block (R-block)

一个包含肯定或否定确认信息的块. 它包含预期的信息块(I-block)的块数.

### 3.24 源节点地址 source node address(SAD)

节点地址(NAD)子域的一部分,用于指定块的发送方.

### 3.25 子域 subfield

一个域中的一种功能成分.

### 3.26 管理块 supervisory block(S-block)

包含传输控制信息的块.

### 3.27 传输控制 transmission control

控制接口设备(IFD)和集成电路卡(ICC)之间进行数据传输. 它包含VPP状态控制、块顺序传输控制、同步以及传输差错的校正.

### 3.28 复位应答文件 Answer-to-Reset file

指示卡操作特性的基本文件.

### 3.29 指令应答对 command-响应 pair

两种信报的组合,一个指令跟着一个响应.

- 3.30 数据单元 data unit  
唯一被引用的最小字节集合。
- 3.31 数据元 data element  
在接口呈现的用于定义名称,描述逻辑文本、格式和编码的信息项。
- 3.32 数据对象 data object  
在接口呈现的涉及标签、长度和值(例如:数据元)的信息。在本规范中数据对象指BER\_TLV,COMPACT\_TLV和SIMPLE\_TLV数据对象。
- 3.33 文件控制参数 file control parameters  
指一个文件在逻辑上、结构上的和安全上的属性。
- 3.34 文件标识符 file identifier  
用于文件寻址的2个字节。
- 3.35 文件管理数据 file management data  
除文件控制参数(例如:终止数据,应用标签)以外的任何文件信息。
- 3.36 内部基本文件 internal elementary file  
用于存储由卡解释的数据的基本文件。
- 3.37 主文件 master file  
强制性和唯一被指定的文件,它代表了根目录下的文件结构。
- 3.38 信报 message  
由接口设备传向卡或由卡传向接口设备的字节串,它不包括面向传输的字符。
- 3.39 父文件 parent file  
在等类分类中,仅优先一个给定文件的专用文件。
- 3.40 口令 password  
一个应用可能需要的代表卡的用户的的数据。
- 3.41 路径 path  
没有定界的文件标识符的连接。如果路径以主文件的识别符开始,它就是一个路径。
- 3.42 提供者 provider  
获得或已经获得权力来创建卡中专用文件的机构。
- 3.43 记录 record  
被卡作为一个整体来处理的字节串,并可通过记录号或记录标识符来引用。
- 3.44 记录标识符 record identifier  
一种与记录相关的值,它用于引用该记录。在一个基本文件中,少数几个记录可以有相同的记录标识符。
- 3.45 记录号 record number  
分配给每一个记录的顺序号,它唯一地标识一个基本文件中的记录。
- 3.46 基本工作文件 working elementary file  
用于存储不由卡来解释的数据的基本文件。
- 3.47 冷复位 cold reset  
激活后的第一次复位。
- 3.48 热复位  
除冷复位以外的任何复位。

#### 4 缩略语

以下缩略语适用于本部分。

AAC	应用鉴定密码
AC	访问条件
AC	应用加密
ACK	确认
AID	应用标识
APDU	应用协议数据单元
ARQC	授权请求密码
ASC	应用专用指令集
ATC	应用交易序号
ATR	复位应答
BCD	二进制编码的十进制
VIH	高电平输入电压
VIL	低电平输入电压
VCC	VCC上的电源电压
VPP	VPP上的编程电压
VOH	高电平输出电压
VOL	低电平输出电压
tr	信号幅度在10%和90%之间的上升时间
tF	信号幅度在90%和10%之间的下降时间
I IH	高电平输入电流
VIL	低电平输入电流
ICC	VCC上的电源电流
IPP	VPP上的编程电流
IOH	高电平输出电流
IOL	低电平输出电流
CIN	输入电容
COU	输出电容
BGT	块保护时间
BWI	块等待时间整数
BWT	块等待时间
CRC	循环冗余检验
CWI	字符等待时间整数
CWT	字符等待时间
DAD	目的节点地址
EDC	差错检测编码
I-block	信息块
IFD	接口设备
IFS	信息域尺寸
IFSC	卡信息域尺寸
IFSD	接口设备信息域尺寸

IFSI	整型信息域尺寸
INF	信息域
LEN	长度
LRC	纵向冗余检验
NAD	节点地址
OSI	开放系统互连
PCB	协议控制字节
R-block	接收准备块
R	接收准备
SAD	源节点地址
S-block	管理块
WTX	扩展等待时间
XOR	异或
APDU	应用协议数据单元
ATR	复位应答
CLA	类字节
DIR	目录
DF	专用文件
EF	基本文件
FCI	文件控制信息
FCP	文件控制参数
FMD	文件管理数据
INS	指令字节
MF	主文件
P1-P2	参数字节
PTS	协议类型选择
RFU	留待将来使用
SM	安全报文处理
SW1-SW2	状态字节
TLV	标记、长度、值

## 5 物理特性、附加信息记录方法和接口要求

符合本规范的集成电路卡应遵守ISO/IEC 7816系列标准中的有关规定。

### 5.1 物理特性

#### 5.1.1 IC卡的一般特性

ISO 7810中规定的各类识别卡的物理特性适用于IC卡, ISO 7813中描述的金融交易卡的全部尺寸要求也应适用于这类卡。

注:

- 1、 ISO 7810中规定的卡的厚度适用于带触点、无凸印的卡。
- 2、 关于抗化学性(见ISO 7810的6.1.4条), 发卡方应注意污染会导致保存在磁条或集成电路中的信息无效。

##### 5.1.1.1 变形特性

卡应有这样的特性, 即其在正常使用期间的变形(弯曲而无折痕)能被记录或印刷设备在操作过程中弹性地变平, 而不损坏卡的功能。

##### 5.1.1.2 可燃性

当需要时, 耐燃性可以在与识别卡各种应用有关的标准中规定。

##### 5.1.1.3 有毒性

卡在正常使用过程中不应存在毒性危害。

##### 5.1.1.4 耐化学性

卡应经受住正常处理和使用时的化学影响。

##### 5.1.1.5 温度稳定性

在环境温度 $-35^{\circ}\text{C}$  ~  $+50^{\circ}\text{C}$ 之间卡应保持结构上可靠和可用。

注: 指定的环境温度不是指卡的温度, 而是指使用卡时的环境温度。

##### 5.1.1.6 湿度

在相对空气湿度5% ~ 95%之间、最大湿球温度 $25^{\circ}\text{C}$ 时, 卡应能可靠使用。

##### 5.1.1.7 光

在正常使用期间卡和其上已印的内容应能防止由于光照而产生变化。

##### 5.1.1.8 带凸印卡

对于带凸印卡, 应特别注意影响其适用性的材料特性, 尤其是在压印机中操作时, 其凸起部分应有耐压碎和耐压扁的能力。

##### 5.1.1.9 带有磁条的卡

下列要求适用于带有磁条的卡。

###### 5.1.1.9.1 卡的材料

卡的材料不应包含有可能位渗入或改变磁性材料的成份, 以致于卡在正常使用期间, 其材料可能变得不能满足一系列关于识别卡标准所规定的特性。

###### 5.1.1.9.2 ID-1型卡的翘曲

把即将发行的凸印/编码卡的正面朝上放到一个平面上, 从该平面到卡正面的任何非凸起部分的最大距离不应大于2mm。在与磁条相对地正面均匀的施加2.2N的力, 应出现离该平面不大于0.08mm的整条压线。

###### 5.1.1.9.3 表面畸变



在B区减A区处(见ISO 7810中的图1)不应有表面畸变、不规则或隆起,否则在卡的背面会妨碍磁头,在卡的下面妨碍磁编码和读出。

如果隆起处是签名条,无论它位于卡的正面或者背面,均与磁条宽度无关,但应满足下列要求:

a. 如果签名长度不小于79.88mm且从卡的右侧边不超过2.92mm处开始放置,则隆起部分与卡的顶边距离应大于16.76mm;

b. 对于其它情况,隆起部分与卡的顶边距离应大于19.05mm。在凸印区(见ISO 7810中的图1, C区减D区处)的隆起部分不应超过0.51mm。

边缘毛刺不应超过0.08mm。

在卡正面或背面所有其余部分的隆起部分不应超过0.25mm。

注:签名条在某些阅读或编码设备操作中可能被划伤或污损。

#### 5.1.1.9.4 污染

卡的材料和附加到卡上的任何材料不应污染读磁条、编码或读卡设备。

#### 5.1.2 IC卡的附加特性

本规范规定的IC卡应遵守ISO/IEC 7816-1的第4.2条的规定。

##### 5.1.2.1 紫外线

超过周围紫外线水平的防护应是卡制造商的责任。

##### 5.1.2.2 X—射线

卡的任何一面曝光0.1Gy剂量,相当于70—140KV的中等能量X—射线(每年的累积剂量),应不引起卡的失效。

##### 5.1.2.3 触点的表面断面

所有的触点及其附近的卡的表面之间在水平上的误差应小于0.10mm。ISO 7810第6.3.条中规定的保护区域应扩大到图中B和C之间的区域(见ISO 7810中的图1)。

##### 5.1.2.4 机械强度(卡和触点)

卡应能抵抗对其表面及其任何组成成分的损害,并在正常使用、保存和处理过程中保持完好。

每个触点表面和触点区域(整个导电表面)在相当于对直径1毫米的钢球施加1.5N的工作压力下不应被破坏。

##### 5.1.2.5 (触点的)电阻

卡连接部件的触点电阻可通过测试卡来确定和测量。该测试卡在内部的触点之间短路。在加50  $\mu$ A至300mA之间的任何直流电流时,任何两列触点(两触点串联)之间测得的电阻应小于0.5欧姆。对于一个峰值为10mA频率为4MHz的交流电流来说,阻抗应使跨过该阻抗的电压保持低于10mV。

##### 5.1.2.6 电磁干扰(磁条和集成电路之间)

如果卡带有磁条,磁条在读、写或抹磁后,IC卡应不被损坏、失效或改变。反之,集成电路的读、写也不应引起磁条失效或其读、写和处理机制的失效。

##### 5.1.2.7 电磁场

卡暴露在79.500A. r/m的磁场中应不造成集成电路的失效,测试应该用指定值的静磁场进行。

警告:磁场将会抹去磁条上的内容(如果用磁条)。

##### 5.1.2.8 静电

带静电的人在正常情况下,应不损坏集成电路。

在任意触点和地之间, 1500V的静电由一个100 pF的电容经过1500欧姆的电阻放电, 卡暴露其中时, 其功能不应降低。

#### 5.1.2.9 散热

卡中集成电路的散热应不大于2.5W。

警告: 无论在什么样的环境条件下应当注意卡的表面温度不能超过50℃。

### 5.2 附加信息记录的方法

#### 5.2.1 凸印〔当IC卡带有凸印时〕

带凸印的ID-1型IC卡应符合ISO 7811-1和ISO 7811-3的规定要求。

#### 5.2.2 磁条〔当IC卡带有磁条时〕

带磁条的IC卡应符合ISO 7811-2、ISO 7811-4、ISO 7811-5和ISO 7813的规定要求。

### 5.3 IC卡的尺寸和触点位置

#### 5.3.1 IC卡的

##### 尺寸

IC卡的外形尺寸应符合ISO/IEC 7810的有关规定。

卡的类型	宽度	高度	厚度
ID-1型	85.60mm	53.98mm	0.76mm

#### 5.3.2 触点尺寸和位置

本部分不定义每一个触点包含的传导区表面和形状。每个触点都应有一个不小于图1中规定尺寸的最小矩形表面区域。除了要求每个触点和其它触点应该电隔离之外, 本部分不规定触点的最大形状或尺寸。

单位: mm

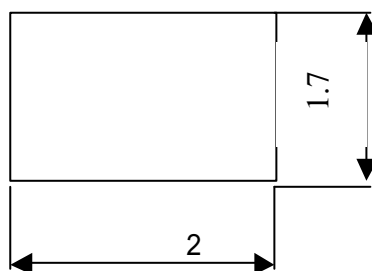


图1 触点的最小尺寸

本部分定义了C1 - C8共8个触点。

触点按图2所示定位。触点应被定位在卡的正面, 其尺寸都以卡表面的左边缘和上边缘为基准。

单位: mm

v.10

第1部分: 卡片规范

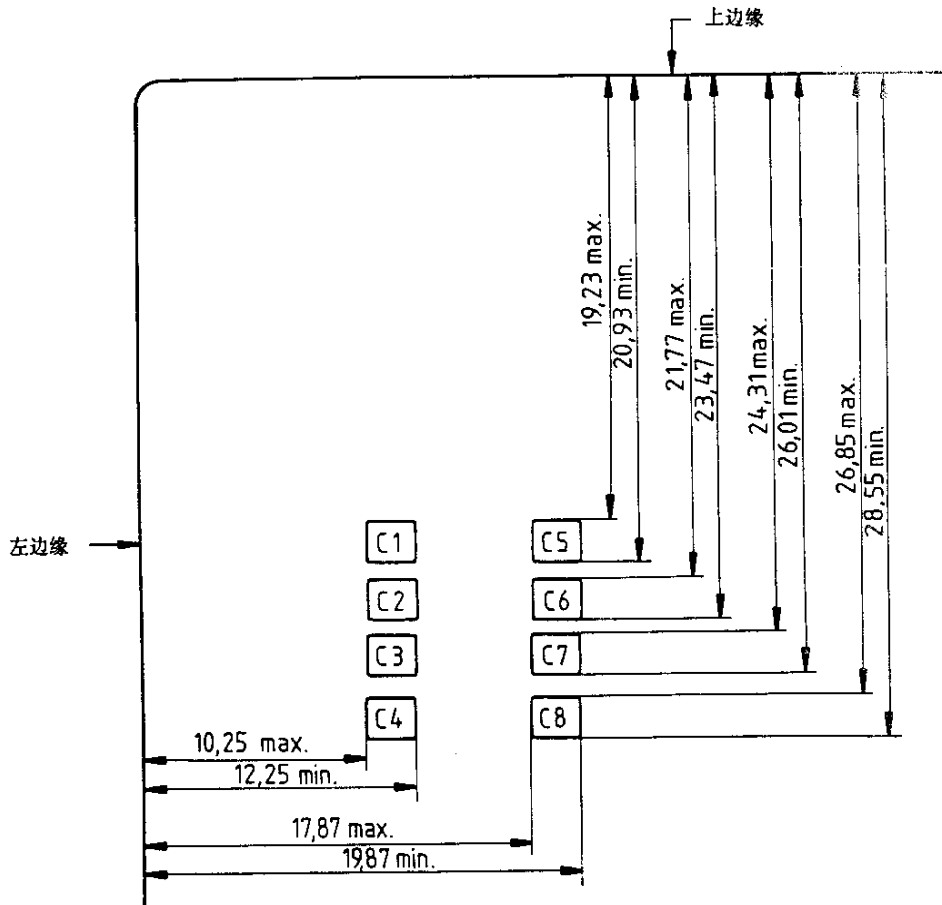


图2 触点位置

## 6 电气特性

### 6.1 总则

#### 6.1.1 电路

本规范规定的IC卡触点分配按照表1中的规定分配。

表1 触点的分配

触点号	分配	触点号	分配
C1	电源电压(VCC)	C5	地 (GND)
C2	复位(RST)	C6	编程电压 (VPP)
C3	时钟(CLK)	C7	输入/输出 (I/O)
C4	保留待未来使用	C8	保留待未来使用

其中:

GND 地, 基准电压.

VCC 电源输入.

I/O 串行数据的输入/输出.

CLK 时钟信号输入.

RST 复位信号输入.

VPP 编程电压输入, 由卡选用.

#### 6.1.2 缩略语

见第4章.

### 6.2 操作条件

#### 6.2.1 操作条件的类别

本部分定义了操作条件的两个类别. 通过触点VCC, 接口设备应向卡提供下列通常的电压支持.

A类: 5V

B类: 3V

因此, 卡和接口设备应或者仅工作在A类、或者仅工作在B类、或者工作在A类及B类〔以AB类表示〕.

A类卡应能操作在A类和AB类接口设备上. AB类卡应能操作在A类、B类和AB类接口设备上. B类卡应能操作在B类和AB类的接口设备上; 应以这种方法设计: 在A类操作条件下他们不被损坏.

#### 6.2.2 操作类别的选择

图3显示了接口设备如何选择适用于卡的操作条件的类别.

当在接口设备中可提供时, 用于卡的第1个操作条件将置于B类.

操作条件在A类时, 一个B类卡将不提供1个复位应答(见8)

如果卡不提供一个复位应答, 则接口设备应不激活卡, 至少需要10ms的延迟. 接口设备应提供下一个类别的操作条件.

如果提供一个复位应答, 不带类别指示器(见8.5.6), 则接口设备将应用或保持A类操作条件(当可提供, 或不激活此卡时)

如果卡提供一个带有类别指示器的复位应答, 并且接口设备支持应用一个卡支持的操作条件等类, 则一般操作将继续.

如果复位应答不激活当前操作条件类别, 但通过接口设备的另一个类别支持, 则接口设备将不激活卡, 之后需要至少10ms 的延时, 接口设备应用那个类别的操作条件.

注: 当以B类操作, 与ISO/IEC 7816—3:1989一致的一些卡将被损坏, 且他们必须仅用于A类接口设备.

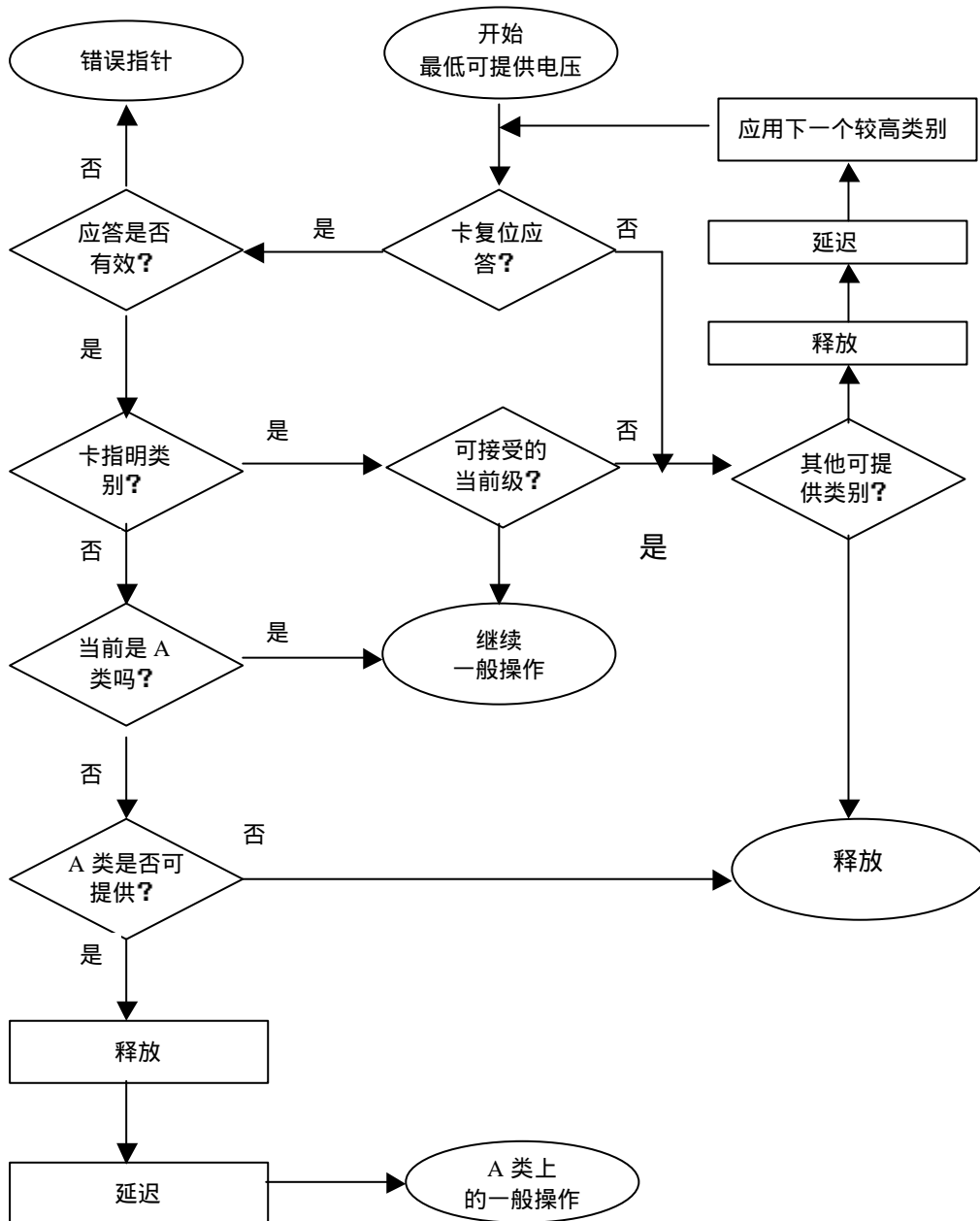


图3 通过接口设备选择操作条件的类别

## 6.3 电压和电流值

### 6.3.1 测量规定

所有测量相对于触点GND进行, 并在环境温度为0℃ - 50℃的范围内定义, 所有流入

卡的电流都假定为正。所有定时应相对6.3.2条到6.3.6条所定义的相应门限电平测量。

当触点相对于其电流小于1mA的GND来说保持在0伏和0.4伏之间时,电路为不工作状态。

### 6.3.2 VCC

本触点用来提供电源电压Vcc,在下表中,电流值是平均大于1ms,最大电流由卡定义。接口设备应能在规定电压值范围内传送此电流值或更大的电流。

表1 正常操作条件下Vcc的电特性

符号	条件	最小值	最大值	单位
Vcc	A类	4.5	5.5	V
	B类	2.7	3.3	
Icc	A类, 在最大允许频率 B类, 在最大允许频率 当时钟停止时〔见7.3.4〕		60 50 0.5	mA

不考虑下表所示的瞬间功耗,电源应保持规定范围内的电压值。

表2 I<sub>cc</sub>的尖峰值

类别	最大电荷量 <sup>a</sup>	最大持续时间	I <sub>cc</sub> 的最大变化量 <sup>b</sup>
A	20nA.s	400ns	100mA
B	10nA.s	400ns	50mA
a. 最大电荷量是最大持续时间和最大变化量乘积的一半。 b. 最大变化量是提供电流与平均电流值的差。			

### 6.3.3 I/O

本触点用作输入(接收模式)或输出(传送模式)。通过触点I/O的信息交换使用以下两种逻辑状态(见ISO1177中定义。):

——状态Z 如果卡和接口设备处于接收状态或由发方强制。

——状态A 如果这个状态是由发方强制。

当线路的两端处于接收模式时,这条线路将处于状态Z(高电平)。当线路的两端处于不匹配传输状态时,则该线路的逻辑状态可能是不固定的。在操作过程中,接口设备和卡不应同时处于传送状态。

当输入电压在允许范围内时,接口设备应能支持规定范围的输入电流。接口设备应在卡上连接一个电阻,以便在允许范围内用以稳定输出电压

表3 正常操作条件下I/O的电特性

符号	条件	最小值	最大值	单位
V <sub>IH</sub>		0.7 × Vcc	Vcc	V
I <sub>IH</sub>	V <sub>IH</sub>	-300	+20	μA
V <sub>IL</sub>		0	0.15 × Vcc	V
I <sub>IL</sub>	V <sub>IL</sub>	-1000	Vcc +20	μA
V <sub>OH</sub>	附加的上拉电阻: 20KΩ到Vcc	0.7 × Vcc	Vcc	V
I <sub>OH</sub>	V <sub>OH</sub>		+20	μA
V <sub>OL</sub>	I <sub>OL</sub> =1mA <sup>a</sup>	0	0.15 × Vcc	V
t <sub>R</sub> t <sub>F</sub>	C <sub>IN</sub> =30pF;		1	μs

	$C_{OUT}=30pF$			
I/O电压应保持在-0.3V和Vcc+0.3V之间				
a: 接口设备的实现不应要求卡吸入大于500μA的电流				

#### 6.3.4 CLK

本触点用于向卡提供时钟信号,时钟信号的实际频率值由f指定,频率值的范围见7.2和8.5.2.

时钟信号的工作周期应在稳定操作期间周期的40% - 60%,当频率从一个值转换到另一个值时,应注意保证没有比短周期的40%更短的脉冲,如8.5.3中表7表示.当转换频率值时,没有信息被改变,对于转换频率值,建议两个不同的时间:

- 在复位应答后立即进行
- 在一个成功的PPS过程完成后立即执行(见9.4)

表4 正常操作条件下CLK的电特性

符号	条件	最小值	最大值	单位
$V_{IH}$		$0.7 \times V_{CC}$	$V_{CC}$	V
$I_{IH}$	$V_{IH}$	-20	+100	μA
$V_{IL}$		0	0.5	V
$I_{IL}$	$V_{IL}$	-1000	+20	μA
$t_R$ $t_F$	$C_{IN}=30pF$		时钟周期的9%	

CLK电压应保持在-0.3V - Vcc+0.3V之间.

#### 6.3.5 RST

按照7.3.2(冷复位)或者7.3.3(热复位),本触点提供卡的复位信号.

表5 正常操作条件下RST的电特性

符号	条件	最小值	最大值	单位
$V_{IH}$		$0.8 \times V_{CC}$	$V_{CC}$	V
$I_{IH}$	$V_{IH}$	-20	+150	μA
$V_{IL}$		0	$0.12 \times V_{CC}$	V
$I_{IL}$	$V_{IL}$	-200	+20	μA
$t_R$ $t_F$	$C_{IN}=30pF$		1	μs

RST电压应保持在-0.3V - Vcc+0.3V之间

#### 6.3.6 VPP

在B类操作条件下,本触点保留待未来使用.

在A类操作条件下,本触点可用来提供编程或删除内部非易失性存储器单元的内容所需的电压.表6规定了触点Vpp上两种工作状态:中止状态和编程状态.除非卡请求工作状态,接口设备应将触点保持在中止状态.

表6 正常操作条件下VPP的电特性

符号	条件	最小值	最大值	单位
$V_{PP}$	中止状态	$0.95 \times V_{CC}$	$1.05 \times V_{CC}$	V
$I_{PP}$			20	mA
$V_{PP}$	工作状态	$0.975 \times P$	$1.025 \times P$	V
$I_{PP}$			1	mA
$t_R$ $t_F$		a	200	$\mu s$
对任意1秒时间取平均值时,功率不大于1.5W. 注: 1. 需要时,卡给接口设备提供P和I的值(见8.5.4). 2. Vpp状态控制在第10节和第11节规定,仅与A类操作条件有关.				
a: Vpp上电压改变的速率不应大于 $2V \cdot \mu s^{-1}$ .				

## 7 卡操作过程

### 7.1 正常操作过程

当卡的触点与接口设备的触点被机械地连接时,电路才被激活.

接口设备和卡的对话应顺序操作:

——接口设备激活电路;

——卡和接口设备之间信息交换,并由冷复位(见7.3.2)启动卡应答.

——接口设备释放电路.

电路的释放顺序应在卡上触点和接口设备上触点之间的机械断开之前结束.

### 7.2 激活

为启动机械连接的卡的互操作,接口设备应按图4所示顺序激活电路:

——RST置为状态L(见6.3.5)

——按照接口设备所选择的操作条件,VCC加电:A类或B类(见6.3.2和表1)

——接口设备上的I/O应置于接收状态(见6.3.3)

A类,Vpp置于中止状态(见6.3.6); B类,Vpp保留待未来使用.

——CLK提供时钟信号(见6.3.4).至少在复位应答期间,时钟f的频率值应在以下范围内:

· 1 ~ 5MHz:A类;

· 1 ~ 4MHz:B类.

电路的激活顺序结束后(RST为状态L,VCC加压,接口设备上的I/O为接收状态,当操作在A类时Vpp为中止状态,CLK提供一个合适且稳定的时钟),按照7.3.2和图4的规定卡准备好冷复位.

### 7.3 信息交换

#### 7.3.1 总则

如果卡支持操作条件的类别,则卡应按照第8章的内容应答任何复位.然后,接口设备



将启动卡的热复位。对于热复位的应答与对前一个的复位的应答不同,无论该复位是冷复位,还是热复位。在完成一个指明协商模式(见8.6)的复位应答后,接口设备可按照第9章的规定启动PPS交换。

命令的操作过程取决于传输协议。第10章规定了以接口设备为主的异步半双工字符串传输协议。第11章规定了异步半双工块传输协议。当不希望从卡信息时(例如:在一个命令完成后与开始下一个命令之前),如果卡支持时钟停止,则接口设备可停止时钟信号。

### 7.3.2 冷复位

按照图4所示,在 $T_a$ 时间对CLK加时钟信号。I/O线路应在时钟信号加于CLK的200个时钟周期( $t_a$ )内被卡置于状态Z( $t_a$ 时间在 $T_a$ 之后)。时钟加于CLK后,保持RST为状态L,至少400周期( $t_b$ 在 $T_a$ 之后)。

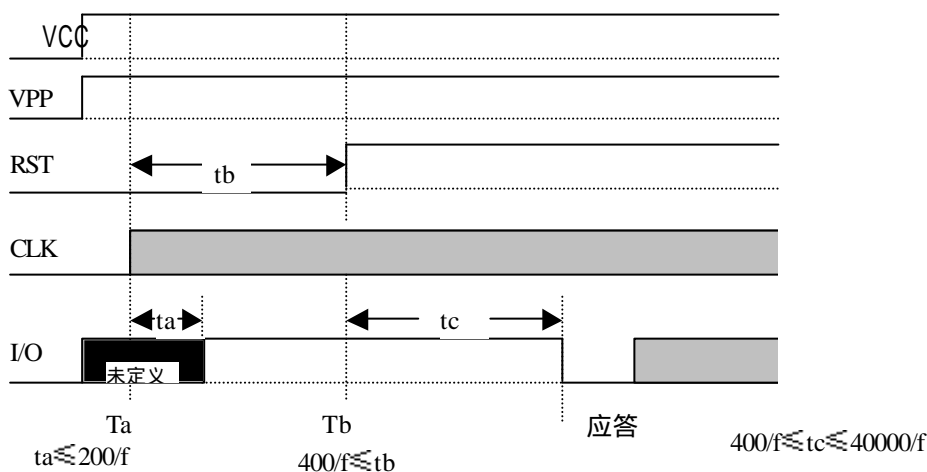


图4 激活和冷复位

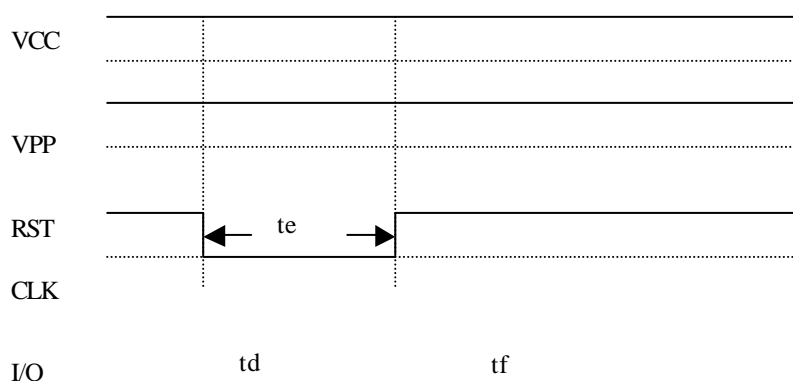
在时间 $T_b$ ,RST被置于状态H。I/O上的应答应在RST上信号的上升沿之后的400~40000个时钟周期( $t_c$ )内开始( $t_c$ 在 $T_b$ 之后)。

在RST处于状态H的情况下,如果应答信号在40000个时钟周期内仍未开始,RST上的信号将返回到状态L,且电路按照7.4被接口设备释放。

- 注:1. 假定卡的内部状态在冷复位前不定,这样卡的设计必须避免不适当的操作。
- 2. 卡的复位可以由接口设备在任意时间随意启动。

### 7.3.3 热复位

按照图5所示,当VCC和CLK保持稳定时,接口设备置RST为状态L至少400时钟周期(时间 $t_e$ )后,接口设备启动热复位。



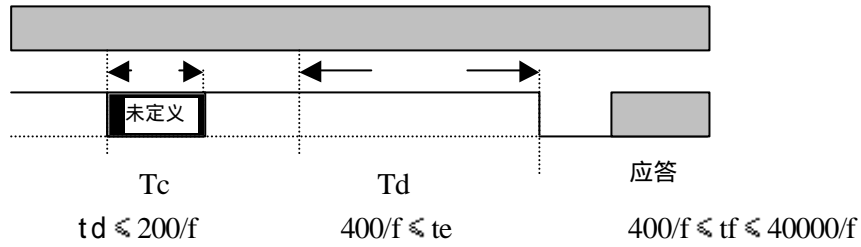


图5 热启动

在时间 $T_d$ , RST置于状态H. I/O的应答在RST上信号上升沿之后的400 - 40000个时钟周期( $t_f$ )之前开始(时间 $t_f$ 在 $T_d$ 之后).

在RST处于状态H时,如果应答信号未在40000个周期之后开始,RST上的信号将返回状态L,且电路按照7.4被接口设备释放.

#### 7.3.4 时钟停止

对于支持时钟停止的卡,当接口设备不希望从卡得到信息时,并且I/O保持在状态Z至少1860个时钟周期( $t_g$ ),则按照图6所示,接口设备可停止CLK上的时钟(在时间 $T_e$ ).

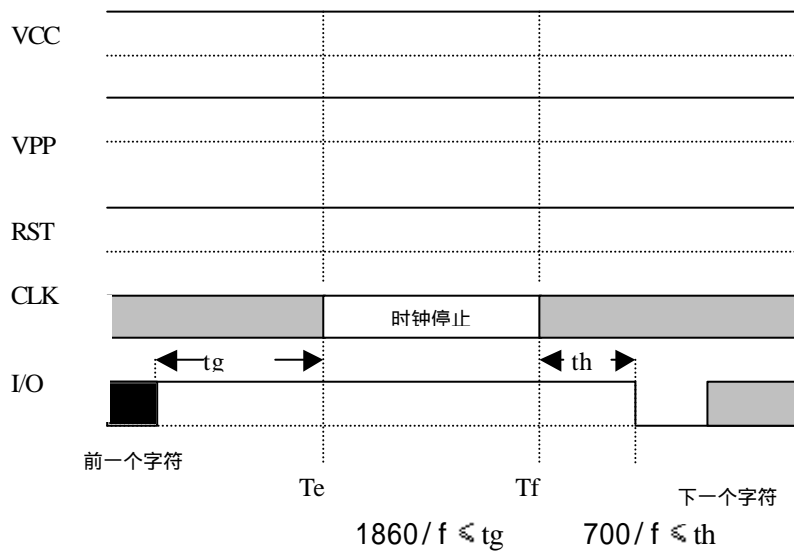


图6 时钟停止

n

当时钟被停止(从 $T_e$ 到 $T_f$ ),CLK应保持在状态H或状态L;这个状态由参数X指明(见8.5.5).

在时间 $T_f$ ,接口设备重启时钟并且I/O上的信息交换可在至少700个时钟周期后继续(时间 $t_h$ 在 $T_f$ 之后).

#### 7.4 释放

当信息交换结束或失败时(例如,无卡响应或发现卡被移出),接口设备应按以下顺序释放电路(见图7)

- RST应被为状态L

- CLK应被为状态L(除非时钟已在状态L上停止)
- VPP应被释放(如果它已被激活)
- I/O应被置为状态A
- VCC应被释放.

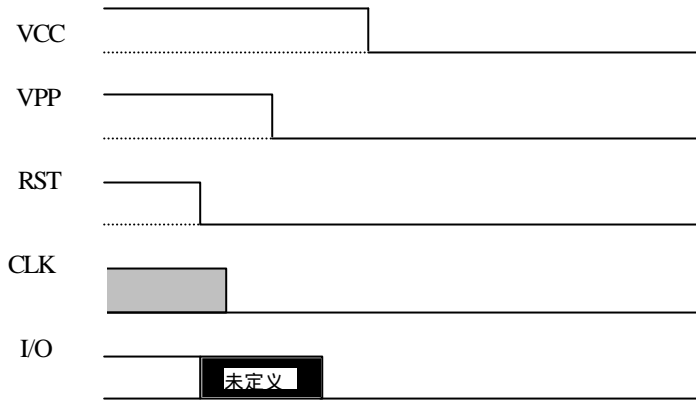


图7 释放

## 8 复位应答

### 8.1 一般结构

根据定义,复位应答是一系列字节的值,这些字节是由卡作为对复位命令的响应发送给接口设备的,在I/O电路上,每个字节在一个异步字符中传输。

每个成功的复位操作都会导致I/O上的一个初始字符TS,TS后面按照下面的次序跟有最多32个字符:

- T0 ..... 格式字符, 强制性的
- TA(i) TB(i) TC(i) TD(i) ..... 接口字符, 可选的
- T1 T2...Tk ..... 历史字符, 可选的
- Tck ..... 检测字符, 有条件的

- 初始字符定义了所有后继字符的解码协议。
- 格式字符声明了第一组接口字符和所有历史字符。
- 接口字符由格式字符声明的位图技术来指明。
- 历史字符由编码在格式字符中的一个数字来指明。
- 校检字符依赖于某些接口字符中参数T的值。

为了表示简明,以下用T0 TA(i) ...T1 ...Tck表示字节及传送字节的字符。

### 8.2 参数T

T参数指明了传输协议和/或接口字节的类型。在每个字节TD(i)[见(8.4.3.1)],

TA(2)(见8.5.7)或PPS0(见9.3),参数T的值由位b4到b1的编码值确定:

- T=0 异步半双工字符传输协议 在第8章中说明。
- T=1 异步半双工块传输协议 在第9章中说明。
- T=2和T=3 保留用于将来的全双工操作。
- T=4 保留用于增强的异步半双工字符传输协议。
- T=5到T=13 保留待未来使用。
- T=14 未由ISO/IEC JTC1 SC17标准化的传输协议

—— $T=15$  不属于传输协议,仅指明了全程接口字节的类型(见8.4.3.2)

### 8.3 异步字符

#### 8.3.1 基本时间单元

在复位应答期间,1etu应与372个时钟周期相等。

$$1\text{etu}=372/f$$

etu的一个可选值,见8.4.1。它的通用表达式,见8.5.2。

#### 8.3.2 字符帧

字符传输前,I/O端应被置为状态Z。如图8所示,一个字符包括10个连续的时刻,每一时刻不是在状态Z,就是在状态A。

——第一个时刻m1被置于状态A,这个时刻称为起始时刻。

——m2 ~ m9这八个时刻传送1个字节。

——最后一个时刻m10确保字符奇偶校验。它传送“奇偶校验位”。

图8 字符帧

在每个字符中,如果在时刻mn结束时状态改变,则从字符上升沿到mn下降沿间的延迟应是 $t_n=(n \pm 0.2)\text{etu}$ 。

发送方的时间起点是字符的上升沿。当寻找一个字符时,收方定期地对I/O取样:取样时间应少于 $0.2\text{etu}$ ,接收方的时间起点是在Z状态的最后一个观察点和A状态的第一个观察点中间。

接收方应在 $0.7\text{etu}$ (接收方时间)之前确认m1,然后应在 $(1.5 \pm 0.2)\text{etu}$ 收到m2,在 $(2.5 \pm 0.2)$ 收到m3, ...在 $8.5 \pm 0.2$ 收到m9,在 $9.5 \pm 0.2$ 收到m10,字符奇偶校验在不工作时进行。

注:这样确保在所有的测试区同传输区区别开。

两个连续字符上升沿之间的延迟至少是 $12\text{etu}$ ,例如,一个字符的持续时间 $(10 \pm 0.2)\text{etu}$ 加上保持时间。在保护时间,接口设备和卡都保持接收状态(因此I/O状态为Z)。

在复位应答期间,卡发出的两个连续字符的上升沿间的延迟应不超过 $9600\text{etu}$ ,这个最大值被称为“初始等待时间”。

#### 8.3.3 差错信号和字符重发

在复位应答期间,下列字符的重发过程取决于协议类型,该过程对使用协议类型T=0是强制性的,对于接口设备和其它卡来说是可选择的。

当奇偶差错时,在 $(10.5 \pm 0.2)\text{etu}$ (接收方时间)时,收方传送一个状态为A,最少为1etu,最大为2etu的差错信号,然后,收方将等待对有争议的字符重发见图9。

为了检测到一个差错信号,发方将检查I/O电路在 $(11 \pm 0.2)\text{etu}$ (发送方时间)时的状态,例如字符的上升沿之后:

——如果I/O为状态Z,即假定为正确接收

——如果I/O状态为A,即假定传输是不正确的。在检测到差错信号后的至少两

个etu的延迟之后，发送方重复该字符。

如果卡没有重发字符

——卡忽略接口设备来的错误信号并不应受其破坏；

——接口设备应能启动重复整个复位操作。

图9—字符传送和重发图

## 8.4 复位应答结构

### 8.4.1 初始字符和编码约定

图 10 为初始字符 TS 的结构：

— m1 到 m4 时刻定义同步序列(Z)AZZA。

— m5 到 m7 时刻以值 AAA 或 ZZZ 分别指明 000 反向或正向约定。

— m8 到 m10 时刻等于 AAZ。

图 10 初始字符 TS

注：同步序列允许接口设备决定卡上初始使用的 etu。etu 的可选值是 TS 最初两个下降边沿之间的延迟的三分之一。卡上的发送和接收机制〔包括 8.3.2 和 8.3.3 中描述的公差〕应与 etu 可选值的定义一致。

TS 定义了所有后继字符中数据字节的编码协议。该协议由下列组成：

— 通过九个时刻 m2 到 m10 的状态 Z 和 A 对值 1 和 0 编码。

— m2 到 m9 八个时刻的位重要性。

m2 到 m10 九个时刻中值为 1 的位的个数为偶数时，字符奇偶校验正确。

TS 有两个可能值，显示为处于状态 Z 或 A 的十个时刻的字符，并且按照编码协议，显示为 1 或 0 值的八位的字节。

— 字符(Z)AZZA AAAAZ 设定状态 A 编码值 1 以及 m2 时刻传输最高有效位(msb)处的反向约定。反向约定解码时，传输的字节等于“3F”。

— 字符(Z)AZZA ZZZAAZ 设定状态 Z 编码值 1 以及 m2 时刻传输最低有效位(lsb)处的正向约定。正向约定解码时，传输的字节等于“3B”。

图 11 图示了后面用到的字节框。字节由八个指定为 b8 到 b1 (值为 1 或 0) 的位组成；b8 是最高有效位 (msb) 而 b1 是最低有效位 (lsb)。

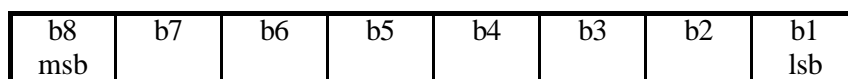


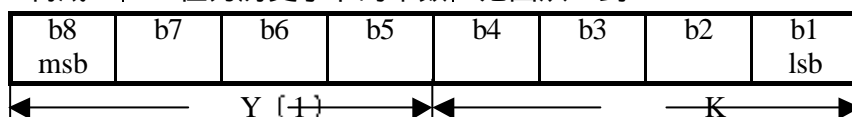
图 11 — 字节框

### 8.4.2 格式字节 T0

按照图 11，字节 T0 由两部分组成。

— 位 b8 到 b5 构成 Y(1)；每个等于 1 的位指明了后继接口字节的存在。

位 b4 到 b1 构成 K, K 值为历史字节的个数, 范围从 0 到 15.



Y (1) .....接口字节存在的标记

b5=1 时 TA(1)存在

b6=1 时 TB(1)存在

b7=1 时 TC(1)存在

b8=1 时 TD(1)存在

K.....历史字节的数目, 从 0 到 15

图 12 — T0 编码

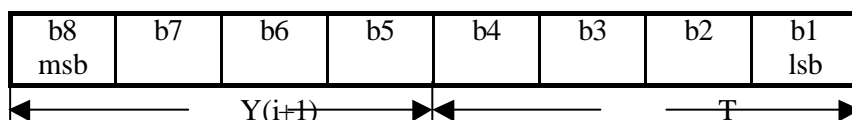
### 8.4.3 接口字节 TA(i) TB(i) TC(i) TD(i)

#### 8.4.3.1 TD(i)

按照图 13, 字节 TD(i)由两部分组成.

— 位 b8 到 b5 构成 Y(i+1); 每个等于 1 的位指明接口字节的存在.

— 位 b4 到 b1 构成 8.2 中定义的参数 T 的值.



Y(i+1).....接口字节存在的标记

b5=1 时 TA(i+1)存在

b6=1 时 TB(i+1)存在

b7=1 时 TC(i+1)存在

b8=1 时 TD(i+1)存在

T.....协议参考和/或接口字节限制符

图 13 — TD(i)编码

进而, T0 传输 Y(1)而 TD(i)传输 Y(i+1). 在传输 Y(i)的字节中, 位 b8 到 b5 表示与 b5 对应的 TA (i)、与 b6 对应的 TB (i)、与 b7 对应的 TC (i)、与 b8 对应的 TD (i) 是否按照这个顺序且在传输 Y(i)的字节后存在 (取决于相应的位是否为一),

如果 TD(i)不存在, 则接口字节 TA(i+1)、TB(i+1)、TC(i+1)和 TD(i+1)也 不存在.

如果两个或多个参数 T 的值存在于 TD(1)TD(2).....中, 它们应当按照数字升序存在. 如果存在, T=0 是第一个, T=15 是最后一个. TD(1)中禁止值 T=15.

“第一提供协议”如下定义:

— 如果 TD(1)存在, 则第一提供是 T.

— 如果 TD(1)不存在, 则唯一提供是 T=0.

#### 8.4.3.2 TA(i) TB(i) TC(i)

接口字节 TA(i)、TB(i)和 TC(i) (i=1,2,3,...) 是全局的或专用的

— 有关卡上集成电路参数的全局接口字节, 见 8.5.

— 有关卡提供传输协议参数的专用参考字节.

接口字节 TA(1)TB(1)TC(1)TA(2)TB(2)是全局的。接口字节 TC(2)是专用的；它是为 T=0 定义的,见 10.2.  $i>2$  时接口字节 TA(i) TB(i) TC(i)的解释依赖于 TD(i-1)中参数的值。

- 如果  $T \neq 15$ , 则字节是协议 T 专用的。
- 如果  $T=15$ , 则字节是全局的。

如果为参数 T 的同一个值定义了超过三个接口字节 TA(i) TB(i) TC(i)并在复位应答中存在, 则它们应相继存在于都指明同样值 T 的 TD(i-1)TD(i) [ $i>2$ ] 之后; 进而, 它们被明确识别为出现在 TD(i-1)中的第一、第二或第 n 个 T 出现之后。

注: 参数 T 与位图技术的组合 允许仅发送有用的接口字节, 并在需要为那些与不存在的接口字节对应的参数使用缺省值。

#### 8.4.4 历史字节 T1 T2.....TK

历史字节标明通用信息, 例如, 卡生产商、插入卡中的芯片、芯片的掩膜 ROM、卡的寿命状态, ISO/IEC 7816-4 规定了历史字节的内容。

如果 K 不空, 则复位应答在 K 个历史字节 T1 T2.....TK 上继续。

#### 8.4.5 校验字节 TCK

字节 TCK 的值应当是从 T0 到 TCK 的所有字节, 包括空位的异或值。

如果仅指明 T=0 (可能通过缺省), 则字节 TCK 不存在。如果 T=0 和 T=15 存在并在所有其它情况下, 字节 TCK 应当存在。

### 8.5 全局接口字节的内容

#### 8.5.1 总则

本条规定了全局接口字节的内容。ISO/IEC JTC1 SC17 保留了所有未在本条中定义的全局字节以及虽定义了但未使用的整数值以备将来使用。

本条规定了 TD(i-1)中 T=15 的第一次出现后  $i>2$  的字节 TA(1) TB(1)TC(1) TA(2) TB(2)和 TA(i)。这些字节以二进制的形式对无符号整数 FI、DI、II、PI1、N、PI2、XI 和 UI 进行编码, 这些无符号整数等于或用于计算此后出现的参数 F、D、N、P、I、X 和 U 的值。

- 如果存在, 为正确处理任一协议应解释该字节。
- 如果该字节不存在, 则当需要时, 相关参数使用缺省值。

TA(1)代码 (见 8.5.2)

- FI, 位 b8 到 b5 上的时钟率转换因子的引用, 见表 7。
- DI, 位 b4 到 b7 上波特率校正因子的引用, 见表 8。

TB(1)b8=0 代码处 (见 8.5.4)

- II, 位 b7 b6 上最大编程电流的引用, 见表 9。
- PI1, 位 b5 到 b1 上编程电压的值。

注: 接口设备可以忽略 TB(1)的位 b8。

TC(1)代码 (见 8.5.3)

- N, 计算八位额外保护时间的引用。

TA(2)是专用模式字节 (见 8.5.7 和 8.6)

TB(2)用八位上的编程电压值 PI2 编码以代替 PI1 (见 8.5.4)。

TA(i)在 TD(i-1)( $i>2$ )中的 T=15 的第一个出现后编码 (见 8.5.5 和 8.5.6)

- XI, 位 b8b7 上时钟停止指示的参考, 见表 10。
- UI, 位 b6 到 b1 上级别指示的参考, 见表 11。

注: 符合 ISO/IEC 7816-3:1989 的接口设备在 TD(i-1) ( $i>2$ )中的 T=15 后正常忽略 TA(i)

TB(i) TC(i)所不支持的协议的接口字节特性。

### 8.5.2 传输因子 F 和 D

参数 F 和 D 分别是时钟率转换因子和波特率调整因子。在电路输入/输出上使用的 etu 依赖于传输因子 F 和 D 的实际值。etu 应等于 F/D 时钟周期。

$$1 \text{ etu} = F/D \times 1/f$$

频率 f 的最小值应当为 1MHz。最大值以 Fi 的函数的形式在表 7 中给出。缺省最大值是 5MHz。

为计算 etu，F 和 D 因子对应当采用下面三对值：

- Fi 和 Di，按照表 7 和 8 在 TA(1)中由卡指示的值；如果 TA(1)不存在，则 Fi 和 Di 设为缺省值；
- Fd 和 Dd，缺省值为 372 和 1；
- Fn 和 Dn，在 Fd 到 Fi 和 Dd 到 Di 范围里成功的 PPS 交换所协商的值。

在复位应答期间，应用 Fd 和 Dd，复位应答后，F 和 D 的值取决于操作模式（见 10.6）。

- 协商模式中，Fd 和 Dd 应继续应用直到 PPS 交换成功完成（见 9.4），PPS 成功交换后，Fn 和 Dn 立即应用。
- 专用模式中（见 8.6.2）
  - □ 如果 TA(2)中 b5=0，复位应答成功完成之后立即应用 Fi 和 Di。
  - □ 如果 TA(2)中 b5=1，使用隐含值。

### 8.5.3 额外保护时间 N

参数 N 是用于从接口设备到发送字符的卡的额外保护时间。从卡发送字符到接口设备不用额外保护时间。缺省值 N=0。

在 0 到 254 范围里，在准备接收下一字符前，N 指明卡要求从前一个字符（也是由卡或接口设备发送的）上沿的后续延迟。

$$12 \text{ etu} + [Q \times N/f]$$

公式中，Q 取两个值中的一个：

- F/D，即，用于计算 etu 的值，当 T=15 不存在于复位应答中时，
- Fi/Di，当 T=15 在复位应答中时。

N=255 指明在传输协议期间，两个连续前沿之间的最小延迟在传输的两个方向是一致。这个最小延迟值是

- T=0 时，12etu
- T=1 时，11etu

表 7 — Fi，指明的时钟率转换因子的值

FI	0000	0001	0010	0011	0100	0101	0110	0111
Fi	372	372	558	744	1116	1488	1860	RFU
f (max) MHz	4	5	6	8	12	16	20	—

RFU=留作未来使用

FI	1000	0001	0010	0011	0100	0101	0110	0111
Fi	RFU	512	768	1024	1536	2048	RFU	RFU
f (max) MHz	—	5	7,5	10	15	20	—	—

表 8 — 指明的波特率校正参数的值



DI	0000	0001	0010	0011	0100	0101	0110	0111
Di	RFU	1	2	4	8	16	32	RFU

DI	1000	1001	1010	1011	1100	1101	1110	1111
Di	12	20	RFU	RFU	RFU	RFU	RFU	RFU

#### 8.5.4 编程参数 P 和 I

编程参数 P 和 I 分别是编程电压和最大编程电流; 它们定义了接触点 VPP 上的编程状态。

- 编程电压:  $V_{pp}=PV$
- 最大编程电流:  $I_{pp}=ImA$

在 5 到 25 范围内, PI1 给出了 P 的值, 单位为伏. PI1=0 指明在卡中 VPP 不是电连接, 该卡从接触点 VCC 供电电源上内部生成编程电流. 任何其它 PI1 值留作未来使用.

在 50 到 250 范围内, PI1 给出了 P 的值, 单位是十分之一伏. 任何其它 PI1 值留作未来使用. 如果 PI2 存在, 则 PI1 的值应忽略.

如果 T=15 不在复位应答中, 缺省值是 P=5 和 I=50. 如果 T=15 存在, VPP 不在卡中连接, 除非 TB(1)和/或 TB(2)存在.

表 9 — 最大编程电流 I

II	00	01	10	11
I	25	50	RFU	RFU

#### 8.5.5 时钟停止符 X

参数 X 按照表 10 指明卡支持 (XI≠00) 或不支持 (XI=00) 时钟停止, 以及支持时, 当时钟停止时在 CLK 上优先选用哪个电状态. 缺省值是 X= “不支持时钟停止”.

表 10 — 时钟停止指示符 X

XI	00	01	10	11
X	不支持	状态 L	状态 H	无优先

#### 8.5.6 级别指示符 U

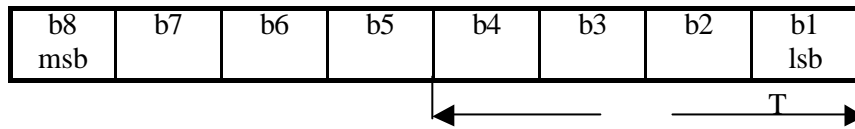
参数 U 指明了卡允许的操作条件的级别. 按照表 11, UI 的每个位代表了 6.2.1 中定义的操作条件的级别: b1 是 A 类, b2 是 B 类. 缺省值是 U= “仅支持 A 类”.

表 11 — 级别指示符 U

UI	00 0010	00 0010	00 0011	任何其它值
U	仅 A	仅 B	A 和 B	RFU

#### 8.5.7 专用模式字节 TA(2)

TA(2)是专用模式字节. 按照图 14, 它描述了卡操作专用模式的有关特点(见 8.6.2).



b8.....改变操作模式能力指示符  
 b8=0 时有改变能力  
 b8=1 时无改变能力  
 b7—b6.....RFU (不用时 00)  
 b5.....参数定义指示符  
 b5=0 时由接口字节定义  
 b5=1 时不由接口字节明确定义  
 T.....在专用模式中使用的协议

图 14 — TA(2)编码

## 8.6 操作模式

### 8.6.1 概述

复位应答后，卡是下面两种操作模式之一：

- 或者在 TA(2)存在时是专用模式
- 或者在 TA(2)不存在时是协商模式

图 15 图示了卡操作模式的开关和选择

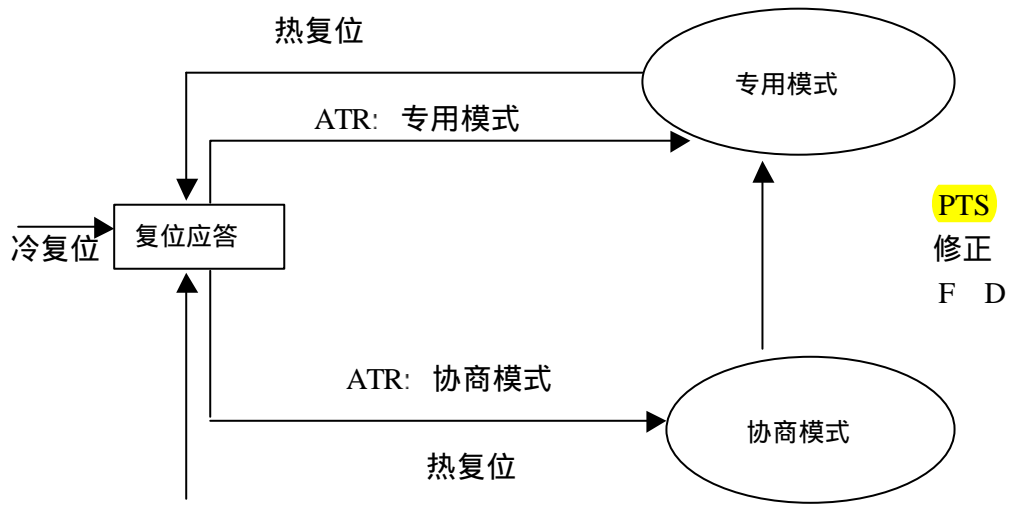


图 15 — 模式选择与开关

### 8.6.2 专用模式

在专用模式中，紧随复位应答之后，由 TA(2)指示的协议应使用：

- TA(2)中 b5=0 时，使用 Fi 和 Di;
- TA(2)中 b5=1 时，使用缺省值。

IFD 可执行热复位来调用 ICC 中的协商模式。

注：

- 1、在不知专用模式存在的情况下，ICC 发送 TA(2)给 IFD，则 ICC 不能使用额外

的复位切换到协商模式。

2、若 IFD 检测到一个 TA(2)字节,则在复位应答完全接受前,或卡已超时的情况下,IFD 不能发出第二个复位命令。

### 8.6.3 协商模式

在协商模式中,只要 IFD 发送给 ICC 的第一字节允许在 PPS 请求与协议命令之间有明显差别,则“缺省选择”是可能的。

一 在复位应答后无 PPS 请求,则“首选协议”〔见 8.4.3.1〕将使用 Fd 和 Dd〔见 8.5.2〕。

一 当协议由 ICC 和/或参数 F、D 的其它值〔F 范围为 Fd 到 Fi, D 的范围为 Dd 到 Di〕提供时,IFD 应发送一个带 Fd 和 Dd 的 PPS 请求,以便从协商模式转到专用模式。成功完成 PPS 交换后〔见 9.4〕,协商协议应使用 Fn 和 Dn。

**如果复位应答仅提供一个协议〔T=0 到 14〕和 Fd、Dd,则该协议应使用 Fd 和 Dd 且紧随复位应答之后。相应的,这样的卡不必支持 PPS。**

既不支持 PPS 又不支持“首选协议的”IFD 可采用复位 ICC 以从协商模式转到 IFD 支持的专用模式,或者可以拒绝卡。

注:

1、协商模式的热复位可以将 ICC 转到专用模式。

2、如果多协议卡包含 T=0,则 T=0 应首先出现在复位应答的第一位中,因此,对于协商模式的卡,只有 T=0 可以作为缺省选项。

3、如果 T=0 或 T=1 带有值 Fi 和 Di,且 Fi、Di 不等于 Fd、Dd,则 IFD 可以:

一 选择带有 Fd、Dd 的缺省协议,

一 发送带 Fd 和 Dd 的 PPS 请求,以协调 Fn、Dn

## 9 协议和参数选择

### 9.1 概述

**本条规范了明确的协议和参数选择。应用在指示协商模式的复位应答之后。**

PPS 请求和应答以与复位应答相同的方式发送。例如,相同的波特率〔使用 Fd 和 Dd〕,符合 TS〔8.4.1〕规定的协议,连续两个字符的上沿具有最小延迟 12etu。然而如果 IFD 字节 TC(1)出现在复位应答中,且值不为“FF”,则应保证有足够的保护时间〔见 8.5.3〕。PPS 响应的两个连续字符的上沿之间的延迟不应超过“初始等待时间”〔见 8.3.2〕。

### 9.2 PPS 协议

只有 IFD 被允许开始 PPS 交换

一 IFD 应发送一个 PPS 请求给 ICC。

一 如果 ICC 收到一个错误 PPS 请求,则它不作任何响应。

一 如果 ICC 收到一个正确 PPS 请求,则应返回一个 PPS 响应,否则将超过初始等待时间。

一 如果超过初始等待时间,则 IFD 或者复位,或者拒绝 ICC。

一 如果 IFD 收到错误 PPS 响应,则 IFD 或者复位,或者拒绝 ICC。

一 如果 PPS 交换失败,则 IFD 或者复位,或者拒绝 ICC。

### 9.3 PPS 请求与相应的结构和内容

PPS 请求和响应分别包括一个初始字节 PPSS,后随格式字节 PPS0,三个可选参数

字节 PPS1、PPS2 和 PPS3 以及一个检测字节 PCK (见图 16)。

PPS 识别 PPS 请求或响应并等于“FF”。

PPS0 通过位 b5、b6、b7 分别指明可选字节 PPS1、PPS2、PPS3 的存在。位 b4 到 b1 传输参数 T 的值以提出协议。位 b8 留作未来使用并设定为 0。

PPS1 允许 IFD 对卡提出 F 和 D 的值。

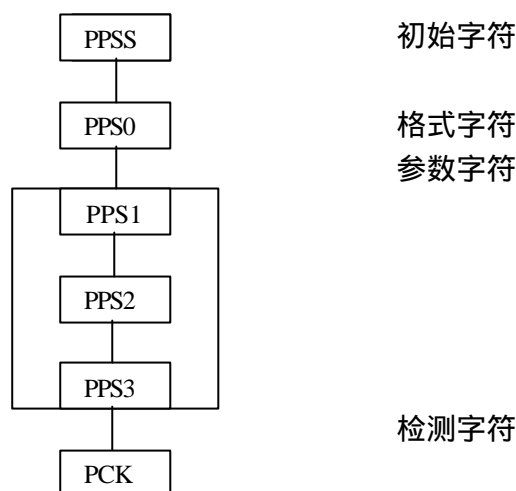


图 16-PPS 请求和相应的结构

#### 9.4 成功的 PPS 交换

如果 PPS 响应准确反应 PPS 请求，则 PPS 交换是成功的。这是最普通的情况，也可能发生其它情况。

当 PPS 响应为下列情况之一时，该 PPS 交换也是成功的：

- PPS 响应 = PPS 请求。
- PPS0 响应：
  - 应回送 b1 至 b4。
  - 回送 b5 或将其置为 0。
    - 如果 b5=1，PPS1 响应 = PPS1 请求。
    - 如果 b5=0，则没有 PPS1 响应，就意味着应使用 Fd 和 Dd。
  - 回送 b6 或将其置为 0。
    - 如果 b6=1，PPS2 响应 = PPS2 请求。
    - 如果 b6=0，则 PPS2 响应和 PPS2 请求都不存在。
  - 回送 b7 或将其置为 0。
    - 如果 b7=1，PPS3 响应 = PPS3 请求。
    - 如果 b7=0，则 PPS3 响应和 PPS3 请求都不存在。

PPS 交换的其它情况都应被解释为不成功。

### 10 T=0，异步半双工字符传输协议

#### 10.1 范围

本节定义了异步半双工字符传输中使用的命令的结构和处理。这些命令由 IFD 启动。本节包括传输控制和专用于卡的控制。

本协议在复位应答 (见 8) 或成功的 PPS 交换之后开始 (见 9)。

#### 10.2 字符级

字符帧同8.3里为复位应答所定义的一样,使用8.4.1中TS定义的协议,同时按照8.6中的操作模式来考虑8.5.2和8.5.3.

任何由过程字节激发的VPP传输都应从字符的上升沿开始,且不超过12etu.

在复位应答中,专用接口字符TC2在b8-b1上编码整型值WI.空值留待将来使用.当复位应答中没有TC2出现时,WI的缺省值为10.由卡发出的任何一个字符的上升沿和由ICC或IFD发出的前一个字符的上升沿之间的间隔应不超过 $960 \times WI \times (Fi/f)$ 个etu.这个最大延迟时间称为工作等待时间.

当超出工作等待时间时,VPP应被置为或保持空闲状态.

## 10. 3 命令的结构和处理

### 10.3.1 概述

命令总是由接口设备启动,他以一个5字节的报头通知卡做什么,并且允许在卡发出的过程字节的控制下传输数据字节.

为了区分输入数据传输指令(执行时数据进入卡)和输出数据传输命令(执行时数据离开卡),假设卡和接口设备预先知道数据方向.

### 10.3.2 命令报头

接口设备通过五个连续字节传送一个报头,这五个字节指定为CLA,INS,P1,P2,P3.

——CLA是指令类别,值'FF'为PTS保留(见8.3.3).

——INS是指令类别中的指令代码.指令代码只有当最高有效半字节不是'6'和'9'时才有效.

——P1、P2是一个完成指令代码的参考符号(例如地址).

——P3对指令期间被传输的数据字节(D1...Dn)的数目n编码.在输出数据的传输命令中,P3=0表示从卡发送256个字节的数据.在输入数据的传输命令中,P3=0代表无数据输入.

这样,在一个5字节报头传输之后,接口设备等待一个过程字节.

### 10.3.3 过程字节

#### 10.3.3.1 概述

过程字节的值将指明接口设备请求的动作.已规定了三种类型的过程字节:

——NULL的值为60.

——在ACK中,除了值'6X'和'9X'以外,在ACK字节中的七个最高有效位(b8至b2)全都等于INS字节中相应位或与之互补.

——SW1的值为'6X'或'9X',但不包括'60'.

在每一个过程字节中,卡可以用一个ACK或NULL字节来把这个命令继续进行下去,或以适当的不应答表示不赞同,或用结束序列SW1-SW2结束这个命令.

字节	值	VPP状态	传输的数据	接受
NULL	'60'	VPP上无进一步动作	无	一个过程字节
ACK	INS	VPP空闲	所有剩余数据	一个过程字节
	INS $\oplus$ '01'	VPP激活	所有剩余数据	一个过程字节

	INS⊕'FF',	VPP空闲	下一个数据字节	一个过程字节
	INS⊕'FE',	VPP激活	下一个数据字节	一个过程字节
SW1	'6X' (≠'60'), '9X'	VPP空闲	无	一个SW2字节

表12 — 过程字节

### 10.3.3.2 NULL 字节

NULL表示不对VPP状态和数据传输施加任何影响。IFD仅等待过程字节。

### 10.3.3.3 确认字节

ACK字节用于控制VPP状态和数据传输〔见4.3.6, 表6和8.5.4〕:

——当用INS字节对ACK字节进行异或运算结果为'00'或者'FF'时,接口设备保持或者置VPP为空闲状态。

——当用INS字节对ACK字节进行与或运算结果为'01'或者'FE'时,接口设备保持或者设置VPP为激活状态。

——当ACK字节中的七个最高有效位和INS字节中的相应位数值相同时,如果有剩余数据字节,那么,要传输所有余下的数据字节(Di...Dn)。

——当ACK字节中的七个最高有效位和INS字节中的相应位互补时,如果有余下数据,那么仅仅下一个数据字节(Di)被传送。

在这些动作完成之后,接口设备等待一个新的过程字节。

### 10.3.3.4 状态字节

**SW1要求将VPP置为或保持在空闲状态。** IFD等待一个传输SW2字节的字符。对SW2的值无限制。

结束序列SW1-SW2在命令的结尾给出卡的状态。SW1-SW2='90'-'00'标志正常结束。本部分没有解释SW1字节为'9X'时其它的结束序列。这些结束序列与应用本身相关。

当SW1的最高有效半字节等于'6'时,SW1的含义是与应用无关的,定义如下五个值:

- '6E' 卡不支持指令类型;
- '6D' 指令代码没有被编程或者无效;
- '6B' 参考错误;
- '67' 长度错误;
- '6F' 没有给出准确的诊断。

其它值保留给将来使用。

当SW1既不等于'6E',也不等于'6D'时,卡支持指令。

## 11 T=1, 异步半双工块传输协议

### 11.1 范围和规则

**本节定义了异步半双工块传输协议使用的命令的结构和处理。这些命令由IFD和ICC启动。本节包括了卡专用的控制,以及诸如流控制、块链和错误校正这样的数据传输控制。**

块传输协议在复位应答〔见8〕或一个成功的PPS交换〔见9〕之后开始,其主要

特征如下:

- 协议从IFD发送的第一个块开始, 然后交替发送一个块;
- 块是可交换的最小的数据单位, 块可以用于传输
  - 对传输协议透明的应用数据;
  - 包括传输差错处理的传输控制数据.

——块结构允许在处理传输的数据之前检测收到的块.

本协议按照OSI参考模型的分层设计原理, 特别注意了将各层界面间的相互影响减到最小, 被定义的有三层:

- 物理层, 符合11.3的异步字符传输;
- 数据链路层, 被定义为字符部分和块部分, 字符部分进行块识别〔识别块的开始和结束〕, 并保证控制符合11.6. 块部分按照11.7进行块交换.
- 应用层, 用于处理命令, 这些命令在每一方向至少包含一个块或一连串的交流.

## 11.2 术语和缩略语

见第4章.

## 11.3 字符帧

字符帧同8.3(不包括8.3.3)为复位应答所定义的一样, 使用8.4.1中TS规定的协议, 同时按照8.6中的操作模式将8.5.2和8.5.3考虑进来.

按照8.3.3, 不使用错误信号和字符重复, 从而使一个块中连续的两个字符的上沿边的延迟减少到11etu, 这与8.5.3中规定的接口字节TC(1)一致.

加上差错检测编码外〔见11.4.4和11.5.4〕, 字符奇偶检验还允许检测块.

## 11.4 块帧

### 11.4.1 概述

一个块由一串字节组成, 每个字节以异步字符的形式传输. 块有下列域构成〔见图17〕:

- 起始域(强制性的)包括节点地址字节、协议控制字节和长度字节;
- 信息域(可选的)由0-254个字节组成;
- 终止域(强制性的)包括一个或两个字节.

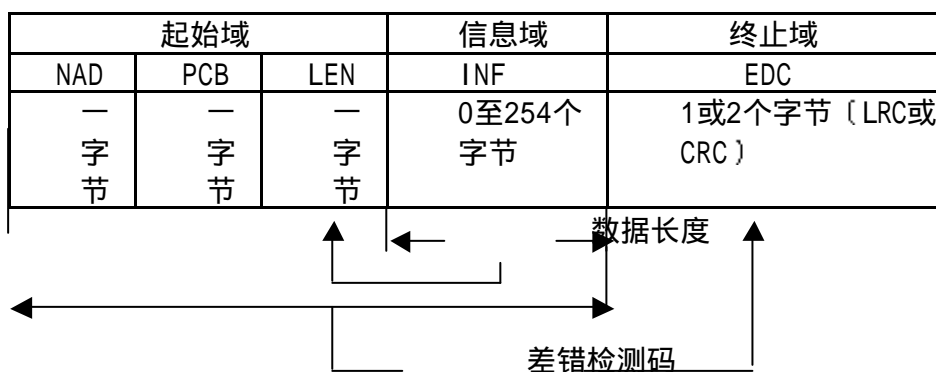


图17 一块帧

本协议定义了三种基本块类型:

- 信息块〔I块〕用于传送应用层信息. 另外, 它传输肯定或否定的确认信息.

——接受准备块【R块】用于发送肯定或否定的确认信息。它的信息域不出现。

——管理块(S块) 用于IFD和ICC之间交换控制信息，  
S块的信息域存在与否取决于S块控制功能的需要。

注：这种分类允许协议控制的设计和微码应用部分的设计彼此相对独立。

#### 11.4.2 起始域

##### 11.4.2.1 节点地址字节

节点地址(NAD)是用于标识块的源和预期目的的一个字节。NAD可用于区分同时存在的多逻辑连接。

b1到b3位指明源节点地址SAD, b5到b7位指明目的节点地址DAD。 b4和b8位用于表示VPP状态控制(见11.6.1)。

**在不使用编址时，SAD和DAD的值都应被置0。当SAD与DAD的值相同时，NAD的其它值留待将来使用。**

由IFD发送的第一个块的NAD确定了SAD和DAD地址的逻辑连接关系。在随后的块中的NAD域也包含相同的SAD/DAD地址对,并具有相同的逻辑关系。在后续的信息交换期间内其它的逻辑连接同样也由相应的SAD/DAD对定义。

注:例如,由IFD发送的块,其SAD的值为X、DAD的值为Y;由ICC发送的块,SAD的值为Y, DAD的值为X,这属于一个逻辑连接,标记为(X,Y)。然而,如由IFD发送的块其SAD值为V、DAD的值为W,由ICC发送的块其SAD值为W, DAD的值为V,则属于另一个逻辑连接(V,W)。

##### 11.4.2.2 协议控制字节【PCB】

**协议控制字节用于传送控制传输所需要的信息。**

本协议定义了三种基本块类型,编码细节见图18, 19, 20。

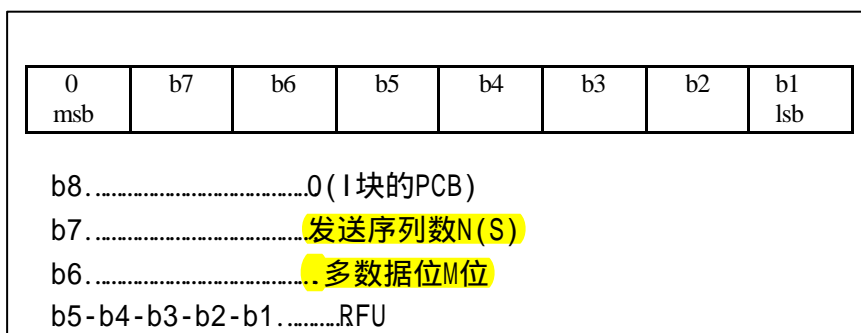
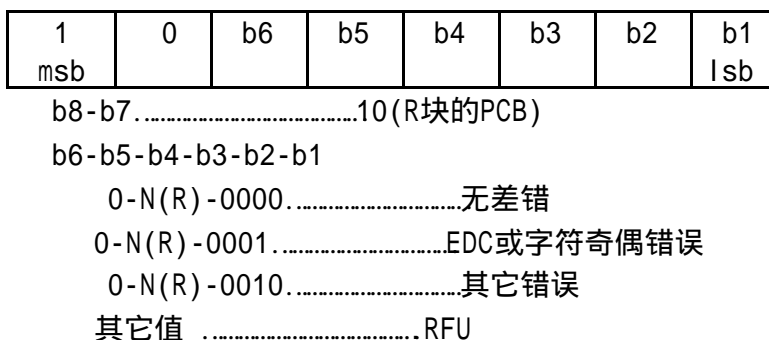


图18 — I块PCB编码



注一 按照N(R)的值,可以知道R块是否有一个错误。位b4至b1的值可选



图19 — R块PCB的编码

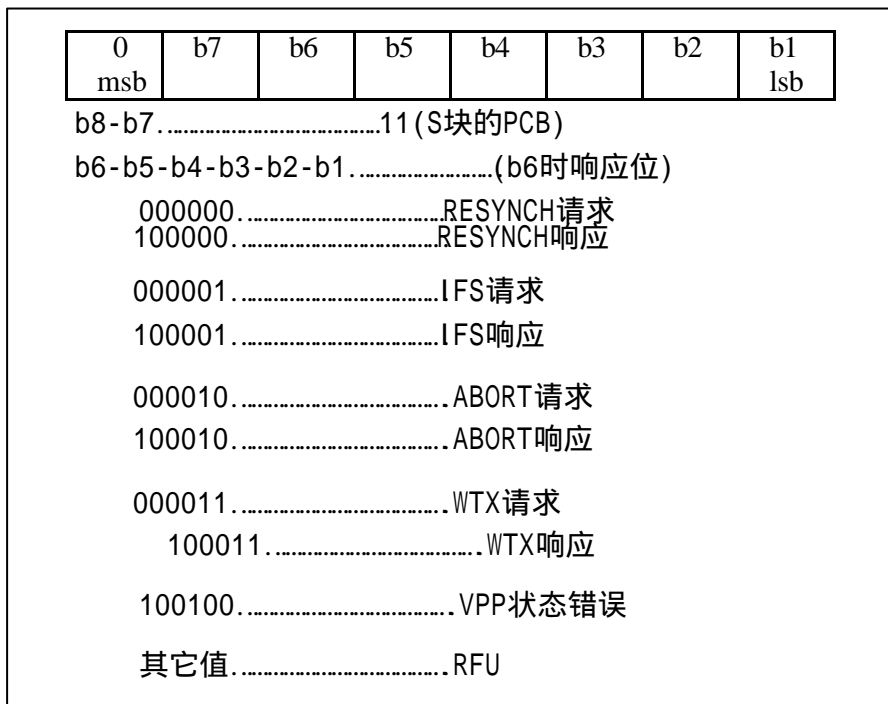


图20 — S块PCB的编码

11.4.2.3 长度【LEN】

LEN指示其块的信息域中被传输的位数(见11.5.2)。编码应是:

- “00”” 表明不存在信息域。
- “01”至“FE” 代表信息域中的字节数,对应为1到254个。 —— “FF”

留待将来使用。

11.4.3 信息域(INF)

- 对INF的使用取决于块的类型。
  - I块中的INF传送应用信息。
  - R块中不存在INF。
  - S块中的INF传送应用信息。
    - INF应与S块中的一个单独字节一起存在,负责调整IFS和WTX。
    - 在一个指示VPP状态出错或管理链中止或再同步的S块中不存在INF。

11.4.4 终止域

该域是强制性的, EDC传输块的差错检测编码。协议定义允许该域是LRC(纵向冗余校验)或CRC(循环冗余校验)。LRC长度为一个字节, CRC长度为两个字节。LRC的值与块中所有字节进行异或运算时结果都为零。关于CRC的值见ISO/IEC 3309。

## 11.5 协议参数

### 11.5.1 T=1时的特殊接口字节

当特殊接口字节TA(i), TB(i)和TC(i)出现在复位应答中, 且在TD(i-1)(i>2)中的T=1第一次出现之后时, 这些接口字节用来将协议参数设为非缺省值。

为了表示简洁, 这三个字节被命名为第一TA(i), 第一TB(i)和第一TC(i)。

### 11.5.2 信息域尺寸

#### 11.5.2.1 卡的信息域尺寸【IFSC】

IFSC是卡能够接收的各块中的信息域的最大长度。IFSC的初始值由第一TA(i)给定。缺省值为32。

#### 11.5.2.2 接口设备的信息域尺寸【IFSD】

IFSD是接口设备能接收的各块中的信息域的最大长度。初始值定为32。

#### 11.5.2.3 IFSC和IFSD的编码

IFSC和IFSD在协议启动时被初始化。协议执行过程中, 由S(IFS请求)和S(IFS响应)调整IFSC和IFSD, 其中INF由一个名为IFS的字节组成。任何情况下, 第一TA(i)和IFS字节应按下述规则编码:

'00' 和 'FF'                      留待将来使用;  
'01' 至 'FE'                      为数字1至254。

注: 块的尺寸是在起始域、信息域和终止域中被传输的所有字节的总数。块的最大尺寸等于IFSC加上4或5(视终止域的长度而定)。

### 11.5.3 等待时间

#### 11.5.3.1 字符等待时间【CWT】

字符等待时间定义为同一块中两个连续字符起始沿之间的最长时间。见图21。

注: 当可能存在长度差错时, CWT可以用来检测一个块的结束。

第一Tb(i)的最低有效半字节(b4至b1)编码为字符等待时间整数值(CWI), 其范围为0—15, CWT的计算公式为:

$$CWT = (2^{CWI} + 11) \text{etu}$$

因此CWT的最小值等于12etu。

CWI的缺省值为13。

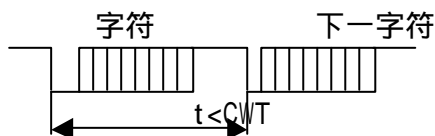


图21 字符等待时间

#### 11.5.3.2 块等待时间(BWT)

一个块等待时间被定义为送达到卡的最后一个字符的起始沿与由卡发送出的第一个字符的起始沿之间的最长时间。见图22。BWT用来检测无响应的卡。

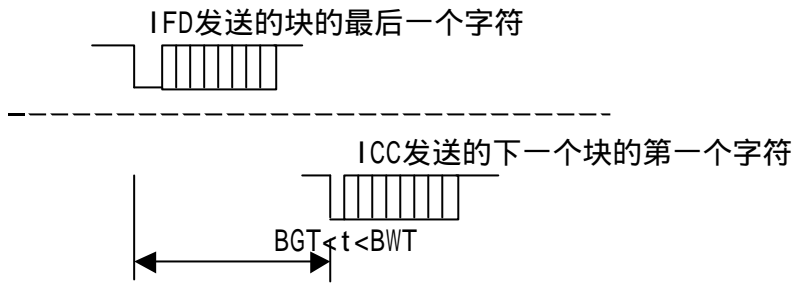


图22 块等待时间和块保护时间

第一TB(i)的最高有效半字节(b8至b5)编码为块等待时间BWI整数值,其范围为0-9,10-15留待将来使用。BWT的计算公式为:

$$BWT=2^{BWI} \times 960 \times 372 / f \times s+11etu$$

BWI的缺省值为4。

#### 11.5.3.3 块保护时间(BGT)

块保护时间为两个相对方向发送的连续字符的起始沿之间的最短时间。因此一个已接收块的最后一个字符与一个被传输块的第一个字符之间的迟延至少应为BGT但小于BWT。见图21。

BGT的值应为22etu。

#### 11.5.4 错误检测编码

第一TC(i)的位b1规定使用的错误检测编码为:

- CRC 如果b1=1
- LRC 如果b1=0

将位b8至b2置为0以留待将来使用。

### 11.6 数据链路层——字符成分

#### 11.6.1 VPP 状态控制

VPP状态〔见6.3.6,表6和10.5.4〕由接口设备,在由卡发送的NAD和PCB字符控制下进行管理。NAD的b8位和b4位指示

- b8=0, b4=0 VPP置为0或保持空闲状态。
- b8=1, b4=0 VPP置成编程状态。直到接受PCB字符。
- b8=0, b4=1 VPP置成编程状态直到接口设备接收另一个NAD字符。
- b8=1, b4=1 为禁用。

如果NAD上发生奇偶错,则VPP应置为或保持空闲状态。

如果发生超时,即:在CWT或BWT期间卡发送一个预期字符失败,则VPP应返回或保持空闲状态。

一个字符触发的所有VPP传输应发生在该字符上升沿起的12etu期间。

#### 11.6.2 无差错的操作

协议开始时,IFD就有权发送。

当接口设备被指定为协议T=1时,仅发送块。

当ICC或IFD已发送了一个完整块时,它转换到接收状态。当ICC或IFD按照长度子域的字符数完成接收时,它将有权发送。

### 11.7 数据链路层——块成分

### 11.7.1 标志

下述标志用于协议的描述.

I块由I(N(S),M)指示:

N(S)是块的发送的序号,M是多数数据位(见11.7.2.2).

$N_a(S)$ ,  $N_b(S)$ 区分由源A或B发送的序号,下标a和b标注N(S).

R块由R(N(R))指示,其中N(R)是预期的I块的个数.

S块如下表示:

S(RESYNCH请求)	S块再同步
S(RESYNEH响应)	S块再同步
S(IFS请求)	S块提供信息域的最大尺寸
S(IFS响应)	S块确认IFS
S(ABORT请求)	S块指示ABORT请求
S(ABORT响应)	S块指示ABORT响应
S(WTX请求)	S块请求扩大等待时间
S(WTX响应)	S块扩大等待时间响应
S(VPP状态差错响应)	S块通知卡VPP差错

S(IFS...)和S(WTX...)包括INF,它们的编码在11.7.2.3的准则3和4中定义.

### 11.7.2 无差错操作

#### 11.7.2.1 通用规程

在协议开始时,IFD发送到ICC的第1个块应为一个I块或S块.

在一个块(I块R块或S块)完成发送之后,应在开始传输下一个块之前应接收一个确认(信号),描述如下:

I块带着它的发送序号N(S)〔由IFD发送的I块的N(S)数与由ICC发送的I块的N(S)数分别计数〕.N(S)包含1个位并且以模2计数.在某个传输协议开始时或在再同步之后,N(S)的初始值为0;之后每发送一个I块其值就会改变.

R块带着N(R),N(R)是在下一个预期的I块中N(S)的值〔在无差错操作中R块用于链接I块,见11.7.2.2〕.

当收到下列信息时,可确认已收到I块:

——此I块的N(S)不同于上一个受到的I块的N(S).

——已接收的下一个R块的N(R)不同于发送的I块的N(S)(见11.7.2.3中的准则2.2).

S块中不载有数目.S(...请求)块载有非确认.S(...响应)块确认某已接收的S(...请求)块.

#### 11.7.2.2 链接

链接功能允许IFD或ICC传输比IFSC或IFSD长的信息(应用数据).

如果IFD或ICC传输的信息必须比相应的IFSC或IFSD长,则该信息应分为几个信息块,

每个块的LEN应小于或等于IFSC或IFSD,并且采用链接功能发送多个块.

图23表示了链接功能.

Applic	ation	Date
--------	-------	------

有下列各段传输

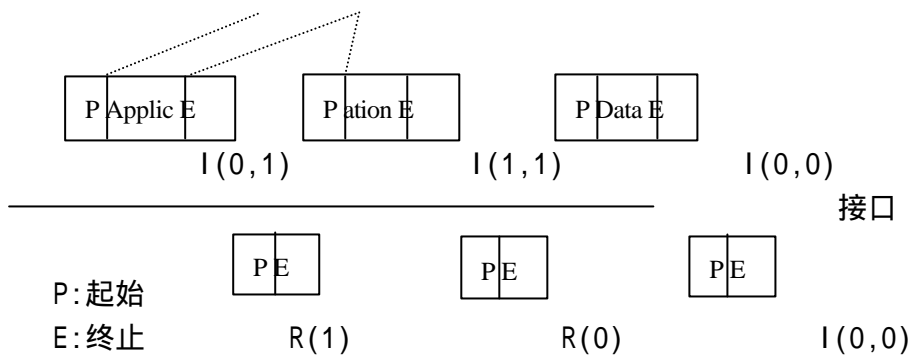


图23 一 链接功能

I块的链接由PCB中的M位(“多数数据位”)控制。M位指示一个I块的两种状态:

M=0表示没有与下一个块链接;

M=1表示链接了下一个块,且其为I块。

当接收方正确接收到多数数据I块时,它应发送 $R(N(R))$ ,其中 $N(R)$ 等于下一个I块的 $N(S)$

注:可在一个链中使用长度为0的I块。

#### 11.7.2.3 无差错操作协议准则

准则1:接口设备发送第1个块,该块或者为一个 $N(S)=0$ 的I块,表示为 $I(0,M)$ ,或者为一个S块。

准则2.1:由A发送的 $I(N_a(S),0)$ 被由B发送的 $I(N_b(S),M)$ 确认,以便传输应用数据并指明准备

接收从A来的下一个I块。

准则2.2:由A发送的 $I(N_a(S),1)$ 被由B发送的 $R(N_b(R))$ 确认[在这里 $N_b(R)$ 不等于 $N_a(S)$ ],以便

指明已接收的块是正确的,并且准备接收从A来的下一个I块。

注:在同一时间只能在一个方向链接。

准则3:若ICC要求更多的BWT处理前面已接收的I块,则发送一个S(WTX请求)。其中INF是一个

二进制整数位乘以BWT的值。接口设备由S(WTX响应)使用同一INF确认。分配的时间起

始为S(WTX响应)块最后一个字符的前沿。

准则4:ICC发送S(IFS请求)指明它能够支持一个新的IFSC,并且这次发送应被带有相同INF的

S(IFS响应)确认。当没有其它IFSC被另一个S(IFS请求)指明时,IFD认定新的IFSC有

效。IFD发送S(IFD请求)指明它能够支持一个新的IFSD,并且这一次发送应被具有相同

INF的S(IFD响应)确认。当不再有别的IFSD被另一个S(IFS请求)指明时,ICC认定新的

IFSD有效。

在这些S块的INF中IFSC和IFSD编码参见11.5.2.3。

准则5:由M位指示一链接,其中 $I(N(S),0)$ 是一个无链接的块或链接的最后一个块。 $I(N(S),1)$

是链接的一部分且后面至少链接一个块。

$R(N(R))$  请求传输下一个链接的 I 块,  $I(N(S)=N(R), \dots)$ , 并且确认已接收的链接 I 块

$I(NOT N(R), 1)$ .

### 11.7.3 差错处理

#### 11.7.3.1 由块的接收方检测差错

这个块的任务是传输块, 发现传输和顺序差错, 处理这些差错并使块传输协议再同步。因此块的数据链路层应能够处理下列差错。

一 BWT 超时: 某块的最后一个字符的上升沿和下一个块的第 1 个字符的上升沿之间的迟延超出 BWT。

一 接收无效块, 实例为:

- a) 在某块的一个或多个字符中有奇偶差错或 EDC 差错;
- b) PCB 无效(因为无法编码);
- c) LEN 无效(传输错、IFSC 或 IFSD 不兼容);
- d) 同步失效(当接收方预期接收的字符数超过了接收到的字符数而欠载, 就是当发方发送的字符超过了接收长度域中指定的值而过载);
- e) 发送相关的 S(... 请求) 之后接收 S(... 响应) 失败。

本协议的再同步应在三个连续类别上尝试。如果一个类别失效, 则在下一个类别上重试。对于 IFD 的这三个类别为:

- a) 块再传输;
- b) 使用 S(RESYNCH 请求);
- c) ICC 重置或启动。

对于 ICC 的这三个类别为:

- a) 块再传输;
- b) 使用 S(RESYNCH 响应);
- c) 没有被 IFD 启动, 卡无应答。

#### 11.7.3.2 差错处理协议准则

准则 6: S(RESYNCH 请求) 只能由 IFD 发送以达到再同步, 并且将块传输协议的通信参数复

位为该参数的初始值。

准则 6.1: 如果接收方发现同步失效, 在 I/O 上的沉寂时间大于 CWT 或 BGT (其中的较大者) 之后接收方恢复发送权。

准则 6.2: S(RESYNCH 请求) 应由来自 ICC 的 S(RESYNCD 响应) 响应。

准则 6.3: 在 IFD 接收到 S(RESYNCH 响应) 之后, 本协议起用。

准则 6.4: 当 IFD 为达到预期的再同步而连续发送 S(RESYNCH 请求) 最多连续失效三次之

后, 它就使 ICC 复位。

准则 6.5: 当接收 S(RESYNCH 请求) 时, 假定没有接收到早先发送的块。

准则 7.1: 当发送一个 I 块且接收块无效或(带有 IFD 的) BWT 超时发生时, 就发送一个 R 块, R

块带有其  $N(R)$  请求以使预期的 I 块  $N(S)=N(R)$ 。

准则7.2:当发送一个R块且接收块无效或(带有IFD的)BWT超时发生时,该R块被重发.

准则7.3:发送一个S(...请求)块且接收到的应答不是S(...响应)块或BWT超时(仅IFD)

发生时,该S(...请求)块重发.发送一个S(...响应)块且接收到的块无效或BWT

超时(仅IFD)发生时,就发送一个R块.

准则7.4.1:在协议的开始没有接收到一个无差错块时,IFD在使ICC重置或停滞之前最多

连续尝试两次.

准则7.4.2:本协议期间,如果IFD接收一个无差错的块失败,它在发送S(RESYNCH请求)之

前最多连续再试两次.

准则7.4.3:如果ICC在连续2次中的尝试之后没有收到一个无差错的块,则它保持接收模

式.

准则7.4.5:本协议开始之后,集成电路卡在接收第1个无效块时发送R(0)作出反应.

准则7.6.:如果IFD发送的第1个块没有在BWT期限内响应,则IFD发送R(0).

准则8:当ICC发送S(IFS请求)并接收到一个无效块时,为了产生一个S(IFS响应)ICC重发

最大多于1个的S(IFS请求)块.在第2次失效后它保持在接收模式.

准则9:链接故障应由链接的发送方或接收方发送一个S(ABORT请求)开始.该S(ABORT请

求)应由一个S(ABORT响应)来应答,随后是否能发送一个R块将依赖于它是否必要

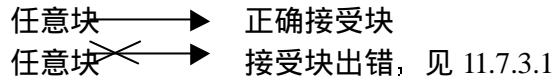
恢复发送权.

注:链接故障可能是由于ICC诸如集成电路卡记忆差错的物理差错.

附录 A  
T=1 的方案

A. 1 标注

本附录的目的是 11.7.1 标注的补充实例。

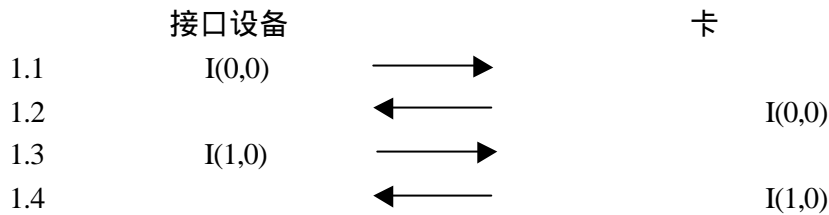


A. 2 无差错操作

[按照 11.7.2.3 的准则]

A.2.1 I 块的交换

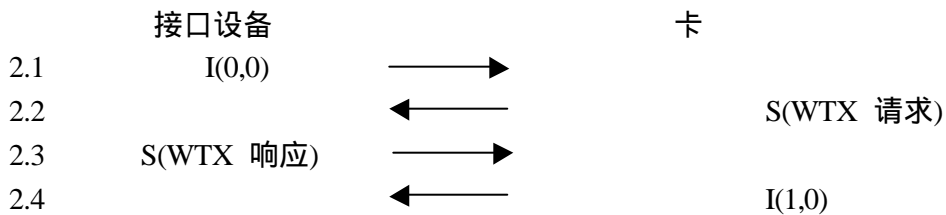
方案 1—— [准则 1 和 2.1]



A.2.2 等待时间扩展

方案 2—— [准则 3]

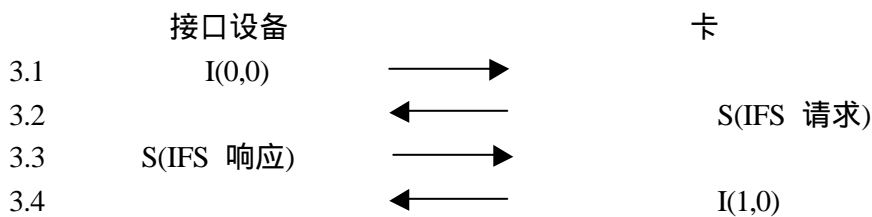
卡要求等待时间扩展。



A.2.3 IFS 调整

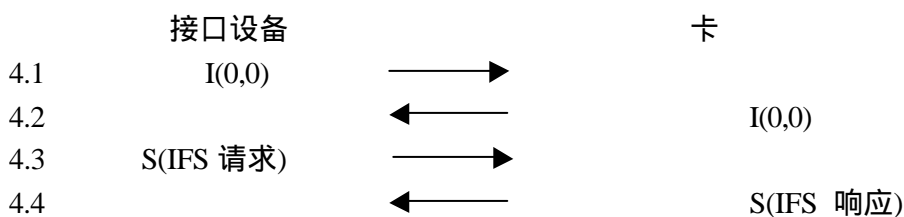
方案 3—— [准则 4]

卡启动 IFS 的调整。

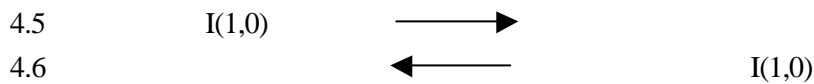


方案 4—— [准则 4]

接口设备启动 IFS 的调整。



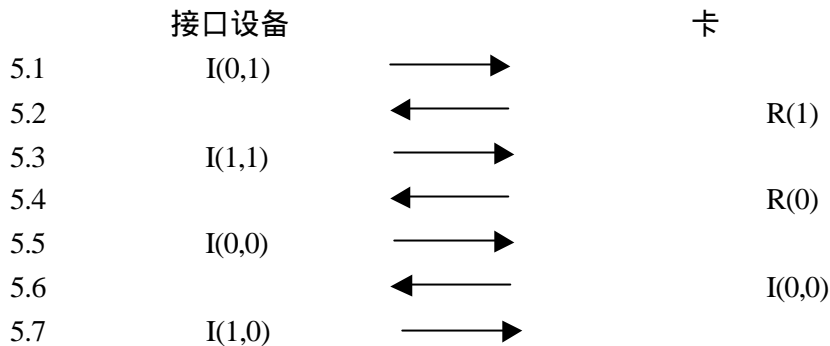




#### A.2.4 链接功能

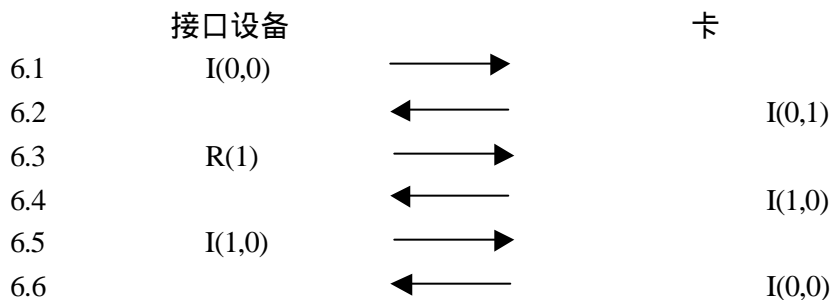
方案 5——〔准则 2.2 和 5〕

接口设备发送链接.



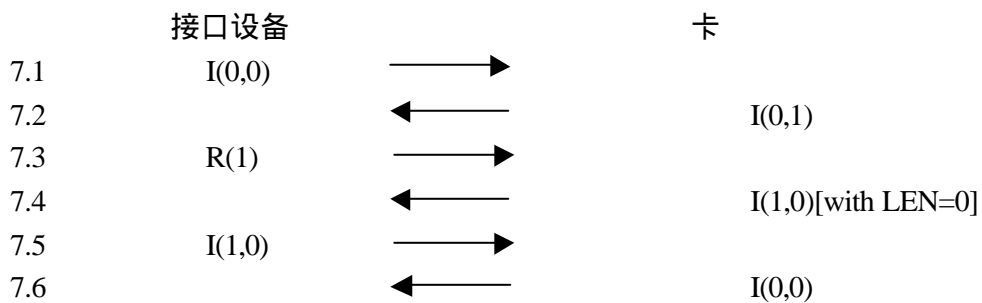
方案 6——〔准则 2.2 和 5〕

卡发送链接.



方案 7——〔见 9.7. 2. 2〕

卡使用 M 字节强制确认发送 I 块.

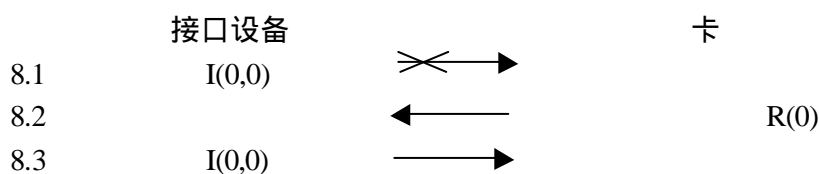


#### A. 3 差错处理

##### A.3.1 I 块的交换

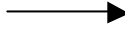
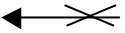
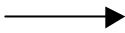

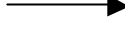
方案 8——〔准则 7.5〕

在协议的

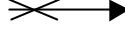
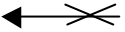
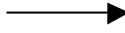
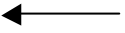
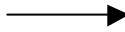
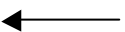
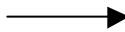


8.4  I(0,0)

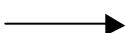
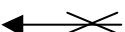


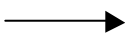
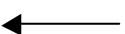
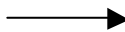
方案 9——〔准则 7. 1 和 7.6〕

	接口设备		卡
9.1	I(0,0)		
9.2			I(0,0)
9.3	R(0)		
9.4			I(0,0)
9.1	I(1,0)		

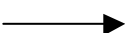
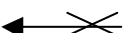
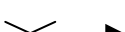
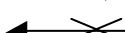

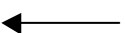
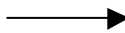
方案 10——〔准则 7. 1、7. 5 和 7.6〕

	接口设备		卡
10.1	I(0,0)		
10.2			R(0)
10.3	R(0)		
10.4			R(0)
10.5	I(0,0)		
10.6			I(0,0)
10.7	I(1,0)		

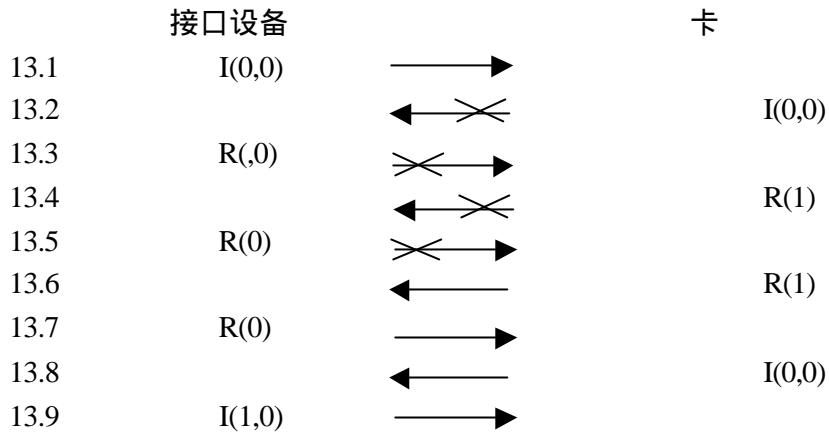
方案 11——〔准则 7. 1 和 7.6〕

	接口设备		卡
11.1	I(0,0)		
11.2			I(0,0)
11.3	R(0)		
11.4			R(1)
11.5	R(0)		
11.6			I(0,0)
11.7	I(1,0)		

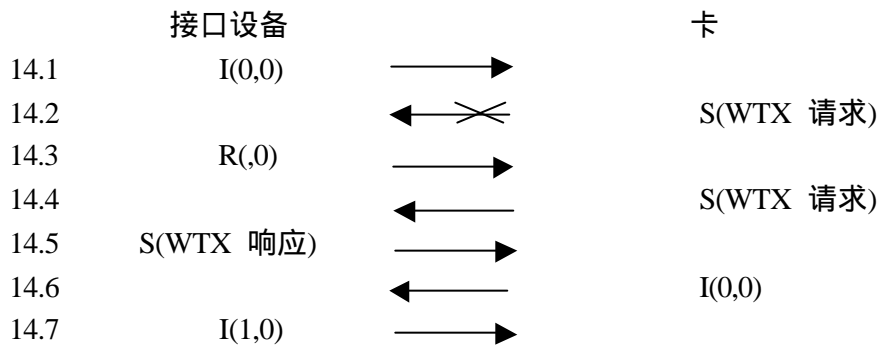
方案 12——〔准则 7. 1 和 7.6〕

	接口设备		卡
12.1	I(0,0)		
12.2			I(0,0)
12.3	R(0)		
12.4			R(1)
12.5	R(0)		
12.6			I(0,0)
12.7	I(1,0)		

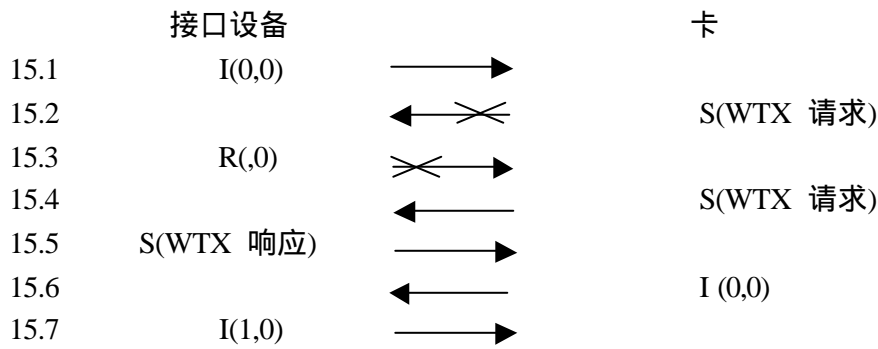
方案 13——〔准则 7. 1 和、7. 2 和 7.6〕



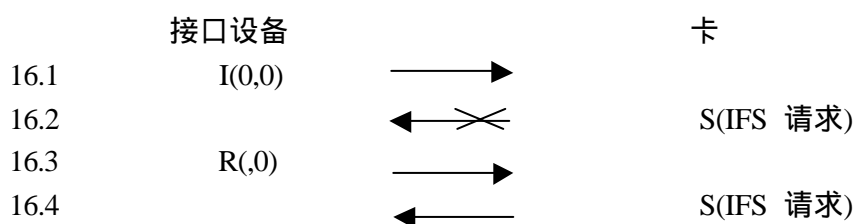
A. 3. 2 等待时间扩展  
 方案 14—〔准则 7.3〕  
 卡要求等待时间扩展.

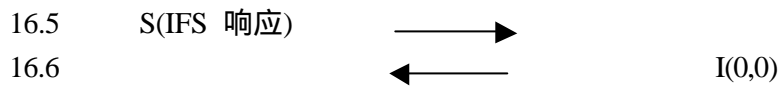


方案 15—〔准则 7.3〕  
 卡要求等待时间扩展.

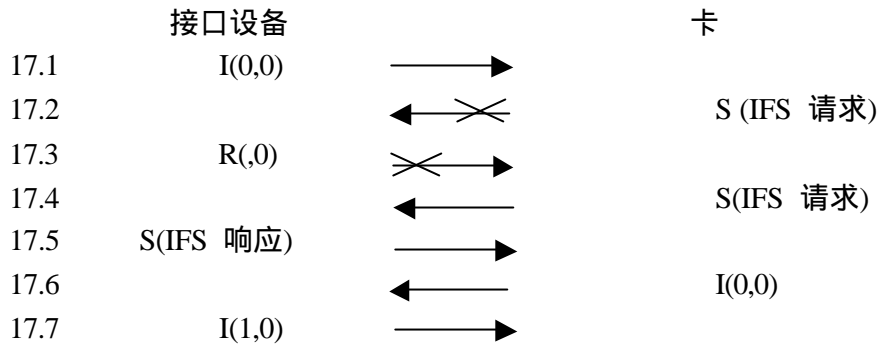


A. 3. 3 IFS 调整  
 方案 16—〔准则 7.3〕  
 卡要求 IFS 调整.

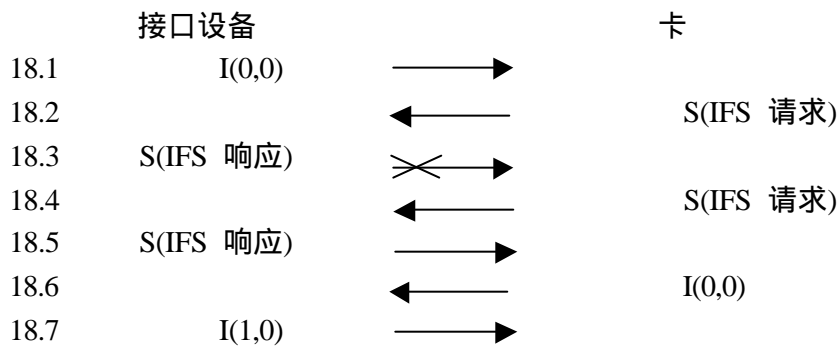




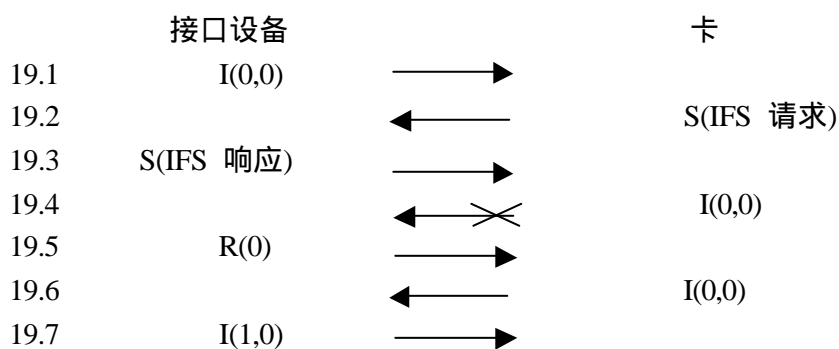
方案 17——〔准则 7.3〕  
卡要求 IFS 调整.



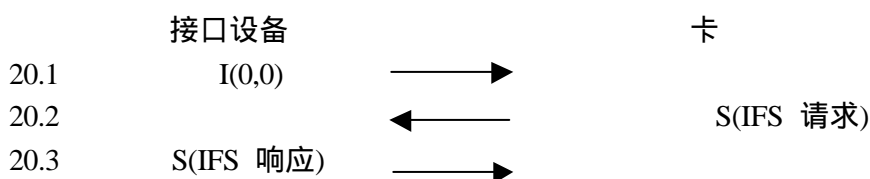
方案 18——〔准则 7.3〕  
卡要求 IFS 调整.



方案 19——〔准则 7.3〕  
卡要求 IFS 调整.



方案 20——〔准则 7.3〕  
卡要求 IFS 调整.



20.4			I(0,0)
20.5	R(0)		
20.6			R(1)
20.7	R(0)		
20.8			I(0,0)
20.9	I(1,0)		

A. 3. 4 链接功能

A. 3. 4. 1 接口设备发送链接.

方案 21——〔准则 7.1〕

	接口设备		卡
21.1	I(0,1)		
21.2			R(1)
21.3	R(0)		
21.4			R(1)
21.5	I(1,1)		
21.6			R(0)
21.7	I(0,0)		
21.8			I(0,0)
21.9	I(1,0)		

方案 22——〔准则 7.1〕

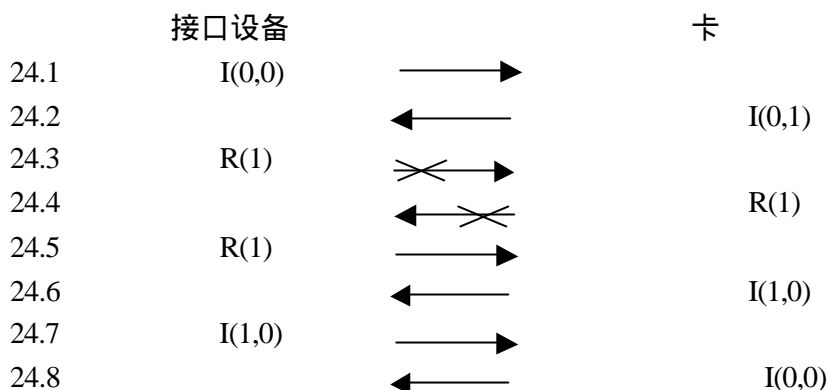
	接口设备		卡
22.1	I(0,1)		
22.2			R(1)
22.3	R(0)		
22.4			R(1)
22.5	I(1,1)		
22.6			R(0)
22.7	I(0,0)		
22.8			I(0,0)
22.9	I(1,0)		

A. 3. 4. 2 卡发送链接

方案 23——〔准则 7.1〕

	接口设备		卡
23.1	I(0,0)		
23.2			I(0,1)
23.3	R(1)		
23.4			R(1)
23.5	R(1)		
23.6			I(1,0)
23.7	I(1,0)		
23.8			I(0,0)

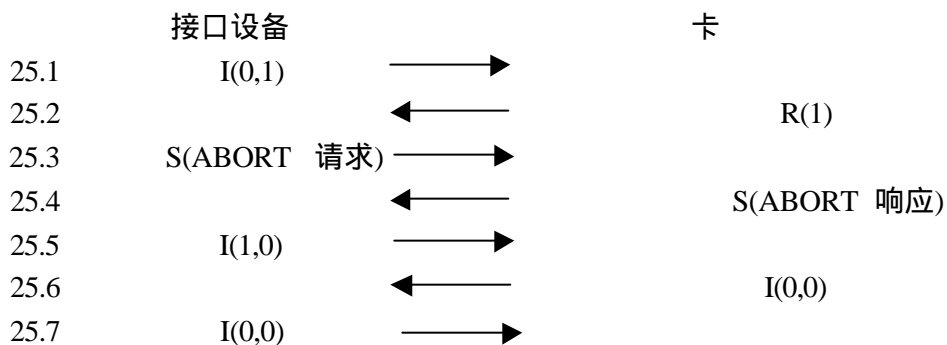
方案 24——〔准则 7.1〕



A. 3. 4. 3 链接发送方启动链接中止

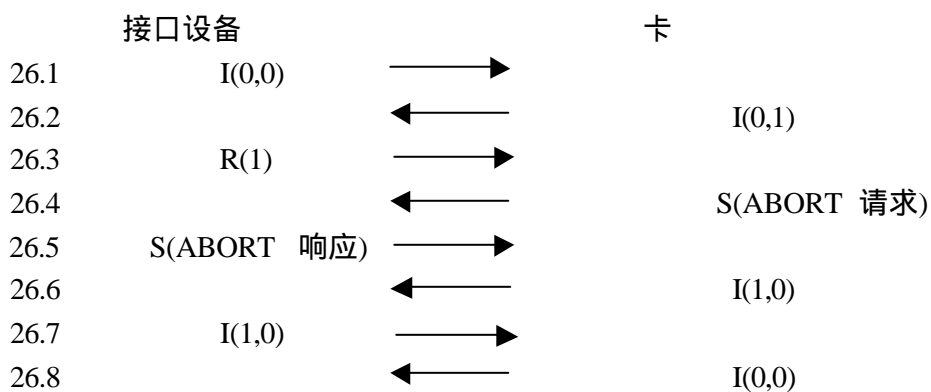
接口设备启动链接中止.

方案 25——〔准则 9〕



方案 26——〔准则 9〕

接口设备启动链接中止.

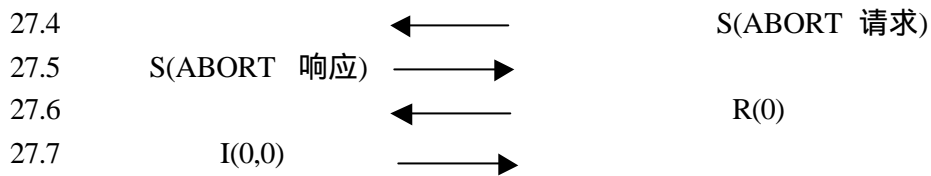


A. 3. 4. 4 链接方启动链接中止.

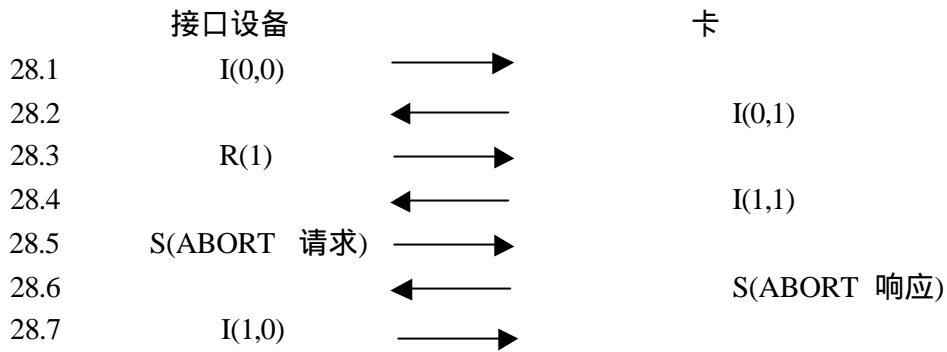
方案 27——〔准则 9〕

卡启动链接中止.



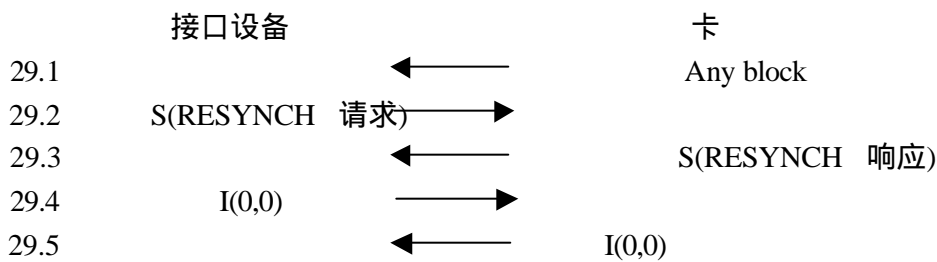


方案 28——〔准则 9〕  
卡启动链接中止。

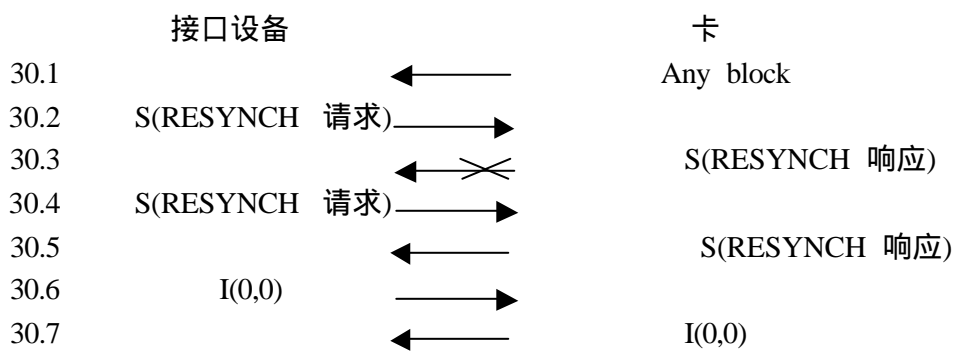


A. 3. 5 再同步

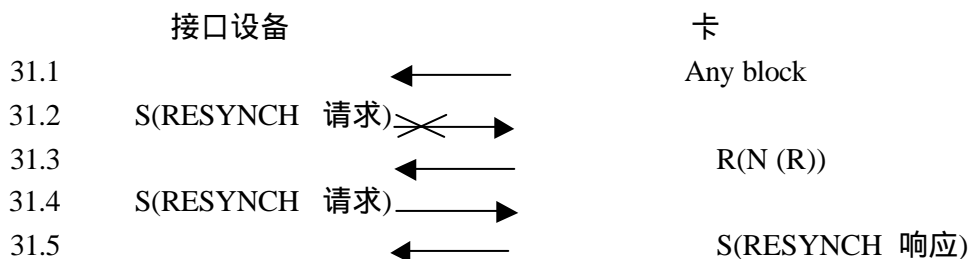
方案 29——〔准则 6.2〕

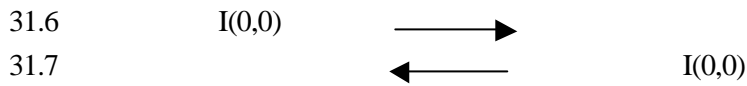


方案 30——〔准则 6.2 和 7.3〕

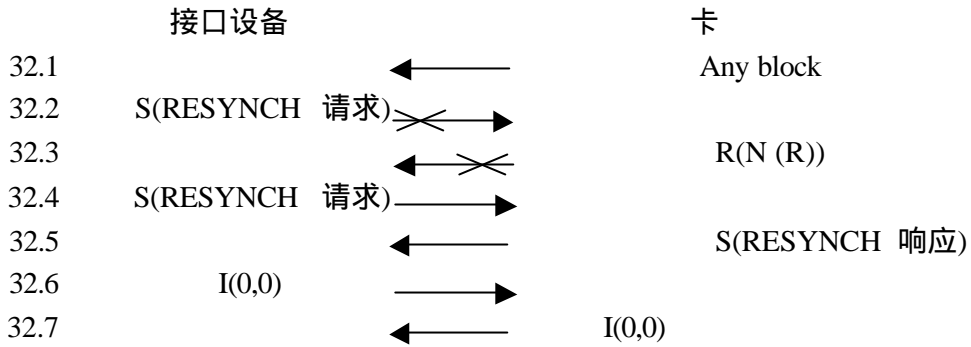


方案 31——〔准则 6. 2、7. 1 和 7. 3〕

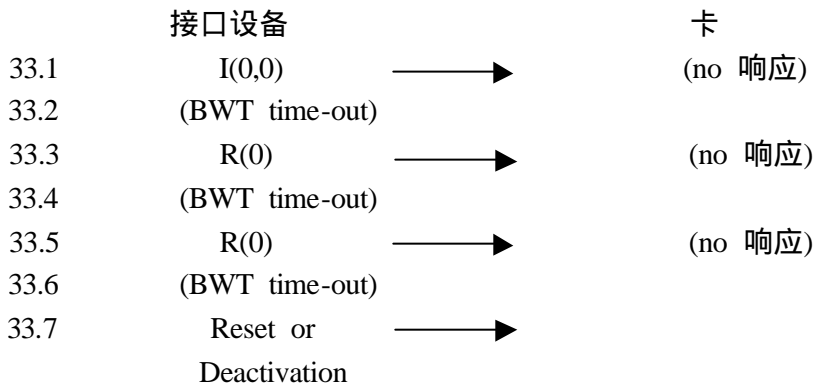




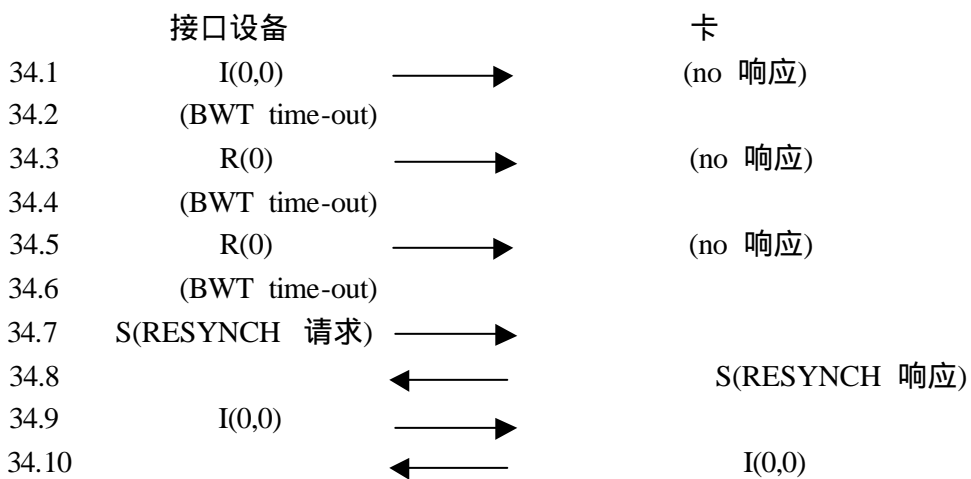
方案 32——



方案 33——〔准则 7. 1 和 7. 4. 1〕  
在协议启动时.



方案 34——〔准则 7. 1、7. 4. 2 和 7. 4. 3〕  
在协议期间.



方案 35——〔准则 6. 4、7. 1、7. 4. 2 和 7.4. 3〕  
在协议期间





35.1	I(0,0)	→	(no 响应)
35.2	(BWT time-out)		
35.3	R(0)	→	(no 响应)
35.4	(BWT time-out)		
35.5	R(0)	→	(no 响应)
35.6	(BWT time-out)		
35.7	S(RESYNCH 请求)	→	(no 响应)
35.8	(BWT time-out)		
35.9	S(RESYNCH 请求)	→	(no 响应)
35.10	(BWT time-out)		
35.11	S(RESYNCH 请求)	→	(no 响应)
35.12	(BWT time-out)		
35.13	Reset or Deactivation	→	

## 第二部分 行业间交换命令

## 目 录

1	范围 .....
2	参考文件 .....
3	定义 .....
4	缩略语和记录 .....
5	基本组织结构 .....
5.1	数据结构 .....
5.2	卡的安全体系结构 .....
5.3	APDU 报文结构 .....
5.4	命令首标、数据字段和响应尾标用的编码约定 .....
5.5	逻辑信道 .....
5.6	安全报文交换 .....
6	基本的行业间命令 .....
6.1	READ BINARY命令 .....
6.2	WRITE BINARY命令 .....
6.3	UPDATE BINARY命令 .....
6.4	ERASE BINARY命令 .....
6.5	READ RECORD命令 .....
6.6	WRITE RECORD命令 .....
6.7	APPEND RECORD命令 .....
6.8	UPDATE RECORD命令 .....
6.9	GET DATA 命令 .....
6.10	PUT DATA 命令 .....
6.11	SELECT FILE 命令 .....
6.12	VERIFY 命令 .....
6.13	INTERNAL AUTHENTICATE 命令 .....
6.14	EXTERNAL AUTHENTICATE 命令 .....
6.15	GET CHALLENGE 命令 .....
6.16	MANAGE CHANNEL 命令 .....
7	面向传输的行业间命令 .....
7.1	GET RESPONSE 命令 .....
7.2	ENVELOPE 命令 .....
8	历史字节 .....
9	与应用无关的卡服务 .....

## 附录

a	通过T=0传输APDU报文 .....
b	通过T=1传输APDU报文 .....
c	记录指针管理 .....
d	使用ANS.1基本编码规则 .....
e	卡轮廓的举例 .....
f	使用的安全报文交换 .....

### 1 范围

本规范规定了：

- 由接口设备至卡以及相反方向所发送的报文、命令和响应的内容；
- 在复位应答期间卡所发送的历史字节的结构及内容；
- 当处理交换用的行业间命令时，在接口处所看到的文件和数据的结构；
- 访问卡内文件和数据的方法；
- 定义访问卡内文件和数据的权利的安全体系结构；
- 安全报文交换的方法；
- 访问卡所处理算法的方法。本标准不描述这些算法。

### 2 参考文件

ISO3166:1993	国家名称表示的代码
ISO/IEC7812-4:1993	识别卡-发行者的标识-第1部分:编号系统
ISO/IEC7816-3:1997	识别卡-带触点的集成电路卡-第3部分:信号
	和传输协议
ISO/IEC7816-5:1994	识别卡-带触点的集成电路卡-第5部分:应用标识符
	的编号系统和登记规程
ISO/IEC7816-6	识别卡-带触点的集成电路卡-第6部分:行业间数据
	据元
ISO/IEC8825:1990	信息技术-开放系统互连-抽象语法记法1(ASN.1)的
	基本编码规则
ISO/IEC9796:1991	信息技术-安全技术-给出报文恢复的数字签名方案
ISO/IEC9797:1994	信息技术-安全技术-使用利用块密码算法

的“密码

检验”函数的数据完整性机制

ISO/IEC9979:1991

数据密码技术-密码算法登记规程

ISO/IEC10116-4:1994

信息技术-安全技术- n比特块密码算法的操作方式

ISO/IEC10118-4:1994

信息技术-安全技术-散列函数-第1部分:概述

分:概述

ISO/IEC10118-2

信息技术-安全技术-散列函数-第2部分:使用n比

特块密码算法的散列函数

特块密码算法的散列函数

### 3 定义

下列定义适用于本规范。

3.1 复位应答文件 Answer-to Reset file

表示卡操作特性的基本文件。

3.2 命令响应对 Command-response pair

两种报文的集合:命令后面紧跟着响应。

3.3 数据单元 data unit

可以无二义性地被引用的最少位集合。

3.4 数据元 data element

在接口处所看到的信息,为它定义了名称、逻辑内容描述、格式和编码。

3.5 数据对象 data object

在接口处所看到的信息,它由标签、长度和值(即,数据元)组成。在本部分规范中,数据对象称之为BER-TLV、压缩TLV和简单TLV数据单元。

3.6 专用文件 dedicated file

包含文件控制信息和任选地供分配用的存储器的文件。它可以是EFs和/或DFs的父辈。

3.7 DF名称 DF name

唯一地标识了卡内专用文件的字节串。

3.8 目录文件 directory file

ISO/IEC7816第5部分定义的基本文件。

3.9 基本文件 elementary file

共享同一文件标识符的数据单元或记录的集合。它不可能是另一文件的父辈。

3.10 文件控制参数 file control parameters

文件的逻辑、结构和安全的属性。

3.11 文件标识符 file identifier

用来寻址文件的2字节二进制值。

3.12 文件管理数据 file management data

除文件控制参数(例如,有效日期,应用标号)外,关于文件的任何信息。

3.13 内部基本文件 internal elementary file

用来存储由卡所解释数据的基本文件。

3.14 主文件 master file

表示文件结构根的强制性唯一专用文件。

3.15 报文 message

由接口设备向卡所发送的字节串，反之亦然，但不包括在ISO/IEC7816第3部分定义的面向传输的字符。

### 3.16 父辈文件 parent file

在分级结构范围内，直接在某一给定文件之前的专用文件。

### 3.17 口令 password

应用可以要求的数据，通过其用户将它呈现给卡。

### 3.18 路径 path

文件标识符的并置，而无需定界，如果路径以主文件的标识符开始，则它是一条绝对的路径。

### 3.19 提供者 provider

具有或曾获得在卡内建立专用文件权利的管理机构。

### 3.20 记录 record

可以由卡处理为一整体的并且可由记录号或记录标识符所引用的字节串。

### 3.21 记录标识符 record identifier

与记录相关的值，用来引用那个记录。在一个基本文件内几个记录可以具有相同的标识符。

### 3.22 记录号 record number

分配给每个记录的顺序号，它唯一地标识其基本文件内的记录。

### 3.23 工作的基本文件 working elementary file

用来存储不由卡所解释数据的基本文件。

## 4 缩略语和记号

下列缩略语适用于本部分规范。

APDU 应用协议数据单元

ATR 复位应答

BER ASN.1的基本编码规则(见附录D)

CLA 类别字节

DIR 目录

DF 专用文件

EF 基本文件

FCI 文件控制信息

FCP 文件控制参数

FMD 文件管理数据

INS 指令字节

MF 主文件

P1-P2 参数字节

PTS 协议类型选择

RFU 保留供将来使用

SM 安全报文交换

SW1-SW2 状态字节

TLV 标记、长度、值

TPDU 传输协议数据单元

下列记法适用于本部分规范。

“0”至“9”和“A”至“F” 16个十六进制数字

(B<sub>1</sub>) 字节(B<sub>1</sub>)的值

B<sub>1</sub> B<sub>2</sub> 字节B<sub>1</sub>(最高有效字节)和B<sub>2</sub>(最低有效字节)的并置

(B<sub>1</sub> B<sub>2</sub>) 字节B<sub>1</sub>和B<sub>2</sub>并置的值

# 编号

## 5 基本组织结构

### 5.1 数据结构

本条包含当处理交换用的行业间命令时在接口处所看到的关于数据逻辑结构的信息，超出本条概述之外的数据和结构信息的实际存储位置不在本部分规范范围内。

#### 5.1.1 文件组织结构

本部分规范支持下列两种文件。

—专用文件(DF)。

—基本文件(EF)。

卡内数据的逻辑组织结构由下列专用文件的结构化分级组成。

—在根处的DF称作主文件(MF)。该MF是必备的。

—其他DF是任选的。

定义了下列两种类型的EF。

—内部EF—那些EF预期用于存储由卡所解释的数据，即，为了管理和控制目的由卡所分析和使用的数据。

—工作的EF—那些EF预期用于不由卡所解释的数据，即，仅仅由外界待使用的数据。

图1示出了卡内逻辑文件组织结构的举例。

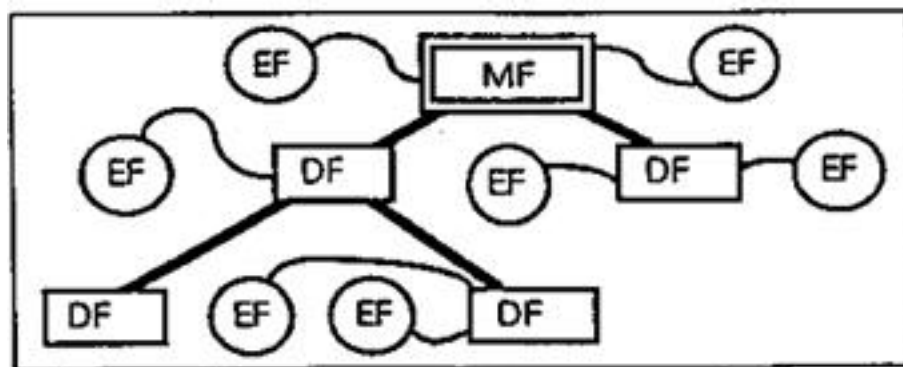


图1 逻辑文件组织结构(举例)

#### 5.1.2 文件引用方法

当文件不能被默认地选择时，应有可能至少通过下列方法之一来选择它。

—通过文件标识符引用—任何文件都可以通过按2字节编码的文件标识符来引用。如果MF通过文件标识符来引用，应使用“3F00”(保留值)。值“FFFF”被保留供将来使用。值“3FFF”被保留(见通过路径的引用)。为了通过文件标识符来选择无二义性的任何文件，直接在给定DF下的所有EF和DF都应具有不同的文件标识符。

—通过路径引用—任何文件都可以通过路径来引用(文件标识符的并

置)。该路径以MF或当前DF的标识符开始，并且以文件自身的标识符结束。在这两个标识符之间，路径由连续父辈DFs(如果有)的标识符组成。文件标识符的次序总是在父级至子级的方向上。如果当前DF的标识符未知，值‘3FFF’(保留值)可以用于路径的开始处。路径允许从MF或当前DF中无二义性地选择任何文件。

—通过短EF标识符引用—任何EF都可以通过值在从1至30范围内的5位编码的短EF标识符来引用。用作短EF标识符的值0引用了当前选择的EF。短EF标识符不能用在路径中或不能作为文件标识符(例如，在SELECT FILE 命令中)。

—通过DF名称引用—任何DF都可以通过按1至16个字节编码的DF名称来引用。为了通过DF名称进行无二义性的选择(例如，当借助ISO/IEC7816第5部分定义的应用标识符选择时)，每个DF名称应在给定的卡内是唯一的。

### 5.1.3 基本文件结构

定义了下列EF的结构。

—透明结构—在接口处EF可看作为一序列数据单元。

—记录结构—在接口处EF可看作为一序列各自可标识的记录。

为按记录构成的EF定义了下列属性。

—记录的长度:固定的或可变的。

—记录的组织结构:按顺序(线性结构)或者按环形(循环结构)。

为了构造EF，卡至少应支持下列四种方法之一。

—透明EF。

—带有固定长度记录的线性EF。

—带有可变长度记录的线性文件。

—带有固定长度记录的循环EF。

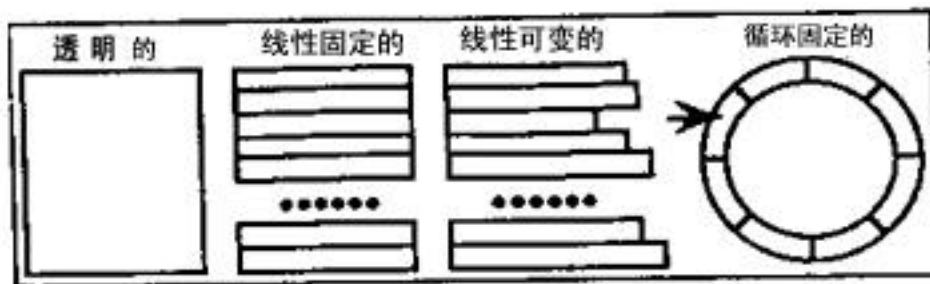


图2示出了这四种EF结构

图2 EF结构

注:图上的箭头引用了最当前写的记录。

### 5.1.4 数据引用方法

数据可以作为记录、数据单元或数据对象加以引用。数据可被认为是存储在单个连续序列记录(在记录结构的EF内)或者是存储在单个连续序列数据单元(在透明结构的EF内)。引用超出EF的记录或数据单元是一次差错。

数据引用方法、记录编号方法和数据单元长度都是与EF有关的特征。卡能在ATR、ATR文件和任何文件控制信息中提供指示。当卡在几个地方提供了指示时，对给定EF有效的指示就是在从MF至那个EF的路径范围内最接近那个EF的一个指示。

#### 5.1.4.1 记录引用

在每个记录结构的EF内，每个记录可以通过记录标识符和/或记录号来引用。



记录标识符和记录号都是带有值的在从‘01’至‘FE’范围内无符号8比特整数。值‘00’被保留用于特定目的。值‘FF’为RFU。

通过记录标识符引用应引起对记录指针的管理。卡的复位、选择文件和运载有效短EF标识符的任何命令都能影响记录指针。通过记录号引用应不影响记录指针。

—通过记录标识符引用—每个记录标识符由应用来提供。如果记录是在报文的数据字段中的简单TLV数据对象(见本部分规范5.4.4)，则记录标识符是数据对象的第1个字节。在记录结构的EF内，记录可以具有相同记录标识符，在此情况下，在记录中所包含的数据可以用来辨别这些记录。

每次使用记录标识符进行引用，一个指针应指定目标记录的逻辑位置：第1个或最后一个出现，下一个或先前一个出现都与记录指针有关。

—在每个线性结构的EF内，当写入或添加时，逻辑位置应有序地被分配，即按建立的次序。因此，第1个建立的记录是在第1个逻辑位置中。

—在每个循环结构的EF内，逻辑位置应按相反的次序来分配，即，最当前建立的记录是在第1个逻辑位置中。

为了线性结构和循环结构，定义下列附加规则。

—第1个出现应是带有规定标识符的记录，并是在第1个逻辑位置中；最后一个出现应是带有规定标识符的记录，并且是在最后一个逻辑位置中。

—当不存在当前记录时，下一个出现应等价于第1个出现；先前一个出现应等价于最后一个出现。

—当存在当前记录时，下一个出现应是带有规定标识符的最近记录，但是在比当前记录更大的逻辑位置中；先前一个出现应是带有规定标识符的最近记录，但是在比当前记录更小的逻辑位置中。

—值‘00’应按编号顺序表示第1个、下一个或先前一个记录，但与记录标识符无关。

—通过记录号引用—在每个记录结构的EF内，记录号是唯一的和顺序的。

—在每个线性结构的EF内，当写入或添加时，记录号应有序地被分配，即按建立的次序。因此，第1个记录(记录号1，#1)是第一个创建的记录。

—在每个循环结构的EF内，记录号应按相反的次序来分配，即，第1个记录(记录号1，#1)是最近建立的记录。

为了线性结构和循环结构，定义了下列附加规则。

—值‘00’应表示当前记录，即，通过记录指针所固定的那个记录。

#### 5.1.4.2 数据单元引用

在每个透明结构的EF内，每个数据单元可以通过偏移量(例如，在READ BINARY命令中，见本部分规范6.1)来引用。它是一个无符号整数，按照相应命令中的选项，它被限制在8位或15位。对于EF的第1个数据单元值为0，对于每个后续数据单元，偏移增加1。

通过默认，即，如果卡没有给出指示，则数据单元的长度为1个字节。

注：

1)记录结构的EF可以支持数据单元引用，在它支持的情况下，数据单元可以包含结构化信息以及数据，例如，线性结构中的记录号。

2)在记录结构的EF内，数据单元引用可以提供预期结果，因为在EF中的记

录存储次序未知，例如在循环结构中的存储次序。

#### 5.1.4.3 数据对象引用

每个数据对象(本部分规范5.4.4定义的)是以引用它的标记起头的，标记在ISO/IEC7816的本部分和其他部分进行规定。

#### 5.1.5 文件控制信息

文件控制信息(FCI)是可用于响应SELECT FILE 命令的数据字节串。对于任何文件，文件控制信息都可以呈现。

当文件控制信息编码为BER—TLV数据对象时，表1引入了预期用来运送文件控制信息的三种样板。

—FCP样板预期用来运送文件控制参数(FCP)，即，在表2中定义的任何BER—TLV数据对象。

—FMD样板预期用来运送文件管理数据(FMD)，即在本部分规范或本规范其他部分中规定的BER—TLV数据对象(例如，第5部分定义的应用标号以及第6部分定义的应用有效日期)。

—FCI样板预期用来运送文件控制参数和文件管理数据。

表1 与FCI相关的样板

标 记	值
'62'	文件控制参数(FCP样板)
'64'	文件管理数据(FMD样板)
'6F'	文件控制信息(FCI样板)

三种样板可以根据选择“SELECT FILE 命令”中的选项(见表59)进行检索。如果FCP或FMD选项被置位，则使用相应的样板是强制性的。如果FCI选项被设置，则使用FCI样板是任选的。

在应用的控制下，文件控制信息的一部分可以附加地存在于工作的EF中，并且可按照标签‘87’加以引用。对于编码的这种EF的文件控制信息，使用FCP或FCI样板是必备的。

不按照本部分规范编码的文件控制信息可以引入如下。

—‘00’或大于‘9F’的任何值—编码的后续字节串是专有的。

—标记=‘53’—数据对象的值字段由未按TLV编码的自由选定的数据组成。

—标记=‘73’—数据对象的值字段由自由选定的BER—TLV数据对象组成。

表2 文件控制参数

标记	L	值	适用于
'80'	2	在文件中的数据字节数，不包括结构信息	透明EFs
'81'	2	在文件中的数据字节数，如果有，包括结构信息	任何文件
'82'	1	文件描述符字节(见表3)	任何文件
	2	文件描述符字节后面紧跟着数据编码字节(见表86)	任何文件
	3 或 4	文件描述符字节后面紧跟着数据编码字节和最大记录长度	带有记录结构的EFs

'83'	2	文件标识符	任何文件
'84'	1 - 16	DF名称	DFs
'85'	变量	专有信息	任何文件
'86'	变量	安全属性编码超出本规范本部分的范围	任何文件
'87'	2	包含扩充FCI的EF标识符	任何文件
'88'		RFU	
'9E'			
'9FX Y'		RFU	

表3 文件描述符字节

b8 b7 b6 b5 b4 b3 b2 b1	含 义
0 x - - - - -	文件可访问性
0 0 - - - - -	不可共享的文件
0 1 - - - - -	可共享的文件
0 - x x x - - -	文件类型
0 - 0 0 0 - - -	工作的EF
0 - 0 0 1 - - -	内部的EF
0 - 0 1 0 - - -	保留供EFs的专有类型用
0 - 0 1 1 - - -	保留供EFs的专有类型用
0 - 1 0 0 - - -	保留供EFs的专有类型用
0 - 1 0 1 - - -	保留供EFs的专有类型用
0 - 1 1 0 - - -	保留供EFs的专有类型用
0 - 1 1 1 - - -	DF
0 - - - - x x	EF结构
x 0 - - - - 0 0	没有信息被给出
0 0 - - - - 0 0	透明
1 0 - - - - 0 1	线性固定，没有进一步的信息
0	

<u>0 - - - - 0 1</u>	<u>—线性固定，简单TLV</u>
<u>1</u>	
<u>0 - - - - 1 0</u>	<u>—线性可变，没有进一步的信息</u>
<u>0</u>	
<u>0 - - - - 1 0</u>	<u>—线性可变，简单TLV</u>
<u>1</u>	
<u>0 - - - - 1 1</u>	<u>—循环，没有进一步的信息</u>
<u>0</u>	
<u>0 - - - - 1 1</u>	<u>—循环，简单TLV</u>
<u>1</u>	
<u>1 x x x x x x x</u>	<u>RFU</u>
<u>“可共享”意味着至少支持在不同逻辑信道上的当前访问。</u>	

## 5.2 卡的安全体系结构

本条描述下列特征：

- 安全状态；
- 安全属性；
- 安全机制。

将安全属性与安全状态相比较，以执行命令和/或访问文件。

### 5.2.1 安全状态

安全状态表示完成下列动作后所获得的可能的当前状态：

- 复位应答(ATR)和可能的协议类型选择(PTS)和/或；
- 单个命令或一序列命令，可能执行的认证规程。

安全状态也可以从完成与所包含实体(如果有)的标识有关的安全规程中产生，例如，

- 通过证明了解口令(例如，使用一个VERIFY命令)；
- 通过证明了解密钥(例如，使用“GET CHALLENGE”命令后面紧接着“EXTERNAL AUTHENTICATE”命令)；
- 通过安全报文交换(例如，报文鉴别)。

考虑了三种安全状态：

—全局安全状态—它可以通过完成与MF相关的鉴别规程进行修改(例如，通过连接到MF的口令或密钥的实体鉴别)；

—文件特定安全状态—它可以通过完成与DF相关的鉴别规程进行修改(例如，通过连接到特定DF的口令或密钥的实体鉴别)；它可以通过文件选择进行维护、恢复或被丢失(见本部分规范6.10.2)；这种修改只与鉴别规程所属的应用相关；

—命令特定安全状态—仅在执行涉及使用安全报文交换(见本部分规范5.6)的命令期间，它才存在；这种命令可以保留未变化的其他安全状态。

如果逻辑信道的概念适用，则特定安全状态可以依赖于逻辑信道(见本部分规范5.5.1)。

### 5.2.2 安全属性

当安全属性存在时，它定义了允许的动作以及完成这种动作要执行的规程。安全属性可以与每个文件相关，并且安排为了允许对文件进行操作而应该满

足的安全条件。文件的安全属性依赖于：

—它的种类(DF或EF)；

—在它的文件控制信息中的和/或在它的父辈文件的文件控制信息中的任选参数。

注：安全状态也可以与其他对象(例如，密钥)相关。

### 5.2.3 安全机制

本规范本部分定义了下列安全机制：

—使用口令的实体鉴别——卡对从外界接收到的数据同保密的内部数据进行比较。该机制可以用来保护用户的权利。

—使用密钥的实体鉴别——待鉴别的实体必须按鉴别规程(例如，使用“GET CHALLENGE”命令后面紧跟着“EXTERNAL AUTHENTICATE”命令)来证明了解的相关密钥。

—数据鉴别——使用保密的或公开的内部数据，卡校验从外界接收到的冗余数据。另一种方法是使用保密的内部数据，卡计算数据元(密码的校验和或者数字签名)，并且将其插入发送给外界的数据中。该机制可以用来保护提供者的权利。

—数据加密——使用保密的内部数据，卡解密在数据字段中接收到的密文。另一种方法是，使用秘密的或公开的内部数据，卡计算密码，并将其插入数据字段中，尽可能与其他数据一起进行。该机制可以用来提供保密性服务，例如，用于密钥管理和有条件的访问。除了密码机制外，数据保密性可以通过数据伪装来获得。在此情况下，卡计算伪装字节串，并通过“异或”运算将其加到从外界接收到的数据字节中，或将其加到发送给外界的数据字节中。该机制可以用来保护秘密，并且减少报文过滤的可能性。

鉴别的结果可以按照应用的要求被登录到内部EF中。

## 5.3 APDU报文结构

应用协议中的一个步骤由发送命令、接收实体处理它以及发回的响应组成。因此，特定的响应对应于特定的命令，称作为**命令响应对**。

应用协议数据单元(APDU)可包含有命令报文或响应报文，它从接口设备发送到卡，或者相反地由卡发送到接口设备。

在命令响应对中，命令报文和响应报文都可以包含有数据，于是引起了由表4概括的四种情况。

表4 命令响应对内的数据

情况	命令数据	期望的响应数据
1	无数据	无数据
2	无数据	有数据
3	有数据	无数据
4	有数据	有数据

### 5.3.1 命令APDU

如图3所示(也见表6)，本规范本部分所定义的命令APDU由下列内容组成：

—必备的4字节首标(CLA INS P1 P2)；

—有条件的可变长度主体。

首标

主体



图3 命令APDU结构

在命令APDU的数据字段中呈现的字节数用L<sub>c</sub>来表示。

在响应APDU的数据字段中期望的字节最大数用L<sub>e</sub>(期望数据的长度)来表示。当L<sub>e</sub>字段只包含0时，则要求有效数据字节的最大数。

图4按照表4定义的4种情况示出了命令APDU的4种结构。

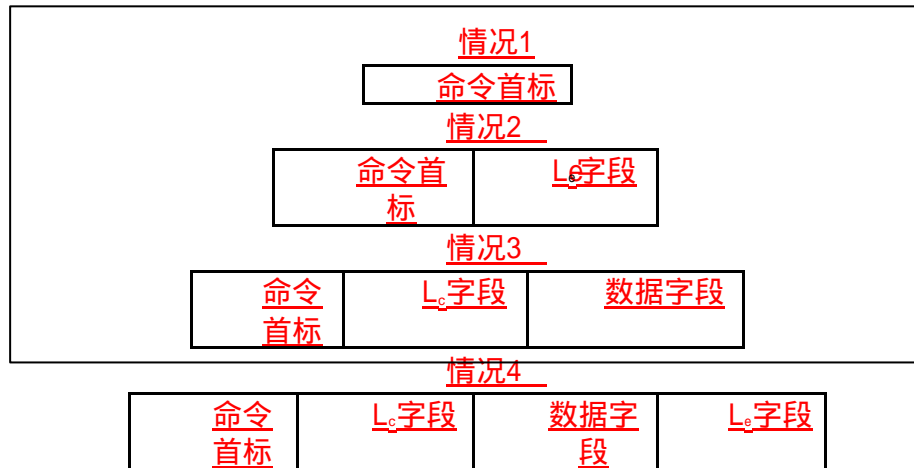


图4 命令APDU的4种结构

在情况1时，长度为空，因此L<sub>c</sub>字段和数据字段都为空。长度L<sub>e</sub>也为空；因此，L<sub>e</sub>字段为空。从而，主体为空。

在情况2时，长度L<sub>c</sub>为空；因此，L<sub>c</sub>字段和数据字段都为空。长度L<sub>e</sub>不为空；因此，L<sub>e</sub>字段存在。从而，主体由L<sub>e</sub>字段组成。

在情况3时，长度L<sub>c</sub>不为空；因此，L<sub>c</sub>字段存在，并且数据字段由L<sub>c</sub>后续字节组成。长度L<sub>e</sub>为空；因此，L<sub>e</sub>字段为空。从而，主体由L<sub>c</sub>字段后紧跟着数据字段组成。

在情况4时，长度L<sub>c</sub>不为空；因此，L<sub>c</sub>字段存在，并且数据字段由L<sub>c</sub>后续字节组成。长度L<sub>e</sub>也不为空；因此，L<sub>e</sub>字段也存在。从而主体由L<sub>c</sub>字段后紧跟着数据字段和L<sub>e</sub>字段组成。

### 5.3.2 命令主体用的解码约定

在情况1时，命令APDU的主体为空。这种命令APDU未运载长度字段。

在情况2、3和4时，命令APDU的主体由B<sub>1</sub>至B<sub>L</sub>所表示的L字节组成，如图5所示。这种主体运载了1或2长度字段；B<sub>1</sub>是第1个长度字段的一部分。

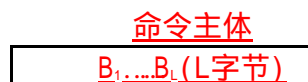


图5 不空的主体

在卡能力(见本部分规范8.3.6)中，在命令APDU内，卡说明了L<sub>c</sub>字段和L<sub>e</sub>字段既可为短的(一个字节、默认值)，也可为扩充的(显式语句)。

因此，情况2、3和4既可为短的(一个字节用于每个长度字段)也可为扩充的(B<sub>i</sub>的值为‘00’，并且每个长度值都按2个其他字节进行编码)。

表5示出了按照表4和图4中定义的四种情况及可能的L<sub>c</sub>、L<sub>e</sub>扩展的命令APDU的解码。

表5 命令APDU的解码

条 件	情 况	
$L=0$ — —	1	
$L=1$ — —	短的2	(2S)
$L=1+(B_1); (B_1) \neq 0; —$	短的3	(3S)
$L=2+(B_1); (B_1) \neq 0; —$	短的4	(4S)
$L=3; (B_1)=0; —$	扩充的2	(2E)
$L=3+(B_2 B_3); (B_1)=0; (B_2 B_3) \neq 0$	扩充的3	(3E)
$L=5+(B_2 B_3); (B_1)=0; (B_2 B_3) \neq 0$	扩充的4	(4E)

任何其他命令APDU为无效的。

$L_e$ 用的解码约定:

如果 $L_e$ 的值不为全空而按1个或2个字节进行编码, 则 $L_e$ 的值等于该字节的值, 它位于从1至255(或65 535)的范围内; 所有这些位的空值意味着 $L_e$ 的最大值为256(或65 536)。

前4种情况适用于所有卡。

情况1— $L=0$ ; 主体为空。

- 没有字节用于值为0的 $L_c$ 。
- 没有数据字节存在。
- 没有字节用于值为0的 $L_e$ 。

情况2S— $L=1$ 。

- 没有字节用于值为0的 $L_c$ 。
- 没有数据字节存在。
- $B_1$  编码值从1至255的 $L_e$ 。

情况3S— $L=1+(B_1)$ , 并且 $(B_1) \neq 0$ 。

- $B_1$  编码了值从1至255的 $L_c(\neq 0)$ 。
- $B_2 \sim B_{L-1}$  都是数据字段中的 $L_c$ 字节。
- 没有字节用于值为0的 $L_e$ 。

情况4S— $L=2+(B_1)$ , 并且 $(B_1) \neq 0$ 。

- $B_1$  编码了值从1至255的 $L_c(\neq 0)$ 。
- $B_2 \sim B_{L-1}$  都是数据字段中的 $L_c$ 字节。
- $B_L$  编码了从1至256的 $L_e$ 。

后3种情况也适用于指示扩充 $L_c$ 和 $L_e$ (见本部分规范8.3.6, 卡能力)的卡。

情况2E— $L=3$ , 并且 $(B_1)=0$

- 没有字节用于值为0的 $L_c$ 。
- 没有数据字节存在。
- $L_c$  字段由3个字节组成, 其中 $B_2$  和 $B_3$  编码了值从1至65 536的 $L_e$ 。

情况3E— $3+(B_2 B_3)$ ,  $(B_1)=0$ , 并且 $(B_2 B_3) \neq 0$ 。

- $L_c$  字段由前3个字节组成, 其中,  $B_2$ 和 $B_3$ 编码了值从1至65 535的 $L_c(\neq 0)$ 。
- $B_4$  至 $B_L$  都是数据字段中的 $L_c$ 字节。

· 没有字节用于值为0的 $L_c$ 。

情况4E— $L=5 + (B_2 \ B_3)$ ,  $(B_1)=0$ , 并且 $(B_2 \ B_3) \neq 0$ 。

·  $L_c$ 字段由前3个字节组成, 其中,  $B_2$ 和 $B_3$ 编码了值从1至65 535的 $L_c$  ( $\neq 0$ )。

·  $B_4$ 至 $B_{L-2}$ 都是数据字段中的 $L_c$ 字节。

·  $L_c$ 字段由最后的2个字节 $B_{L-1}$ 和 $B_L$ 组成; 它们编码了值从1至65 536的 $L_c$ 。

对于本规范本部分定义的每个传输协议, 附属到本部分的附录(每个协议一个)规定了先前7种情况中的每一种用的运输APDU的命令响应对。

### 5.3.3 响应APDU

如图6所示(也见表7), 本规范本部分定义的响应APDU由下列内容组成:

—有条件的可变长度主体;

—必备的2字节尾标(SW1 SW2)。



图6 响应APDU结构

在响应APDU的数据字段中呈现的字节数用 $L_r$ 来表示。

尾标编码了处理“命令响应对”之后的接收实体的状态。

注: 如果该命令被放弃, 则响应APDU是一个尾标, 它按2个状态字节来编码差错条件。

## 5.4 命令首标、数据字段和响应尾标用的编码约定

表6示出了命令APDU的内容

表6 命令APDU内容

代码	名称	长度	描述
CLA	类别	1	指令的类别
INS	指令	1	指令代码
P1	参数1	1	指令参数1
P2	参数2	1	指令参数2
$L_c$ 字段	长度	变量1或	在命令的数据字段中呈现的字节数 在命令的数据字段中发送的字节串 在向命令响应的数据字段中期望的字节最大数
数据字段	数据	3	
$L_r$ 字段	长度	变量= $L_c$ 变量 3	

表7示出了响应APDU的内容

表7 响应APDU内容

代码	名称	长度	描述
数据字段	数据	变量 = $L_r$	在响应的数据字段中收到的字节串
SW1	状态字节 1	1	命令处理状态
SW2	状态字节 2	1	命令处理受限字符

后续条规定了类别字节、指令字节、参数字节、数据字段字节和状态字节用的编码约定。



除非另有规定，在这些字节中，RFU的比特都编码为0，并且RFU字节也都编码为‘00’。

#### 5.4.1 类别字节

按照与表9一起使用的表8，命令中的类别字节CLA用来指出：

- 命令和响应在什么程度上应遵循本规范本部分；
- 当适用(见表9)时，安全报文交换的格式及逻辑信道号。

表8 CLA的编码及含义

值	含 义
‘0X’	按照本规范定义 命令和响应的结构和编码(对于编码‘X’ 见表9)。
‘10’ - ‘7F’	RFU
‘8X’ ‘9X’	按本规范定义命令和响应的结构，‘X’除外(对于‘X’ 编码 见表9) 命令和响应的编码及含义是专有的。
‘AX’	除非通过应用上下文另有规定 按照本规范定义命令和响应的 编码及含义(对于编码‘X’ 见表9)。
‘B0’ - ‘CF’	按照本规范本部分 命令和响应的结构
‘D0’ - ‘FE’	命令和响应的专有结构
‘FF’	保留供PTS用

表9 当CLA=‘0X’、‘8X’、‘9X’或‘AX’时，半字节‘X’的编码及含义

b4 b3 b2 b1	含 义
x x — —	安全报文(SM)格式
0 x — —	· 没有SM或SM不按照5.6
0 0 — —	—没有SM或没有SM指示
0 1 — —	—专有的SM格式
1 X — —	· 安全报文交换按照5.6
1 0 — —	—不被鉴别的命令首标
1 1 — —	—被鉴别的命令首标 (关于命令首标的用法见5.6.3.1)
— — x x	逻辑信道号(按照5.5) (当不使用逻辑信道号时或当逻辑信 道#0被选择时，b2 b1=00)

#### 5.4.2 指令字节

命令中的指令字节INS应予以编码，以便允许使用本规范第3部分定义的任何协议进行传输。表10示出了必然无效的ISN代码

表10 无效的INS代码

b8 b7 b6 b5 b4 b3 b2 b1	含 义
x x x x x x x 1	—奇数值
0 1 1 0 x x x x	—‘6X’
1 0 0 1 x x x x	—‘9X’

表11示出了本规范定义的INS代码，当CLA的值位于从‘00’至‘7F’的范围内时，INS代码的其他值有待ISO/IEC JTC1 SC17进行分配。

表11 本规范定义的INS代码

值	命令名称	条款
‘0E’	ERASE BINARY	6.4
‘20’	VERIFY	6.12
‘70’	MANAGE CHANNEL	6.16
‘82’	EXTERNAL AUTHENTICATE	6.14
‘84’	GET CHALLENGE	6.15
‘88’	INTERNAL AUTHENTICATE	6.13
‘A4’	SELECT FILE	6.11
‘B0’	READ BINARY	6.1
‘C0’	GET RESPONSE	7.1
‘C2’	ENVELOPE	7.2
‘CA’	GET DATA	6.9
‘D0’	WRITE BINARY	6.2
‘D2’	WRITE RECORD	6.6
‘D6’	UPDATE BINARY	6.3
‘DA’	PUT DATA	6.10
‘DC’	UPDATE RECORD	6.8
‘E2’	APPEND RECORD	6.7

#### 5.4.3 参数字节

命令中的参数字节P1-P2可以具有任何值，如果参数字节不提供进一步的限定，则它应置为‘00’。

#### 5.4.4 数据字段字节

每个数据字段应具有下列三种结构之一：

- 每个TLV编码的数据字段应由一个或多个TLV编码的数据对象组成；
- 每个非TLV编码的数据字段应按照相应命令的规范由一个或多个数据元组成；
- 专用编码的数据字段结构在本规范中不予规定。

本规范支持在数据字段中的下列两种类型的TLV编码的数据对象：

- BER-TLV数据对象；
- 简单TLV数据对象。

本规范不使用‘00’或‘FF’作为标记值。

每个BER-TLV数据对象应由2个或3个连续的字段(见ISO8825和附录D)组成。

- 标记字段T由一个字节或多个连续字节组成，它编码了类别、类型和编号。
- 长度字段由一个字节或多个连续字节组成，它编码了整数L。
- 如果L不为空，则值字段V由L个连续字段组成，如果L为空，则数据对象为空：不存在值字段。

每个简单TLV数据对象应由2个或3个连续字段组成。

- 标记字段T由单个字节组成，从1至254中的一个编号(例如，一个记录标识符)。它对类别和结构类型不进行编码。

——长度字段由1个字节或3个连续字节组成。如果长度字段的首字节处于从‘00’至‘FE’的范围内，则长度字段由单个字节组成，该字节编码从0至254中的一个整数L。如果首字节等于‘FF’，则长度字段后续2个字节使用从0至65535中的值编码了一个整数L。

——如果L不为空，则值字段V由L个连续字节组成。如果L为空，则数据对象为空：不存在有效字段。

某些命令(例如，SELECT FILE)的数据字段，简单TLV数据对象的值字段和某些原始BER-TLV数据对象的值字段都预期用于编码一个或多个数据元。

某些其他命令(例如，面向记录的命令)的数据字段、其他原始BER-TLV数据对象的值字段都预期用于编码一个或多个简单TLV数据对象。

某些其他命令(例如，面向对象的命令)的数据字段和结构化BER-TLV数据对象的值字段都预期用于编码一个或多个BER-TLV数据对象。

注：在TLV编码的数据对象之前、之间或之后，无任何含义‘00’或‘FF’字节可以出现(例如，由于擦除的或修改的TLV编码的数据对象引起的)。

#### 5.4.5 状态字节

响应的状态字节SW1-SW2表示了卡内的处理状态。图7示出了本规范本部分定义的结构方案。

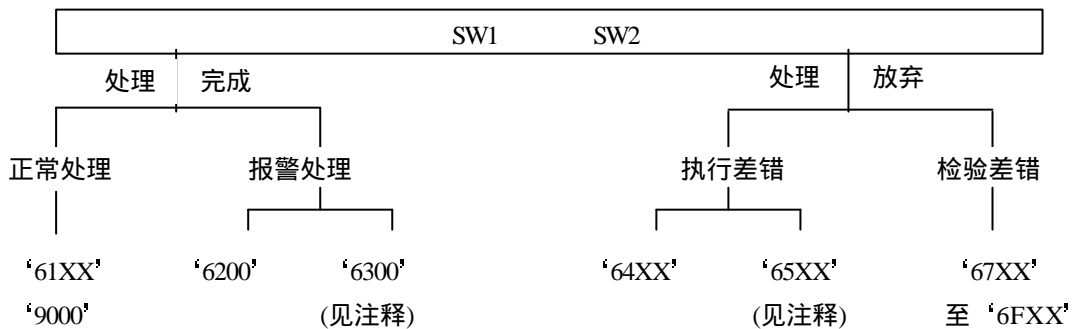


图7 状态字节的结构方案

注：当SW1=‘63’或‘65’时，非易失存储器的状态变化。当SW1=除‘63’和‘65’外的‘6X’时，非易失存储器的状态不变化。

由于本规范本部分的规定的原因，本部分不定义SW1-SW2的下列值：

——‘60XX’；

——如果‘XX’=‘00’，在每种情况下，‘67XX’，‘6BXX’，‘6DXX’，‘6EXX’，‘6FXX’；

——如果‘XXX’=‘000’，‘9XXX’。

无论哪个协议被使用，SW1-SW2的下列值要予以定义(见附录A的举例)。

——如果使用响应(其中SW1=‘6C’)来中途停止命令，当在发出任何其他命令之前重新发出同一命令，则SW2指示将该值给予短的L<sub>0</sub>字段(被请求数据的准确长度)。

——如果使用响应(其中SW1=‘61’)来处理命令(它可以是情况2或4，见表4和图4)，则在发出任何其他命令之前发出的GET RESPONSE命令中，SW2指示将最大值给予短的L<sub>0</sub>字段(额外数据长度仍然有效)。

注：类似于由‘61XX’所提供的功能可以在应用级上通过‘9FXX’来提供。

通过表13-18所完成的表12示出了本规范本部分定义的SW1-SW2值的一般含义。对于每个命令，相应的条款提供了更详细的含义。

当SW1的值为‘62’，‘63’，‘65’，‘68’，‘69’和‘6A’时，表13-18规定了SW2的值。除了本规范本部分不定义的从‘F0’至‘FF’值之外，表13至表18不定义的SW2值都是RFU。

表12 SW1-SW2的编码

SW1-SW2	含 义
‘9000’ ‘61XX’	<u>正常的处理</u> -无进一步限定 -SW2指示仍然有效的响应字节数 (见下面文本)
‘62XX’ ‘63XX’	<u>报警处理</u> -非易失存储器状态不变化 (在SW2中进一步的限定，见表13) -非易失存储器状态变化 (在SW2中进一步的限定，见表14)

续表12 SW1-SW2的编码

‘64XX’ ‘65XX’	<u>执行差错</u> -非易失存储器状态不变化 (SW2=‘00’，其他值都是RFU) -非易失存储器状态变化 (在SW2中进一步的限定，见表15)
‘66XX’	-保留供安全相关的发布使用 (本规范本部分不定义)
‘6700’ ‘68XX’ ‘69XX’ ‘6AXX’ ‘6B00’ ‘6CXX’ ‘6D00’ ‘6E00’ ‘6F00’	<u>校验差错</u> -错误的长度 -GLA的功能不被支持 (在SW2中进一步的限定，见表16) -不允许的命令 (在SW2中进一步的限定，见表17) -错误的参数P1-P2 (在SW2中进一步的限定，见表18) -错误的参数P1-P2 -错误的长度L <sub>0</sub> : SW2指示准确的长度 (见下面的文本) -指令代码不被支持或无效 -类别不被支持 -没有精确的诊断

表13 当SW1=‘62’时，SW2的编码

SW2	含 义
‘00’	没有信息被给出
‘81’	返回数据的一部分可能被损坏
‘82’	读出L <sub>0</sub> 字节之前，文件/记录已结束
‘83’	选择的文件无效
‘84’	FCI未按照5.1.5格式化

表14 当SW1=‘63’时，SW2的编码

<u>SW2</u>	<u>含 义</u>
<u>'00'</u>	<u>没有信息被给出</u>
<u>'81'</u>	<u>通过最后写入来填满文件</u>
<u>'CX'</u>	<u>通过'X' (值从0至15)提供的计数器 (正确的含义依赖于命令)</u>

表15 当SW1='65'时, SW2的编码

<u>SW2</u>	<u>含 义</u>
<u>'00'</u>	<u>没有信息被给出</u>
<u>'81'</u>	<u>存储器故障</u>

表16 当SW1='68'时, SW2的编码

<u>SW2</u>	<u>含 义</u>
<u>'00'</u>	<u>没有信息被给出</u>
<u>'81'</u>	<u>逻辑信道不被支持</u>
<u>'82'</u>	<u>安全报文不被支持</u>

表17 当SW1='69'时, SW2的含义

<u>SW2</u>	<u>含 义</u>
<u>'00'</u>	<u>没有信息被给出</u>
<u>'81'</u>	<u>命令与文件结构不兼容</u>
<u>'82'</u>	<u>安全状态不被满足</u>
<u>'83'</u>	<u>认证方法被阻塞</u>
<u>'84'</u>	<u>引用的数据无效</u>
<u>'85'</u>	<u>使用的条件不被满足</u>
<u>'86'</u>	<u>命令不被允许(无当前EF)</u>
<u>'87'</u>	<u>期望的SM数据对象失踪</u>
<u>'88'</u>	<u>SM数据对象不正确</u>

表18 当SW1='6A'时, SW2的编码

<u>SW2</u>	<u>含 义</u>
<u>'00'</u>	<u>没有信息被给出</u>
<u>'80'</u>	<u>在数据字段中的不正确参数</u>
<u>'81'</u>	<u>功能不被支持</u>
<u>'82'</u>	<u>文件未找到</u>
<u>'83'</u>	<u>记录未找到</u>
<u>'84'</u>	<u>无足够的文件存储空间</u>
<u>'85'</u>	<u>L。与TLV结构不一致</u>
<u>'86'</u>	<u>不正确的参数P1-P2</u>
<u>'87'</u>	<u>L。与P1-P2不一致</u>
<u>'88'</u>	<u>引用的数据未找到</u>

## 5.5 逻辑信道

### 5.5.1 一般概念

在接口处看到的逻辑信道作为与DF的逻辑链路进行工作。

在一个逻辑信道上应存在独立的活动，而与另一个信道上的活动无关，也就是说，在一个逻辑信道上的命令相互关系应独立于另一个逻辑信道上的命令相互关系。然而，逻辑信道可以共享与应用相关的安全状态，因此，可以具有与安全有关的跨越逻辑信道的命令相互关系(例如，命令VERIFY)。

提供给某一逻辑信道的命令运载了在CLA字节中的相应逻辑信道号(见表8和9)。逻辑信道的编号从0至3。如果卡支持逻辑信道机制，则有效逻辑信道的最大编号可以在卡能力中指出(见本部分规范8.3.6节)命令响应对按当前描述的那样进行工作。

本规范本部分仅支持在启动后续命令响应对之前应完成的命令响应对。跨越逻辑信道应该没有命令及其响应的交错；在收到命令与发送响应给该命令之间，只有一个逻辑信道是活动的，当逻辑信道被开放时，它保持开放，直到由MANAGE CHANNEL 命令显式地关闭为止。

注：

1)如果不排除，对同一DF，可以开放一个以上的逻辑信道(见5.1.5文件可访问性)。

2)如果不排除，一个以上的逻辑信道可以选择同一EF(见5.1.5文件可访问性)。

3)在逻辑信道上的SELECT FILE 命令将开放当前DF及可能的当前EF。因此，每个逻辑信道有一个当前DF及可能的一个当前EF作为SELECT FILE 命令行为的结果以及使用短EF标识符访问命令的文件。

#### 5.5.2 基本逻辑信道

基本逻辑信道永久有效。当被编号时，它的编号为0。当类别字节按照表8和9进行编码时，位1和2编码了逻辑信道号。

#### 5.5.3 打开逻辑信道

逻辑信道可通过成功地完成下列内容来打开：

——通过分配类别字节中的大于0的逻辑信道号来完成引用的DF的SELECT FILE 命令；

——或者，完成MANAGE CHANNEL 命令的开放功能，该命令分配在命令APDU中的0以外的逻辑信道号或请求卡分配的和响应中返回的逻辑信道号。

#### 5.5.4 关闭逻辑信道

MANAGE CHANNEL 命令的关闭功能可以用来显式地使用逻辑信道号关闭逻辑信道。关闭之后，逻辑信道号可供重新使用，基本逻辑信道应不予关闭。

### 5.6 安全报文交换

安全报文交换的目的是通过确保两种基本安全功能来保护往返于卡的报文部分，这两种功能是：数据鉴别和数据安全性。

安全报文交换可通过应用一种或多种安全机制来获得，每种安全机制都涉及算法、密钥、自变量、经常还有初始数据。

· 对于安全机制的执行，数据字段的发送和接收可以交错进行。本规范不妨碍通过顺序地分析哪些机制和哪些安全项目应该用于处理数据字段的其余部分所作的决定。

· 两种或两种以上的安全机制可以使用带有不同操作方式的相同算法(见

ISO10116)。填充规则的现有规范不排除这种特征。

本条定义了SM相关数据对象的三种类型：

- 普通值数据对象，预期用来运载普通数据；
- 安全机制数据对象，预期用来运载安全机制的计算结果；
- 辅助安全数据对象，预期用来运载控制引用和响应描述符。

#### 5.6.1 SM格式概念

在涉及基于密码安全机制的每个报文中，数据字段应符合ASN.1的基本编码规则(见ISO8825和附录D)，除非通过类别字节另有指示(见本部分规范5.4.1)。

在数据字段中，可以选择现存的SM格式：

- 隐式地选择，即，在发出命令之前已知；
- 显式地选择，即，通过类别字节来固定(见表9)。

本规范本部分定义的SM格式是BER-TLV编码的。

- 上下文特定的标签类别(范围从‘80’至‘BF’)被保留供SM用。
- 其他类别的数据对象可以呈现(例如，应用特定类别的数据对象)。
- 某些与SM相关的数据对象是递归的：它们的普通值字段仍然是BER-TLV编码的，因此上下文特定的类别自然是被保留供SM用。

在上下文特定类别中，标签的位1决定了SM相关的数据对象会(b1=1)或不会(b1=0)集成到认证用的数据对象的计算中。如果呈现，其它类别数据对象会集成到该计算中。

#### 5.6.2 普通值数据对象

对于不按BER-TLV编码的数据以及对于包括与SM相关的数据对象的BER-TLV，ENVELOPE都是强制性的。对于不包括与SM相关的数据对象的BER-TLV，ENVELOPE是任选的。表19示出了ENVELOPE用的普通值数据对象。

表19 普通值数据对象

标 记	值
	简明值由下列内容组成
‘B0’，‘B1’	—BER-TLV 包括与SM相关的数据对象
‘B2’，‘B3’	—BER-TLV 但不包括与SM相关的数据对象
‘80’，‘81’	—不是BER-TLV编码的数据
‘99’	—SM状态信息(例如 SW1-SW2)

#### 5.6.3 认证用的数据对象

##### 5.6.3.1 密码“校验和”数据对象

密码校验和的计算(见ISO9797)包含有初始校验块、密钥以及不可逆密码算法块。

在相关密钥的控制下，该算法在本质上将现行的K字节(典型地是8或16)输入块变换成现行的相同长度输出块。

密码校验和的计算按下列连续步骤执行：

- 初始步骤—初始步骤设置下列块之一的初始校验块：
  - 空块，即，K字节值为‘00’

· 链接块，即，由先前计算命令(先前命令的最后一校验块)和响应(先前响应的最后一校验块)的结果，例如，由外界提供的初始值块；

· 按照相关密钥从变换辅助数据产生的辅助块。如果辅助数据小于K字节，则它通过置为0的位为起头，直至块长度。

—相继步骤—当表9可用(CLA='0X'，'8X'，'9X'或'AX')时，如类别字节的位b4和b3置为1，则第1个数据、块由命令APDU(CLA INS P1 P2)的首标后随值为'80'的一个字节和值为'00'的5个字节的K组成。

密码“校验和”应集成具有标记(b1=1)的任何SM相关数据对象和带有标记超出范围'80'至'BF'的任何数据对象，这些数据对象应通过数据块集成到当前的校验块中。分解数据块应按下列方法进行：

—分块应在被集成的相邻数据对象之间的边界处继续进行。

—填充应在被集成的每个数据对象(既可后随不被集成的数据对象，也可不后随进一步的数据对象)的结束处使用。

填充由一个值为'80'的必备字节组成，如果需要可后随置为'00'的0至(K-1)个字节，直到相应的数据块被填充直至K个字节为止。当填充字节不被发送时，鉴别的填充不影响传输。

操作方式为“密码块链接”(见ISO10116)。第1个输入是初始校验块与第1个数据块的异或运算结果。第1个输出由第1个输入产生。当前输入是先前输出与当前数据块的异运算结果。最终的校验块就是最后的输出。

—最终步骤—最终步骤从最终校验块中抽取出密码“校验和”(开始的m个字节，至少4个)。

表20示出了密码“校验和”数据对象。

表20 密码“校验和”数据对象

标记	值
'8E'	密码校验和(至少4个字节)

### 5.6.3.2 数字签名数据对象

数字签名计算典型地基于非对称密码技术，数字签名有两种类型：

—带有附件的数字签名；

—给出报文恢复的数字签名。

带有附件数字签名的计算隐含着使用散列函数(见ISO10118)。数据输入或者由数字签名输入数据对象的值(见表21)组成，或者通过本部分规范5.6.3.1定义的机制来确定。

给出报文恢复的数字签名的计算(见ISO9796)不隐含使用散列函数。然后，根据应用的需要，散列代码可以呈现作为恢复报文的一部分，而该报文本身可以是BER-TLV编码的。表21示出了数字签名相关的数据对象。

表21 数字签字有关的数据对象

标记	值
'9A'，'BA'	数字签名输入数据
'9E'	数字签名

### 5.6.4 保密性的数据对象

保密性数据对象预期用于运载密码，其普通值由下列三种情况之一组成：



- BER-TLV，包含SM相关的数据对象；
- BER-TLV，不包含SM相关的数据对象；
- 不是BER-TLV编码的数据。

当普通值不是由BER-TLV编码数据组成时，则必须指示填充。当填充被应用但未被指示时，则本部分规范5.6.3.1定义的规则可应用。

表22示出了保密性的数据对象。

表22 保密性的数据对象

标 记	值
'82'，'83'	密码，简明值由下列内容组成： —BER-TLV，包含SM相关数据对象
'84'，'85'	—BER-TLV，但不是SM相关的数据对象
'86'，'87'	填充指示符字节(见表23)后随密码(普通值不在BER-TLV中编码)

保密性的每一个数据对象可以使用任何密码算法以及任何操作方式，归用于适合的算法引用(见本部分规范5.6.5.1)。在不存在算法引用的情况下以及当没有机制可隐式地被选择用于保密性时，一种默认机制可应用。

对于密码的计算，该密码之前是填充指示符时，默认机制就是使用“电子密码本”的块密码法(见ISO10116)。使用块密码可以包含填充。保密性的填充对传输有影响，因为密码(一个或多个块)大于明文。

表23示出了填充指示符字节

表23 填充指示符字节

值	含 义
'00'	—没有进一步的指示
'01'	—按本部分规范5.6.3.1定义进行填充
'02'	—没有填充
'80' - '8E'	—专有的其他值为RFU

对于密码的计算，该密码之前不是填充指示符字节时，默认机制就是使用“异或”运算的流密码。在这种情况下，密码是被伪装的数据字节串与伪装的相同长度串进行“异或”运算的结果。因此，伪装要求在由相同操作所恢复的值字段中没有填充及被伪装的数据对象。

#### 5.6.5 辅助安全数据对象

算法、密钥和可能的初始数据可以被选择用于每种安全机制。

- 显示地，即，发布命令之前已知；
- 显示地，通过控制引用嵌套在控制引用样板中。

每个命令报文可以运载响应描述符样板，该样板安排了在响应中所要求的数据对象。在响应描述符内，安全机制仍不被应用；接收实体应将该机制应用到构造响应中。

##### 5.6.5.1 控制引用

表24示出了控制引用样板。

表24 控制引用样板

标 记	含 义
'B4'，'B5'	—对密码“校验和”有效的样板

<u>‘B6’, ‘B7’</u>	<u>—对数字签名有效的样板</u>
<u>‘B8’, ‘B9’</u>	<u>—对保密性有效的样板</u>

控制引用样板的最后可能位置正好在被引用机制使用的第1个数据对象之前。例如，密码“校验和”用的样板的最后可能位置正好在被集成到计算中的第1个数据对象之前。每个控制引用保持有效，直到新的控制被提供用于相同机制。例如，一个命令可以为下一个命令安排控制引用。

每个控制引用样板预期用于运载控制引用数据对象(见表25): 算法引用，文件引用，密钥引用，初始数据引用，并且仅在保密性的控制引用样板中，密码内容引用。

算法引用安排了算法及其操作方式(见ISO9979和10116)。算法引用的结构和编码不在本规范本部分中定义。

文件引用表示了密钥引用有效文件，如果没有文件引用呈现，则密钥引用在当前DF中有效。

密钥引用标识了被使用的密钥。

当初始数据引用被应用到密码“校验和”时，该初始数据引用安排了初始校验块，如果没有初始数据引用呈现，并且没有初始校验块显式地被选择，则空块应被使用。此外，在发送保密性的第1个数据对象之前，使用一种流密码时，保密性用的样板应为伪装字节串的计算提供辅助数据。

密码内部引用规范了密码的内容(例如，秘密密钥、初始口令、控制字)。值字段的第1个字节是已命名的密码描述符字节，并且是强制性的。范围‘00’至‘7F’为RFU。范围‘80’至‘FF’为专有的。

表25 控制引用数据对象

标 记	值
<u>‘80’</u>	<u>算法引用</u>
	<u>文件引用</u>
<u>‘81’</u>	<u>—文件标识符或路径</u>
<u>‘82’</u>	<u>—DF名称</u>
	<u>密钥引用</u>
<u>‘83’</u>	<u>—对于直接使用</u>
<u>‘84’</u>	<u>—对于计算会话密钥</u>
	<u>初始数据引用</u>
	<u>*初始校验块</u>
<u>‘85’</u>	<u>—L=0, 空块</u>
<u>‘86’</u>	<u>—L=0, 链接块</u>
<u>‘87’</u>	<u>—L=0, 先前的初始值块加1</u>
	<u>L=k, 初始值块</u>
	<u>*辅助数据</u>
<u>‘88’</u>	<u>—L=0, 先前交换的询问加1</u>
	<u>L=0, 没有进一步的指示</u>
<u>‘89’ 到</u>	<u>—L=0, 专用数据元的索引</u>
<u>‘8D’</u>	<u>—L=0, 专用数据元的值</u>
<u>‘8E’</u>	<u>密码内部引用</u>

#### 5. 6. 5. 2 响应描述符

如果在命令APDU的数据字段中呈现响应描述符样板，则它应安排相应响应的

结构。空数据对象应列出产生响应所需要的全部数据。

处理命令报文的数据字段所使用的安全项目(算法、密钥及初始数据)可以不同于产生后续响应报文的数据字段所使用的那些安全项目。

下列规则应该使用。

—卡应填充每个空的原始数据对象。

—呈现在响应描述符中的每个控制引用样板对于算法、文件和密钥而言应该在带有相同控制引用的相同位置上呈现在响应中。如果响应描述符提供了辅助数据，则在响应中数据对象应为空。如果辅助数据的空引用数据对象呈现在响应描述符中，则在响应中它应为空。

—通过相关的安全机制，使用选择的安全项目时，卡应产生所有请求的安全机制数据对象。表26示出了响应描述符样板。

表26 响应描述符样板

标记	值
'BA' 'BB'	响应描述符

### 5. 6. 6 SM状态条件

在使用安全报文交换的任何命令中，下列特定差错条件可能发生。

—SW1='69'，同时SW2=

. '87':期望的SM数据对象失踪。

. '88':SM数据对象不正确。

## 6 基本的行业间命令

对于遵循本规范本部分的所有卡而言，应该不强制要求支持本部分描述的所有命令或支持命令的所有选项。

当进行国际交换时，卡系统服务及相关命令和选项的集合应按照第9章的定义加以使用。表11提供了本规范本部分定义的命令概要。

安全报文交换(见本部分规范5.6)对报文结构的影响不在本章中描述。

在6.X.5的每一条中所给出的差错和报警条件的列表不是穷举的(见本规范本部分5.4.5)。

### 6. 1 READ BINARY命令

#### 6.1.1 定义和范围

READ BINARY响应报文给出了带有透明结构的EF内容的一部分。

#### 6.1.2 使用与安全的条件

当命令包含了有效的短EF标识符时，它将文件置位为当前EF。

根据当前选择的EF来处理该命令。仅当安全状态满足了用于该功能的为该EF而定义的安全属性时，才能执行该命令。

如果命令被应用到不带有透明结构的EF，则应放弃该命令。

#### 6.1.3 命令报文

表27 READ BINARY命令APDU

<u>CLA</u>	<u>按5.4.1定义的</u>
<u>INS</u>	<u>‘B0’</u>
<u>P1-P2</u>	<u>见以下文本</u>
<u>L<sub>c</sub>字段</u>	<u>空</u>
<u>数据字段</u>	<u>空</u>
<u>L<sub>c</sub>字段</u>	<u>待读的字节数</u>

如果在P1中b8=1，则P1的b7和b6置为0(RFU若干位)，P1的b5至b1是短EF标识符，并且P2是在从文件开始的数据单元中被读的第1个字节的偏移。

如果在P1中b8=0，则P1 P2是在从文件开始的数据单元中被读的第1个字节的偏移。

#### 6.1.4 响应报文(标称情况)

如L<sub>c</sub>字段仅包含若干“0”，则对于短的长度在不超过256的范围内或者对扩充长度在不超过65536的范围内，所有字节(直到文件结束为止)应被读出。

表28 READ BINARY响应APDU

<u>数据字段</u>	<u>读的数据(L<sub>c</sub>字节)</u>
<u>SW1-SW2</u>	<u>状态字节</u>

#### 6.1.5 状态条件

下列特定报警条件可能发生。

—SW1=‘62’，同时SW2=

· ‘81’：被返回数据的一部分可以被损坏。

· ‘82’：读L<sub>c</sub>字节之前达到的文件结束。

下列特定差错条件可能发生。

—SW1=‘67’，同时SW2=

· ‘00’：错误的长度(错误的L<sub>c</sub>字段)。

—SW1=‘69’，同时SW2=

· ‘81’：命令与文件结构不兼容。

· ‘82’：安全状态不被满足。

· ‘86’：命令不被允许(没有当前EF)。

—SW1=‘64’，同时SW2=

· ‘81’：功能不被支持。

· ‘82’：文件未被找到。

—SW1=‘6B’，同时SW2=

· ‘00’：错误的参数(偏移超出EF)。

—SW1=‘6C’，同时SW2=

‘XX’：错误的长度(错误的L<sub>c</sub>字段；‘XX’表示正确长度)。

## 6.2 WRITE BINARY命令

### 6.2.1 定义和范围

WRITE BINARY命令报文启动将二进制值写入EF。

根据文件属性，命令应执行下列操作之一：

—早已存在卡内的位与在命令APDU中给出的位进行逻辑“或”运算(该文件位的逻辑擦除状态为“0”)。

—对早已存在卡内的位与在命令APDU中给出的位进行逻辑“与”运算(该文件位的逻辑擦除状态为“1”)。

—将命令APDU中给出的位一次写入卡的操作。

当在数据编码字节中未给出指示(见表86)时，则逻辑“或”行为应该适用。

### 6.2.2 使用与安全的条件

当命令包含了有效的短EF标识符时，它将文件置位为当前EF。

根据当前选择的EF来处理该命令。仅当安全状态满足了用于写功能的安全属性时，才能执行该命令。

一旦WRITE BINARY已经被应用到一次写EF的数据单元，如果数据单元的内容或被连接到该数据单元的逻辑擦除状态指示符(如果有)不同于逻辑擦除状态，则涉及该数据单元的任何进一步的写操作将被放弃。

如果命令被施加到不带有透明结构的EF，则应放弃该命令。

### 6.2.3 命令报文

表29 WRITE BINARY命令APDU

CLA	按5.4.1定义的
INS	‘D0’
P1-P2	见以下文本
L <sub>c</sub> 字段	后续数据字段的长度
数据字段	待写的数据单元串
L <sub>c</sub> 字段	空

如果在P1中b8=1，则P1的b7和b6显域 0(RFU若干位)，P1的b5至b1是短EF标识符，并且P2是在从文件开始的数据单元中被写的第1个字节的偏移。

如果在P1中b8=0，则P1 P2是在从文件开始的数据单元中被写的第1个字节的偏移。

### 6.2.4 响应报文(标称情况)

表30 WRITE BINARY响应APDU

数据字段	空
SW1-SW2	状态字节

### 6.2.5 状态条件

下列特定报警条件可能发生。

—SW1=‘63’，同时SW2=

· ‘CX’:计数器(成功的写，但是在使用内部重试例行程序之后，‘X’=0表示重试数；‘X’=0意味着没有计数器被提供)。

下列特定差错条件可能发生。

—SW1=‘65’，同时SW2=

· ‘81’:存储器故障(不成功的写)

—SW1=‘67’，同时SW2=

· ‘00’:错误的长度(错误的L<sub>c</sub>字段)

—SW1=‘69’，同时SW2=

· ‘81’:命令与文件结构不兼容

· ‘82’:安全状态不被满足

· ‘86’:命令不被允许(没有当前EF)

—SW1=‘64’，同时SW2=

- ‘81’ :功能不被支持
- ‘82’ :文件未被找到
- SW1=‘6B’，同时SW2=
- ‘00’ :错误的参数(偏移超出EF)。

### 6.3 UPDATE BINARY命令

#### 6.3.1 定义和范围

UPDATE BINARY命令报文启动使用在命令APDU中给出的位来更新早已呈现在EF中的位。

#### 6.3.2 使用与安全的条件

当命令包含了有效的短EF标识符时，它将文件置位为当前EF。

根据当前选择的EF来处理该命令。仅当安全状态满足了用于更新功能的安全属性时，才能执行该命令。

如果命令被施加到不带有透时结构的EF，则应放弃该命令。

#### 6.3.3 命令报文

表31 UPDATE BINARY命令APDU

CLA按	按5.4.1定义的
INS	见以下文本
P1-P2	后续数据字段的长度
数据字段	待更新的数据单元串
L <sub>c</sub> 字段	空

如果在P1中b8=1，则P1的b7和b6置为0(RFU若干位)，P1的b5至b1是短EF标识符，并且P2是在从文件开始的数据单元中被更新的第1个字节的偏移。

如果在P1中b8=0，则P1 P2是在从文件开始的数据单元中被更新的第1个字节的偏移。

#### 6.3.4 响应报文(标称情况)

表32 UPDATE BINARY响应APDU

数据字段	空
SW1-SW2	状态字节

#### 6.3.5 状态条件

下列特定报警条件可能发生。

—SW1=‘63’，同时SW2=

· ‘CX’ :计数器(成功的更新，但是在使用内部重试例行程序之后，‘X’=0表示重试数‘X’=0意味着没有计数器被提供)。

下列特定差错条件可能发生。

—SW1=‘65’，同时SW2=

· ‘81’ :存储器故障(不成功的更新)。

—SW1=‘67’，同时SW2=

· ‘00’ :错误的长度(错误的L<sub>c</sub>字段)。

—SW1=‘69’，同时SW2=

· ‘81’ :命令与文件结构不兼容。

- ‘82’ :安全状态不被满足。
- ‘86’ :命令不被允许(没有当前EF)。
- SW1=‘6A’，同时SW2=
- ‘81’ :功能不被支持。
- ‘82’ :文件未被找到。
- SW1=‘69’，同时SW2=
- ‘00’ :错误的参数(偏移超出EF)。

## 6.4 ERASE BINARY命令

### 6.4.1 定义和范围

ERASE BINARY命令报文顺序地从给出的偏移开始将EF的内容的一部分置为其逻辑擦除的状态。

### 6.4.2 使用与安全的条件

当命令包含了有效的短EF标识符时，它将文件置为当前EF。

根据当前选择的EF来处理该命令。仅当安全状态满足了用于擦除功能的安全属性时，才能执行该命令。

如果命令被施加到不带有透明结构的EF，则应放弃该命令。

### 6.4.3 命令报文

表33 ERASE BINARY命令APDU

CLA	按5.4.1定义的
INS	‘0E’
P1-P2	见以下文本
L <sub>c</sub> 字段	空或‘02’
数据字段	见以下文本
L <sub>c</sub> 字段	空

如果在P1中b8=1，则P1的b7和b6置为‘0’(RFU若干位)，P1的b5至b1是短EF标识符，P1是在从文件开始的数据单元中被擦除的第1个字节的偏移。

如果在P1中b8=0，则P1 P2是在从文件开始的数据单元中被擦除的第1个字节的偏移。

如果数据字段呈现，它编码不被擦除的第1个数据单元的偏移。该偏移应大于在P1-P2中编码的一个偏移。当数据字段为空时，该命令擦除到该文件的结束端。

### 6.4.4 响应报文(标称情况)

表34 ERASE BINARY响应APDU

数据字段	空
SW1-SW2	状态字节

### 6.4.5 状态条件

下列特定报警条件可能发生。

—SW1=‘63’，同时SW2=

· ‘CX’ :计数器(成功的擦除，但是在使用内部重试例行程序之后，‘X’ 0表示重试数，‘X’ =0意味着没有计数器被提供)。

下列特定差错条件可能发生：

- SW1='65'，同时SW2=
- '81'：存储器故障（不成功的擦除）。
- SW1='67'：同时SW2=
- '00'：错误的长度（错误的L<sub>e</sub>字段）。
- SW1='69'，同时SW2=
- '81'：命令与文件结构不兼容。
- '82'：安全状态不被满足。
- '86'：命令不被允许（没有当前EF）。
- SW1='6A'，同时SW2=
- '81'：功能不被支持
- '82'：文件未找到
- SW1='6B'，同时SW2=
- '00'：错误的参数（偏移超出EF）。

## 6.5 READ RECORD命令

### 6.5.1 定义和范围

READ RECODE响应报文给出了EF的规定记录的内容或EF的一个记录开始部分的内容。

### 6.5.2 使用与安全的条件

仅当安全状态满足了用于读功能的该EF的安全属性时，才能执行该命令。

如果在发出命令的时刻，当前选择了EF，则该命令可以被处理，而无需该文件的标识。

当命令包含了有效的短EF标识符时，它将文件置为当前EF，并且复位当前记录指针。

如果命令被施加到不带有记录结构的EF，则应放弃该命令。

### 6.5.3 命令报文

表35 READ RECORD [S] 命令APDU

CAL	按5.4.1定义的
INS	'B2'
P1	记录号或被读的第1个记录的标识符 ( '00' 表示当前记录)
P2	引用控制 按照表36
L <sub>e</sub> 字段	空
数据字段	空
L <sub>e</sub> 字段	被读字节数

表36 引用控制P2的编码

b8 b7 b6 b5 b4 b3 b2 b1	含 义
0 0 0 0 0 - -	—当前选择的EF
-	—短EF标识符
x x x x x - - -	RFU
(不全相等)	
1 1 1 1 1 - -	



二	
<u>- - - - - 1 × ×</u> <u>- - - - - 1 0</u> 0 <u>- - - - - 1 0</u> 1 <u>- - - - - 1 1</u> 0 <u>- - - - - 1 1</u> 1 <u>- - - - - 0 × ×</u> <u>- - - - - 0 0</u> 0 <u>- - - - - 0 0</u> 1 <u>- - - - - 0 1</u> 0 <u>- - - - - 0 1</u> 1	<u>利用P1中的记录号</u> <u>—READ RECODE#P1</u> <u>—读从P1到最后的所有记录</u> <u>—读从最后到P1的所有记录</u> RFU <u>利用P1中的记录标识符</u> <u>—读第1个出现(标识符)</u> <u>—读最后一个出现(标识符)</u> <u>—读下一个出现(标识符)</u> <u>—读先前一个出现(标识符)</u>

6.5.4 响应报文(标称情况)

如果L<sub>e</sub>字段仅包含了若干‘0’，则根据P2的b3 b2 b1，并且对于短的长度在不超过256的范围内，或者对扩充的长度在不超过65536的范围内，该命令应完整地读出：

- 单个请求的记录；
- 或请求的记录序列

表37 READ RECODE [S] 响应APDU

数据字段 SW1-SW2	L <sub>r</sub> (可以等于L <sub>e</sub> )字节 见表38 状态字节
-----------------	---

当记录是简单TLV数据对象(见5.4.5)时，表38示出了响应报文数据字段的格式。

表38-4 当读一个记录时，响应的数据字段

情况a—部分读的一个记录

I <sub>n</sub> 1个字节	L <sub>n</sub> 1个或3个字节	记录中的前面若干数据字节
------------------------	---------------------------	--------------

-----L<sub>e</sub>字节

----- 当L<sub>e</sub>字段不仅包含了‘0’时，该情况适用。

情况b—完整读的一个记录

I <sub>n</sub> 1个字节	L <sub>n</sub> 1个或3个字节	记录的整个数据字节 L <sub>e</sub> 字节
------------------------	---------------------------	--------------------------------

当L<sub>e</sub>字段仅包含若干‘0’时，该情况适用。

表38-2 当读几个记录时，响应的数据字段

情况c—部分读的记录序列

记录#n	.....	记录#n + m中的前面若干字节
$T_n$ $L_n$ $V_n$	.....	$T_{n+m}$ $L_{n+m}$ $V_{n+m}$

----- $L_e$ 字节

当 $L_e$ 字段不仅包含了若干‘0’时，该情况适用。

情况d—读多个记录直到文件结束

记录#n	.....	记录#n + m
$T_n$ $L_n$ $V_n$	.....	$T_{n+m}$ $L_{n+m}$ $V_{n+m}$

当 $L_e$ 字段仅包含了若干‘0’时，该情况适用。

数据字段的长度与其TLV结构相比较给出了数据的性质：唯一记录(读一个记录)或最后一个记录(读所有记录)是不完整的，完整的或添加的。

注：如TLV编码不被使用，则读所有记录的功能导致接收的几个记录没有标准的记录定界。

#### 6.5.5 状态条件

下列特定报警条件可能发生。

—SW1=‘62’，同时SW2=

· ‘81’：被返回数据的一部分可以被损坏。

· ‘82’：在 $L_e$ 字节之前已到达记录结束端。

下列特定差错条件可能发生。

—SW1=‘67’，同时SW2=

· ‘00’：错误的长度(空的 $L_e$ 字段)

—SW1=‘69’，同时SW2=

· ‘81’：命令与文件结构不兼容。

· ‘82’：安全状态不被满足。

—SW1=‘6A’：同时SW2=

· ‘81’：功能不被支持。

· ‘82’：文件未被找到。

· ‘83’：记录未被找到。

—SW1=‘6C’，同时SW2=

· ‘XX’：错误的长度(错误的 $L_e$ 字段；‘XX’表示正确的长度)。

## 6.6 WRITE RECORD命令

### 6.6.1 定义和范围

WRITE RECORD命令报文启动下列操作之一：

—写一次记录；

—对早已呈现在卡内的记录数据字节与在命令APDU中给出的记录数据字节进行逻辑“或”运算；

—对早已呈现在卡内的记录数据字节与在命令APDU中给出的记录数据字节

进行逻辑“和”运算。

当在数据编码字节中未给出指示(见表86)时，逻辑“或”运算应该适用。

当使用当前记录寻址时，该命令应将记录指针设置在成功的WRITE RECORD上。

### 6.6.2 使用与安全的条件

仅当安全状态满足了用于写功能的该EF的安全属性时，才能执行该命令。

如果在发出命令的时刻，当前选择了EF，则该命令可以被处理，而无需该文件的标识。

当命令包含了有效的短EF标识符时，它将文件置为当前EF，并且复位当前记录指针。

如果命令被施加到不带有记录结构的EF，则应放弃该命令。

被施加到循环文件的“先前”的命令选项(P2 = xxxxx011)具有和APPEND RECORD相同的行为。

### 6.6.3 命令报文

表39 WRITE RECORD命令APDU

CLA	按5.4.1定义的
INS	‘D2’
P1	P1 = ‘00’ 指明当前记录 P1 ‘00’ 是所规定记录的号
P2	按照表40
L <sub>c</sub> 字段	后续数据字段的长度
数据字段	待写的记录
L <sub>e</sub> 字段	空

表40 引用控制P2的编码

b8 b7 b6 b5 b4 b3 b2 b1	含义
0 0 0 0 0 - - - x x x x x - - - (不全相等)	—当前选择的EF —短EF标识符
- - - - - 0 0 0	—第1个记录 —最后一个记录
- - - - - 0 0 1	—下一个记录 —先前一个记录
- - - - - 0 1 0	—在P1中给出的记录号
- - - - - 0 1 1	
- - - - 1 0 0	
任何其他值	RFU

当记录为简单TLV数据对象(见本部分规范5.4.4)时，表41示出了命令报文数据字段的格式。

表41 命令的数据字段  
完整写的一个记录

<u>T<sub>0</sub></u> <u>1个字节</u>	<u>L<sub>0</sub></u> <u>1个或3个字节</u>	<u>记录的整个数据字</u> <u>节</u> <u>L<sub>0</sub>字节</u>
-------------------------------------	--	---

#### 6.6.4 响应报文(标称情况)

表42 WRITE RECORD响应APDU

<u>数据字段</u> <u>SWL-SW2</u>	<u>空</u> <u>状态字节</u>
-------------------------------	-------------------------

#### 6.6.5 状态条件

下列特定报警条件可能发生。

—SW1 = ‘63’， 同时SW2 =

· ‘CX’ :计数器(成功的写，但是使用内部重试例行程序之后，‘X’ 0’表示重试数；‘X’ = ‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

—SW1 = ‘65’， 同时SW2 =

· ‘81’ :存储器故障(不成功的写)。

—SW1 = ‘67’， 同时SW2 =

· ‘00’ :错误的长度(空的L<sub>0</sub>字段)。

—SW1 = ‘69’， 同时SW2 =

· ‘81’ :命令与文件结构不兼容。

· ‘82’ :安全状态不被满足。

· ‘86’ :命令不被允许(没有当前EF)。

—SW1 = ‘6A’， 同时SW2 =

· ‘81’ :功能不被支持。

· ‘82’ :文件未被找到。

· ‘83’ :记录未被找到。

· ‘85’ :L<sub>0</sub>与TLV结构不一致。

## 6.7 APPEND RECORD命令

### 6.7.1 定义和范围

APPEND RECORD命令报文启动在线性结构EF的结束端添加记录，或者在循环结构(见本部分规范5.1.4)的EF内写记录号1。

命令应将记录指针设置在成功添加的记录上。

### 6.7.2 使用与安全的条件

仅当安全状态满足了用于添加功能的该EF的安全属性时，才可执行该命令。

如果在发布命令的时刻，当前选择了EF，则该命令可以被处理，而无需该文件的标识。当命令包含了有效的短EF标识符时，它将文件置位为当前EF，并且复位当前记录指针。

如果命令被应用到不带有记录结构的EF，则应放弃该命令。

注:如果该命令被应用到有很多记录的循环结构EF，则带有最高记录号的记录可被代替。该记录变成为记录号1。

### 6.7.3 命令报文

表43 APPEND RECORD命令APDU

CLA	按RPV 5.4.1定义
INS	'E2'
P1	只有P1 = '00' 是有效的
P2	按照表44
L <sub>c</sub> 字段	后续数据字段的长度
数据字段	待添加的记录
L <sub>e</sub> 字段	空

表44 引用控制P2的编码

b8 b7 b6 b5 b4 b3 b2	含义
b1	
0 0 0 0 0 - - -	—当前选择的EF
x x x x x - - -	—短EF标识符
(不全相等)	
任何其他值	RFU

当记录是简单TLV数据对象(见5.4.4)时, 表45示出了命令报文数据字段的格式。

表45 命令的数据字段  
完整添加的一个记录

I <sub>0</sub> 1个字节	L <sub>0</sub> 1个或3个字节	记录的完整数据字节 L <sub>0</sub> 字节
------------------------	---------------------------	--------------------------------

#### 6.7.4 状态条件

下列特定报警条件可能发生。

—SW1 = '63', 同时SW2 =

· 'CX': 计数器(成功的添加, 但是在使用内部重试例行程序之后, 'X' 0 表示重试数; 'X' = '0' 意味着没有计数器被提供)。

下列特定差错条件可能发生。

—SW1 = '65', 同时SW2 =

· '81': 存储器故障(不成功的添加)。

—SW1 = '67', 同时SW2 =

· '00': 错误的长度(空的L<sub>c</sub>字段)。

—SW1 = '69', 同时SW2 =

· '81': 命令与文件结构不兼容。

· '82': 安全状态不被满足。

· '86': 命令不被允许(没有当前EF)。

—SW1 = '6A', 同时SW2 =

· '81': 功能不被支持。

· '82': 文件未被找到。

· '84': 无足够的文件存储空间。

· '85': L<sub>c</sub>与TLV结构不一致。

## 6.8 UPDATE RECORD命令

### 6.8.1 定义和范围

UPDATE RECORD命令报文启动使用命令APDU给出的位来更新特定记录。  
当使用当前记录寻址时，该命令应将记录指针设置在成功的更新记录上。

### 6.8.2 使用与安全的条件

仅当安全状态满足了用于更新功能的该EF的安全属性时，才能执行该命令。  
如果在发布命令的时刻，当前选择了EF，则该命令可以被处理，而无需该文件的标识。

当命令包含了有效的短EF标识符时，它将文件置位为当前EF，并且复位当前记录指针。

如果命令被施加到不带有记录结构的EF，则应放弃该命令。

当命令适用于带有线性固定结构或循环结构的EF时，如果该记录长度不同于现有记录的长度，则应放弃该命令。

当命令适用于带有线性可变结构的EF时，并且当该记录长度不同于现有记录的长度时，则可以完成该命令。

被施加到循环文件的“先前”的命令选项(P2 = XXXXX011)具有和APPEND RECORD相同的行为。

### 6.8.3 命令报文

表47 UPDATE RECORD命令APDU

CLA	按5.4.1定义的
INS	‘DC’
P1	P1 = ‘00’ 指明当前记录 P = ‘00’ 是所规定记录的号
P2	按照表48
L <sub>c</sub> 字段	后续数据字段的长度
数据字段	待更新的记录
L <sub>e</sub> 字段	空

表48 引用控制P2的编码

b8 b7 b6 b5 b4 b3 b2	含义																																																								
<table border="1"> <tr><td colspan="7">b1</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-</td><td>-</td></tr> <tr><td colspan="7">=</td></tr> <tr><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>-</td><td>-</td></tr> <tr><td colspan="7">[不全相等]</td></tr> </table>	b1							0	0	0	0	0	-	-	=							x	x	x	x	x	-	-	[不全相等]							<u>—当前选择的EF</u> <u>—短EF标识符</u>																					
b1																																																									
0	0	0	0	0	-	-																																																			
=																																																									
x	x	x	x	x	-	-																																																			
[不全相等]																																																									
<table border="1"> <tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td></tr> <tr><td colspan="7">0</td></tr> <tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>0</td></tr> <tr><td colspan="7">1</td></tr> <tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>0</td><td>1</td></tr> <tr><td colspan="7">1</td></tr> <tr><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>1</td><td>0</td></tr> <tr><td colspan="7">0</td></tr> </table>	-	-	-	-	-	0	0	0							-	-	-	-	-	0	0	1							-	-	-	-	-	0	1	1							-	-	-	-	-	1	0	0							<u>—第1个记录</u> <u>—最后一个记录</u> <u>—先前一个记录</u> <u>—在P1中给出的记录号</u>
-	-	-	-	-	0	0																																																			
0																																																									
-	-	-	-	-	0	0																																																			
1																																																									
-	-	-	-	-	0	1																																																			
1																																																									
-	-	-	-	-	1	0																																																			
0																																																									
任何其他值	RFU																																																								

当记录是简单TLV数据对象(见5.4.4)时，表49示出了命令报文数据字段的格式。

表49 命令的数据字段

### 完整更新的一个记录

$T_n$ 1个字节	$L_n$ 1个或3个字节	记录的完整数据字节 $L_n$ 字节
---------------	------------------	-----------------------

#### 6.8.4 响应报文(标称情况)

表50 UPDATE RECORD响应APDU

数据字段 SWL-SW2	空 状态字节
-----------------	-----------

#### 6.8.5 状态条件

下列特定报警条件可能发生。

—SW1 = ‘63’，同时SW2 =

· ‘CX’:计数器(成功的更新,但是在使用内部重试例行程序之后,‘X’=0表示重试数;‘X’=‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

—SW1 = ‘65’，同时SW2 =

· ‘81’:存储器故障(不成功的更新)。

—SW1 = ‘67’，同时SW2 =

· ‘00’:错误的长度(空的 $L_c$ 字段)。

—SW1 = ‘69’，同时SW2 =

· ‘81’:命令与文件结构不兼容。

· ‘82’:安全状态不被满足。

· ‘86’:命令不被允许(没有当前EF)。

—SW1 = ‘6A’，同时SW2 =

· ‘81’:功能不被支持。

· ‘82’:文件未被找到。

· ‘83’:记录未被找到。

· ‘84’:无足够的文件存储空间。

· ‘85’: $L_c$ 与TLV结构不一致。

## 6.9 GET DATA 命令

### 6.9.1 定义和范围

GET DATA 命令可在当前上下文(例如,应用特定环境或当前DF)范围内用于检索一个原始数据对象或者包含在结构化数据对象中所包含的一个或多个数据对象。

### 6.9.2 使用与安全的条件

仅当安全状态满足了通过功能用的上下文范围内的应用所定义的安全条件时,才能执行该命令。

### 6.9.3 命令报文

表51 GET DATA 命令APDU

<u>CLA</u>	<u>按 5.4.1定义的</u>
<u>INS</u>	<u>‘CA’</u>
<u>P1-P2</u>	<u>见表52</u>
<u>L<sub>c</sub>字段</u>	<u>空</u>
<u>数据字段</u>	<u>空</u>
<u>L<sub>e</sub>字段</u>	<u>在响应时期望的字节数</u>

表52 参数P1-P2的编码

<u>值</u>	<u>含 义</u>
<u>‘0000’至‘003F’</u>	<u>RFU</u>
<u>‘0040’至‘00FF’</u>	<u>P2中的BER-TLV标签(1个字节)</u>
<u>‘0100’至‘01FF’</u>	<u>应用数据(专有编码)</u>
<u>‘0200’至‘02FF’</u>	<u>P2中的简单TLV标签</u>
<u>‘0300’至‘3FFF’</u>	<u>RFU</u>
<u>‘0400’至‘FFFF’</u>	<u>P1-P2中的BER-TLV标签(2个字节)</u>

得到应用数据

· 当P1-P2的值位于从‘0100’至‘01FF’的范围时，P1-P2的值应是被保留的一个标识符，它可在给定的应用上下文范围内用于卡内部测试和用于有意义的专有服务。

GET DATA对象

· 当P1-P2的值位于从‘0040’至‘00FF’的范围时，P2的值应是单个字节的BER-TLV

标签。值‘00FF’被保留，为了获得上下文内可读的所有公共的BER-TLV数据对象。

· 当P1-P2的值位于从‘0200’至‘02FF’的范围时，P2的值应是简单TLV标签。值‘0200’是RFU。值‘02FF’被保留，为了获得在上下文内可读的所有公共的简单TLV数据对象。

· 当P1-P2的值位于从‘0400’至‘FFFF’的范围时，P1-P2的值应是2个字节的BER-TLV标记。值‘4000’和‘FFFF’是RFU。

当请求原始数据对象时，响应报文的数据字段应包含结构化数据对象的值。

当请求原始数据对象时，响应报文的数据字段应包含结构化数据对象的值，即包含其标签，长度和值的数据对象。

6.9.4 响应报文(标称情况)

如果L<sub>c</sub>字段仅包含若干“0”，则对于短的长度在不超过256的范围内或者对于扩充的长度在不超过65536的范围内，所有要求的信息应被返回。

表53 GET DATA响应APDU

<u>数据字段</u>	<u>L<sub>r</sub>(可以等于L<sub>e</sub>)字节</u>
<u>SWL-SW2</u>	<u>状态字节</u>

6.9.5 状态条件

下列特定报警条件可能发生。

—SW1 = ‘62’，同时SW2 =

· ‘81’：被返回数据的一部分可以被损坏。



下列特定差错条件可能发生。

—SW1 = ‘67’， 同时SW2 =

· ‘00’ :错误的长度(空的L<sub>e</sub>字段)

—SW1 = ‘69’， 同时SW2 =

· ‘82’ :安全状态不被满足。

· ‘85’ :使用的条件不被满足。

—SW1 = ‘6A’， 同时SW2 =

· ‘81’ :功能不被支持。

· ‘82’ :文件未被找到。

· ‘88’ :引用的数据(数据对象)未被找到。

—SW1 = ‘6C’， 同时SW2 =

· ‘XX’ :错误的长度(错误的L<sub>e</sub>字段;‘XX’表示正确的长度)。

## 6.10 PUT DATA 命令

### 6.10.1 定义和范围

PUT DATA 命令可在当前上下文(例如, 应用特定环境或当前DF)范围内用于存储一个原始数据对象或者包含在结构化数据对象中的一个或多个数据对象。正确的存储功能(写一次和/或更新和/或添加)通过数据对象的定义和性质来引出。

注:例如, 该命令可用来更新数据对象。

### 6.10.2 使用与安全的条件

仅当安全状态满足了通过功能用的上下文内的应用所定义的安全条件时, 才能执行该命令。

### 6.10.3 命令报文

表54 PUT DATA 命令APDU

<u>CLA</u>	<u>按 5.4.1定义的</u>
<u>INS</u>	<u>‘DA’</u>
<u>P1-P2</u>	<u>见表55</u>
<u>L<sub>e</sub>字段</u>	<u>后续数据字段的长度</u>
<u>数据字段</u>	<u>待写的参数和数据</u>
<u>L<sub>e</sub>字段</u>	<u>空</u>

表55 参数P1-P2的编码

<u>值</u>	<u>含 义</u>
<u>‘0000’ 至‘003F’</u>	<u>RFU</u>
<u>‘0040’ 至‘00FF’</u>	<u>P2中的BER-TLV标记(1个字节)</u>
<u>‘0100’ 至‘01FF’</u>	<u>应用数据(专有编码)</u>
<u>‘0200’ 至‘02FF’</u>	<u>P2中的简单TLV标记</u>
<u>‘0300’ 至‘3FFF’</u>	<u>RFU</u>
<u>‘4000’ 至‘FFFF’</u>	<u>P1-P2中的BER-TLV标记(2个字节)</u>

### 存储应用数据

· 当P1-P2的值位于从‘0100’至‘01FF’的范围内时，P1-P2的值应是被保留的一个标识符，它可在给定的应用上下文范围内用于卡内部测试和用于有意义的专有服务。

### 存储数据对象

· 当P1-P2的值位于从‘0040’至‘00FF’的范围内时，P2的值应是单个字节的BER-TLV标记，值‘00FF’被保留，为了表示数据字段运载了BER-TLV数据对象。

· 当P1-P2的值位于从‘0200’至‘02FF’的范围时，P2的值应是简单TLV标记。

值‘0200’为RFU，值‘02FF’被保留，为了表示数据字段运载了简单TLV数据对象。

· 当P1-P2的值位于从‘4000’至‘FFFF’的范围内时，P1-P2的值应是2个字节的BER-TLV标记，值‘4000’和‘FFFF’为RFU。

当提供了原始数据对象时，命令报文的数据字段应包含对应于原始数据对象的值。

当提供了结构化数据对象时，命令报文的数据字段应包含结构化数据对象的值，即包括其标记、长度和值的数据对象。

## 6.10.4 响应报文(标称情况)

表56 PUT DATA响应APDU

<u>数据字段</u> SW1-SW2	<u>空</u> 状态字节
------------------------	------------------

## 6.10.5 状态条件

下列特定报警条件可能发生。

—SW1 = ‘63’，同时SW2 =

· ‘CX’：计数器(成功的存储，但是在使用内部例行程序之后，‘X’ 0表示重试数；‘X’ = ‘0’意味着没有计数器被提供)。

下列特定差错条件可能发生。

—SW1 = ‘65’，同时SW2 =

· ‘81’：存储器故障(不成功的存储)

—SW1 = ‘67’，同时SW2 =

· ‘00’：错误的长度(错误的L<sub>c</sub>字段)

—SW1 = ‘69’，同时SW2 =

· ‘82’：安全状态不被满足。

· ‘85’：使用的条件不被满足。

—SW1 = ‘6A’，同时SW2 =

· ‘80’：数据字段中的不正确参数。

· ‘81’：功能不被支持。

· ‘84’：无足够的文件存储空间。

· ‘85’：L<sub>c</sub>与TLV结构不一致。

## 6.11 SELECT FILE 命令

### 6.11.1 定义和范围

成功的SELECT FILE在逻辑信道内(见本部分规范5.5)设置当前文件。后续命令可以通过那个逻辑信道隐式地引用该当前文件。

选择DF(它可以是MF)时可将其设置为当前DF。在这种选择之后，隐式当前EF可以通过那个逻辑信道来引用。

选择EF时设置了一对当前文件:EF及其父辈文件。

在应答复位之后，MF可通过基本逻辑信道(见5.5.2)隐式地进行选择，除非在历史字节(见本部分规范第8章)中或在初始数据串(见本部分规范第9章)中有不同的规定。

注:利用DF名称的直接选择可以用来选择按照本规范第5部分所登记的应用。

### 6.11.2 使用与安全的条件

下列条件应该适用于每个开放逻辑信道。

除非另有规定，否则按照下列规则，正确执行命令可修改安全状态(见5.2.1)。

—在当前EF被改变时，或在没有当前EF时，专门针对以前的当前DF的安全状态(如果有)被丢失。

—在当前DF是以前的当前DF的后代，或同代时，专门针对以前的当前EF的安全状态被保持。

—在当前DF既不是以前的当前DF的后代，也不是同代时，专门针对以前的当前EF的安全状态被丢失。先前和新当前DF的所有共同祖先，所共用的安全状态被保持。

### 6.11.3 命令报文

表57 SELECT FILE 命令APDU

CLA	按5.4.1定义的
INS	'A4'
P1	选择控制，见表58
P2	选择选项，见表59
L <sub>c</sub> 字段	空或后续数据字段的长度
数据字段	如果存在下列内容，则按照P <sub>1</sub> -P <sub>2</sub>
	—文件标识符
	—MF的路径
	—当前DF的路径
	—DF名称
L <sub>c</sub> 字段	空或在响应时期望的数据最大长度

表58 选择控制P1的编码

b8	b7	b6	b5	b4	b3	b2	含义
b1							
0	0	0	0	0	0	× ×	通过文件标识符来选择
0	0	0	0	0	0	0	—选择MF、DF或EF (数据字段 = 标识符或空)
0							—选择子女DF (数据字段 = DF标识符)
0	0	0	0	0	0	0 1	—根据当前DF选择EF (数据字段 = EF标识符)
0	0	0	0	0	0	1 0	—选择当前DF的父辈DF (空的数据字段)
0	0	0	0	0	0	1 1	

续表58 选择控制P<sub>1</sub>的编码

b8 b7 b6 b5 b4 b3 b2 b1	含义
0 0 0 0 0 1 × × 0 0 0 0 0 1 0 0	通过DF名称来选择 —通过DF名称直接选择 (数据字段 = DF名称) RFU RFU RFU
0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1	通过路径来选择(见5.1.2) —由MF选择(数据字段 = 路径 而 无需MF的标识符) —由当前DF选择(数据字段 = 路 径 而无需当前DF的标识符) RFU RFU RFU
任何其它值	RFU

— 当P<sub>1</sub> = '00' 时，卡会知道选择的文件是否为MF、DF或EF，是因为有文件标识的特定编码或者因为有命令执行的上下文。

当P<sub>1</sub>-P<sub>2</sub> = '0000' 时，如果文件标识符被提供，则在下列环境下，该文件标识符应是唯一的：

- 当前DF的直接子女，
- 父辈DF，
- 父辈DF的直接子女，

如果P<sub>1</sub>-P<sub>2</sub> = '0000'，如果数据字段为空或等于'3F00'，则选择MF。

当P<sub>1</sub> = '04' 时，数据字段为DF名称，但可能权利被截断。当被支持时，带有相同数据字段的这种连续命令应选择名称与数据字段相匹配的DF，即，以命令数据字段开始。如果卡接受了带有空数据字段的SELECT FILE 命令，则全部DF或DF的子集可以连续地被选择。

注：关于卡所支持的选择方法见本部分规范8.3.6。

表59 选择选项P<sub>2</sub>的编码

b8 b7 b6 b5 b4 b3 b2 b1	含义
0 0 0 0 — — 0 0	—第1个或唯一出现(选项) —最后一个出现(选项)
0 0 0 0 — — 0 1	—下一个出现(选项) —先前一个出现(选项)
0 0 0 0 — — 1 0	
0 0 0 0 — — 1 1	

0 0 0 0 × × —	文件控制信息选项(见5.1.5) —返回FC1, 任选的样板 —返回FCP样板 —返回FMD样板
0 0 0 0 0 0 —	
0 0 0 0 0 1 —	
0 0 0 0 1 0 —	
任何其他值	RFU

#### 6.11.4 响应报文(标称情况)

如果 $L_c$ 字段仅包含“0”，则对于短的长度在不超过256的范围内或对于扩充的长度在不超过65536的范围内，对应于选择选项的全部字节应被返回。

表60 SELECT FILE响应APDU

数据字段 SW1-SW2	信息按照P2(至多 $L_c$ 个字节) 状态字节
-----------------	------------------------------

#### 6.11.5 状态条件

下列特定报警条件可能发生。

- SW1 = ‘62’，同时SW2 =
- ‘83’：选择的文件无效。
- ‘84’：FCI格式化未按照5.1.5。

下列特定差错条件可能发生。

- SW1 = ‘6A’，同时SW2 =
- ‘81’：功能不被支持。
- ‘82’：文件未找到。
- ‘86’：不正确的参数P1-P2。
- ‘87’： $L_c$ 与P1-P2不一致。

## 6.12 VERIFY 命令

### 6.12.1 定义和范围

VERIFY 命令启动从接口设备送入卡内的验证数据与卡内存储的引用数据(例如，口令)进行比较。

### 6.12.2 使用与安全的条件

安全状态可以被修改为比较的结果。不成功的比较可以记录在卡内(例如，为了限制使用引用数据的进一步企图数)。

### 6.12.3 命令报文

表61 VERIFY 命令APDU

CLA	按 5.4.1定义的
INS	‘20’
P1	‘00’(其他值为RFU)
P2	引用数据的限定符，见表62
$L_c$ 字段	空或后续数据字段的长度
数据字段	空或验证数据
$L_c$ 字段	空

表62 引用控制P2的编码

b8 b7 b6 b5 b4 b3 b2	含义
<u>          b1</u> 0 0 0 0 0 0 0	—没有信息被给出
<u>          0</u> 0 - - - - - - -	—全局引用数据 (例如 卡的口 令)
<u>          1</u> 1 - - - - - - -	—特定引用数据 (例如 DF特定口 令)
<u>      - x x - - - - -</u> - - - - x x x x x	‘00’ (其他值为RFU)
<u>      - - - - x x x x x</u> - - - - x x x x x	—引用数据号

注:

1)在VERIFY 命令无二义性地引用了保密数据的那些卡中, P2 = ‘00’ 被保留, 用来指示没有特定的限定符被使用.

2)例如, 引用数据号可以是一个口令号或一个短EF标识符.

3)当主体为空时, 命令既用来检索进一步允许的重试数(SW1-SW2 = ‘63CX’) 或用来校验是否不要求验证(SW1-SW2 = ‘9000’).

#### 6.12.4 响应报文(标称情况)

表63 VERIFY响应APDU

数据字段	空
SW1-SW2	状态字节

#### 6.12.5 状态条件

下列特定报警条件可能发生.

—SW1 = ‘63’, 同时SW2 =

· ‘00’:没有信息被给出(验证失败).

· ‘CX’:计数器(验证失败;‘X’表示进一步允许的重试数).

下列特定差错条件可能发生.

—SW1 = ‘69’, 同时SW2 =

· ‘83’:鉴别方法被阻塞.

· ‘84’:引用的数据无效.

—SW1 = ‘6A’, 同时SW2 =

· ‘86’:不正确的参数P1-P2.

· ‘88’:引用的数据未被找到.

### 6.13 INTERNAL AUTHENTICATE 命令

#### 6.13.1 定义和范围

INTERNAL AUTHENTICATE命令启动卡使用从接口设备发送来的询问数据和在卡内存的相关秘密(例如, 密钥)来计算鉴别数据.

当该相关秘密被连接到MF时, 命令可以用来鉴别整个卡.

当该相关秘密被连接到另一个DF时, 命令可以用来鉴别那个DF.

#### 6.13.2 使用与安全的条件

命令的成功执行可能受先前命令(例如, VERIFY, SELECT FILE)或选择(例如, 相关的秘密)的成功完成的支配.

当发布命令时，如果当前选择了密钥和算法，则该命令可以隐式地使用该密钥和算法。

已发出命令的次数可以记录在卡内，以便限制使用相关秘密或算法的进一步企图数。

### 6.13.3 命令报文

表64 INTERNAL AUTHENTICATE 命令APDU

CLA	按 5.4.1定义的
INS	'88'
P1	在卡内引用的算法
P2	引用的秘密，见表65
L <sub>c</sub> 字段	后续数据字段的长度
数据字段	鉴别相关的数据(例如，询问)
L <sub>r</sub> 字段	在响应中期望的字节最大数

P1 = '00' 表示没有信息被给出，在发出命令之前引用的算法为已知，或在数据字段中提供。

P2 = '00' 表示没有信息被给出，在发出命令之前引用的秘密为已知，或在数据字段中提供。

表65 引用控制P2的编码

b8	b7	b6	b5	b4	b3	b2	含义
b1							
0	0	0	0	0	0	0	—没有信息被给出
0							—全局引用数据 (例如, MF特定密钥)
0	-	-	-	-	-	-	—特定引用数据 (例如, DF特定密钥)
-							'00' (其他值为RFU)
1	-	-	-	-	-	-	—秘密的号
-	x	x	-	-	-	-	
-	-	x	-	x	x	x	

注: 例如，秘密的号可以是一个密钥号或一个短EF标识符。

### 6.13.4 响应报文(标称情况)

表66 INTERNAL AUTHENTICATE响应APDU

数据字段	鉴别相关的数据 (例如，对询问的响应)
SW1-SW2	状态字节

注: 响应报文可以包括对一步的应用安全功能有用的数据(例如: 随机数)。

### 6.13.5 状态条件

下列特定差错条件可能发生。

—SW1 = '69'，同时SW2 =

· '84': 引用的数据无效。

· '85': 使用的条件不被满足。

—SW1 = '6A'，同时SW2 =

· '86': 不正确的参数P1-P2。

· '88': 引用的数据未被找到。

## 6.14 EXTERNAL AUTHENTICATE 命令

### 6.14.1 定义和范围

EXTERNAL AUTHENTICATE 命令使用卡计算的结果(是或否)有条件地来更新安全状态, 而该卡的计算是以该卡先前发出(例如, 通过GET CHALLENGE命令)的询问、在卡内存储的可能的秘密密钥以及接口设备发送的鉴别数据为基础的。

### 6.14.2 使用与安全的条件

命令的成功执行要求从卡获得的最后询问是有效的。

不成功的比较可以被记录在卡内(例如, 为了限制使用引用数据的进一步企图数)。

### 6.14.3 命令报文

表67 EXTERNAL AUTHENTICATE 命令APDU

CLA	按 5.4.1定义的
INS	'82'
P1	在卡内引用的算法
P2	秘密的引用, 见表68
L <sub>c</sub> 字段	空或后续数据字段的长度
数据字段	空或鉴别相关的数据(例如, 对询问的响应)
L <sub>c</sub> 字段	空

P1 = '00' 表示没有信息被给出, 在发出命令之前引用的算法为已知, 或在数据字段中提供。

P2 = '00' 表示没有信息被给出, 在发出命令之前引用的秘密为已知, 或在数据字段中提供。

表68 引用控制P2的编码

b8 b7 b6 b5 b4 b3 b2 b1	含 义
0 0 0 0 0 0 0 0	—没有信息被给出 —全局引用数据 (例如, MF特定密钥)
0 - - - - - - 1	—特定引用数据 (例如, DF特定密钥)
- - - - - - - x x - - - -	'00' (其他值为RFU)
- - - - - - - - - x x x x x	—秘密的号

注:

1) 例如, 秘密的号可以是一个密钥号或一个短EF标识符。

2) 当主体为空时, 命令既可用于检索进一步允许的重试数(SW1-SW2 = '63CX')或用来校验是否不要求验证(SW1-SW2 = '9000')。

### 6.14.4 响应报文(标称情况)

表69 EXTERNAL AUTHENTICATE响应APDU

数据字段 SW1-SW2	空 状态字节
-----------------	-----------



#### 6.14.5 状态条件

下列特定报警条件可能发生。

—SW1 = ‘63’，同时SW2 =

· ‘00’：没有信息被给出(鉴别失效)。

· ‘CX’：计数器(鉴别失效；‘X’表示进一步允许的重试数)。

下列特定差错条件可能发生。

—SW1 = ‘67’，同时SW2 =

· ‘00’：错误的长度(L<sub>c</sub>字段不正确)。

—SW1 = ‘69’，同时SW2 =

· ‘83’：鉴别方法被阻塞。

· ‘84’：引用的数据无效。

· ‘85’：使用的条件不被满足(在上下文中命令不被允许)。

—SW1 = ‘6A’，同时SW2 =

· ‘86’：不正确的参数P1-P2。

· ‘88’：引用的数据未被找到。

### 6.15 GET CHALLENGE命令

#### 6.15.1 定义和范围

GET CHALLENGE 命令要求发出一个询问(例如，随机数)以便用于安全相关的规程(例如，EXTERNAL AUTHENTICATE 命令)。

#### 6.15.2 使用与安全的条件

询问至少对下一个命令是有效的。在本规范本部分未规定进一步的条件。

#### 6.15.3 命令报文

表70 GET CHALLENGE 命令APDU

CLA	按5.4.1定义的
INS	‘84’
P1-P2	‘0000’ (其他值为RFU)
L <sub>c</sub> 字段	空
数据字段	空
L <sub>w</sub> 字段	在响应中期望的最大长度

#### 6.15.4 响应报文(标称情况)

表71 GET CHALLENGE响应APDU

数据字段	询问
SWL-SW2	状态字节

#### 6.15.5 状态条件

下列特定差错条件可能发生。

—SW1 = ‘6A’，同时SW2 =

· ‘81’：功能不被支持。

· ‘86’：不正确的参数P1-P2。

## 6.16 MANAGE CHANNEL命令

### 6.16.1 定义和范围

MANAGE CHANNEL 命令打开和关闭逻辑信道。

开放功能打开了新逻辑信道，而不是基本逻辑信道。提供选项为了卡分配逻辑信道号，或为了将逻辑信道号供应给卡。

关闭功能显式地关闭逻辑信道，而不是基本逻辑信道。在成功关闭之后，该逻辑信道可加以重新使用。

### 6.16.2 使用与安全的条件

当由基本逻辑信道执行开放功能时，则在成功开放之后，MF应隐式地被选择作为当前DF，并且新逻辑信道的安全状态应和ATR之后的基本逻辑信道的安全状态相同。新逻辑信道的安全状态应和任何其他逻辑信道的安全状态分开。

当由不是基本逻辑信道的某一逻辑信道执行开放功能时，则在成功开放之后，曾发出命令的逻辑信道的当前DF应被选择作为当前DF，并且新逻辑信道的安全状态应和曾执行开放功能的逻辑信道的安全功能相同。

在成功的关闭功能之后，与该逻辑信道相关的安全状态被丢失。

### 6.16.3 命令报文

表72 MANAGE CHANNEL 命令APDU

<u>CLA</u>	<u>按5.4.1定义的</u>
<u>INS</u>	<u>'70'</u>
<u>P1</u>	<u>P1 = '00' 打开逻辑信道</u> <u>P1 = '80' 关闭逻辑信道(其他值为RFU)</u>
<u>P2</u>	<u>'00', '01', '02', '03' (其他值为RFU)</u>
<u>L<sub>c</sub>字段</u>	<u>空</u>
<u>数据字段</u>	<u>空</u>
<u>L<sub>c</sub>字段</u>	<u>'01', 如果P1-P2 = '0000'</u> <u>空, 如果P1-P2 = '0000'</u>

P1的位b8用来表示开放功能或关闭功能;如果b8为“0”，则MANAGE CHANNEL应打开逻辑信道，如果b8为“1”，则MANAGE CHANNEL应关闭逻辑信道。

对于开放功能(P1 = “00”)，P2的位b1和b2用来按照与类别字节(见本部分规范5.4.1)相同的方式来编制逻辑信道号;P2的其他位为RFU。

——当P2的b1和b2为空时，则卡将分配在数据字段的位b1和b2中返回的逻辑信道号。

——当P2的b1和/或b2不为空时，它编码某一逻辑信道号，而不是基本逻辑信道，则卡将打开外部分配的逻辑信道号。

### 6.16.4 响应报文(标称情况)

表73 MANAGE CHANNEL响应APDU

<u>数据字段</u>	<u>逻辑信道号, 如果P1-P2 = '0000'</u> <u>空, 如果P1-P2 = '0000'</u>
<u>SW1-SW2</u>	<u>状态字节</u>

### 6.16.5 状态条件

下列特定报警条件可能发生。

——SW1 = '62', 同时SW2 =

· ‘00’ : 没有信息被给出。

## 7 面向传输的行业间命令

对于遵循本规范本部分的所有卡而言，应该不强制要求支持本部分描述的所有命令或所支持命令的所有选项。

当要求进行国际交换时，卡的系统服务及相关命令的集合和选项应按照本规范本部分第9章中的定义使用。

表11提供了本规范本部分定义的命令概要。

安全报文交换(见5.6)对报文结构的影响不在本章中描述。

在7.X.5的每一条中所给出的差错和报警条件的列表不是穷举的(见5.4.5)。

### 7.1 GET RESPONSE 命令

#### 7.1.1 定义和范围

GET RESPONSE命令用于从卡发送至接口设备、用可用的协议不能传送的那一些的APDU(或APDU的一部分)。

#### 7.1.2 使用与安全的条件

无条件。

#### 7.1.3 命令报文

表74 GET RESPONSE 命令APDU

CLA	按 5.4.1定义的
INS	‘CO’
P1-P2	‘0000’ (其他值为RFU)
L <sub>c</sub> 字段	空
数据字段	空
L <sub>s</sub> 字段	在响应中期望的数据最大长度

#### 7.1.4 响应报文(标称情况)

如果L<sub>s</sub>字段仅包含“0”，则对于短的长度在不超过256，或者对于扩展的长度不超过65536的范围内，所有有效字节应被返回。

表75 “GET RESPONSE”响应APDU

数据字段	按照L <sub>s</sub> 的APDU(的一部分)
SWL-SW2	状态字节

#### 7.1.5 状态条件

下列特定正常处理可能发生。

—SW1 = ‘61’，同时SW2 =

· ‘XX’ : 正常处理:更多的数据字节是有效的(‘XX’表示在后续GET RESPONSE仍然有效的额外字节数)。

下列特定报警条件能发生。

—SW1 = ‘62’，同时SW2 =

· ‘81’ : 返回数据的一部分可能已损坏。

下列特定差错条件可能发生。

—SW1 = ‘67’，同时SW2 =

· ‘00’ : 长度错误(L<sub>s</sub>字段不正确)。

—SW1 = ‘6A’，同时SW2 =

- ‘86’:参数P1-P2不正确。
- SW1 = ‘6C’, 同时SW2 =
- ‘XX’:长度错误(L<sub>c</sub>字段错误;‘XX’表示正确的长度)。

## 7.2 ENVELOPE 命令

### 7.2.1 定义和范围

ENVELOPE命令用来发送那些不能由有效协议来发送的APDU, 或APDU的一部分, 或任何数据串。

注:对于SM的ENVELOPE命令的用法在附录F中示出。

### 7.2.2 使用与安全的条件

无条件。

### 7.2.3 命令报文

表76 ENVELOPE 命令APDU

CLA	按 5.4.1定义的
INS	‘C2’
P1-P2	‘0000’ (其他值为RFU)
L <sub>c</sub> 字段	后续数据字段的长度
数据字段	APDU(的一部分)
L <sub>c</sub> 字段	空或期望数据的长度

当对于发送数据串而言根据T = 0来使用ENVELOPE命令时, 在ENVELOPE命令APDU中的空数据字段意味着“数据串的开始”。

### 7.2.4 响应命令(标称情况)

表77 ENVELOPE响应APDU

数据字段	空或按照L <sub>c</sub> 的APDU(的一部分)
SWL-SW2	状态字节

注:状态字节属于ENVELOPE 命令所有。在ENVELOPE 命令的数据字段中所发送的命令的状态字节可能在ENVELOPE 命令响应的数据字段中找到。

### 7.2.5 状态条件

下列特定差错条件可能发生。

- SW1 = ‘67’, 同时SW2 =
- ‘00’:长度错误的(不正确的L<sub>c</sub>字段。)

## 8 历史字节

### 8.1 目的和一般结构

当按照本规范7816-3确定传输协议时, 历史字节告诉外界如何使用该卡。

历史字节数(至多15个字节)按本规范7816-3进行规定并编码。

历史字节所运载的信息也可以在ATR文件(默认EF标识符 = ‘2F01’)中找到。

如果存在, 历史字节可由3个数据字段组成:

- 一个必备的种类指示符(1个字节);
- 任选的压缩TLV数据对象;
- 一个有条件的状态指示符(13个字节)。

### 8.2 种类指示符(必备的)

种类指示符是第1个历史字节。如果种类指示符等于‘00’，‘10’或‘8X’，则历史字节的格式应符合本规范本部分。

表78 种类指示符的编码

值	含义
‘00’	状态信息应呈现在历史字节的结束处(不在TLV中)。
‘10’	在本部分规范8.5中规定
‘80’	状态信息(如果存在)包含在任选的压缩TLV数据对象中。
‘81’至‘8F’ 其他值	RFU 专有的

### 8.3 任选的压缩TLV数据对象

压缩TLV数据对象的编码可从适合于带有标记 = ‘4X’ 及长度 = ‘0Y’ 的RER-TLV数据对象的ASN.1基本编码规则(见ISO8825和附录D)推导出。这种数据对象的编码用‘XY’来代替，后面紧跟数据的Y字节。在本章中，‘X’系指标记号，‘Y’系指长度。

除本章中定义的数据对象外，历史字节还可以包含有本规范第5部分定义的数据对象。在这种情况下，在本规范第5部分中定义的标记和长度字段的编码应按上述要求进行修改。

当在本章中定义的压缩TLV数据对象出现在ATR文件中时，它们应按照ASN.1的基本编码规则进行编码(即标记 = ‘4X’，长度 = ‘0Y’)。

在本规范中未定义的所有应用类别标记被保留供ISO用。

#### 8.3.1 国家/发行者指示符

当存在时，该数据对象表示一个国家或一个发行者。

该数据对象可通过‘1Y’或‘2Y’来引入。

表79 国家/发行者指示符的编码

标记	长度	值
‘1’	可变	国家代码和国家数据
‘2’	可变	发行者标识号

标记‘1’后面紧跟着适合的长度(1个4位字节)以及紧跟着ISO3166定义的表示国家的3个数字。后面紧跟着(奇数个4位字节)的数据由相关的国家标准团体进行选择。

标记‘2’后面紧跟着适合的长度(1个4位字节)以紧跟着ISO7812第1部分定义的发行者标识号。如果发行者标识号包含有奇数个数字，则它应使用值为‘F’的4位字节正确地进行填充。

#### 8.3.2 卡服务数据

该数据对象表示为了支持第9章描述的服务在卡内有效的方法。

该数据对象通过‘31’来引入。

当该数据对象不存在时，卡仅支持显式应用选择。

表80 与应用无关的卡服务用的卡轮廓

<u>b8</u> <u>b7</u> <u>b6</u> <u>b5</u> <u>b4</u> <u>b3</u> <u>b2</u> <u>b1</u>	含义
<u>1</u> - - - - -	<u>—通过全DF名称的直接应用选择</u>
<u>-</u> 1 - - - - -	<u>—通过部分DF名称的选择 (见9.3.2)</u>
-	

续表80 与应用无关的卡服务用的卡轮廓

<u>b8</u> <u>b7</u> <u>b6</u> <u>b5</u> <u>b4</u> <u>b3</u> <u>b2</u> <u>b1</u>	含义
<u>-</u> - 1 - - - -	<u>数据对象有效</u> <u>—在DIR文件中</u> <u>—在ATR文件中</u>
<u>-</u> - - 1 - - -	
<u>-</u> - - - 1 - -	<u>文件I/O服务, 通过:</u> <u>READ RECORD命令</u> <u>READ BINARY命令</u>
<u>-</u> - - - 0 - -	
<u>-</u> - - - - × ×	<u>‘000’ (其他值为RFU)</u>
×	

注: DIR文件和ATR文件的内容可以给出关于选择方法的信息。

### 8.3.3 初始访问数据

该任选的数据对象允许检索在本规范中定义的数据对象串。该数据对象所检索的串称作“初始数据串”。

该数据对象通过‘41’，‘42’或‘45’来引入。

在本章中所描述的任何命令APDU被假定为是在复位应答之后所发送的第1个命令。因此，在该点的有效数据不是可以以后可检索的。

#### 8.3.3.1 长度 = ‘1’

当仅提供1个字节的信息时，它表示为了检索初始数据串而执行的命令长度。执行的命令是按如下结构的READ BINARY命令。

表81 当长度 = ‘1’ 时，命令的编码

<u>CLA</u>	<u>‘00’ (5.4.1)</u>
<u>INS</u>	<u>‘B0’</u>
<u>P1-P2</u>	<u>‘0000’</u>
<u>L<sub>c</sub>字段</u>	<u>空</u>
<u>数据字段</u>	<u>空</u>
<u>L<sub>e</sub>字段</u>	<u>初始访问数据中的值字段的第1个字节并且是唯一的字节(表示被读的字节数)</u>

#### 8.3.3.2 长度 = ‘2’

当提供2个字节的信息时，第1个字节表示文件结构(透明或记录)和被读的EF的短标识符。第2个字节表示为了检索初始数据串而执行的读命令长度。

表82 第1个字节的结构

b8	= 0面向记录的文件 = 1透明文件
b7-b6	'00' (其他值为RFU)
b5-b1	EF短标识符

当b8 = 0时，执行的命令是按如下结构的READ RECORD命令。

表83 当b8 = 0时，命令的编码

CLA	'00' (见5.4.1)
INS	'B2'
P1	'01'
P2	短EF标识符(来自初始访问数据的第1个字节)后面紧跟着b3-b2-b1 = 110
L <sub>c</sub> 字段	空
数据字段	空
L <sub>e</sub> 字段	初始访问数据中的值字段的第2个字节和最后一个字节(表示被读的字节数)

当b8 = 1时，执行的命令是按如下结构的READ BINARY命令。

表84 当b8 = 1时，命令的编码

CLA	'00' (见5.4.1)
INS	'B0'
P1	初始访问数据中的第1个字节的值
P2	'00'
L <sub>c</sub>	字段空
数据字段	空
L <sub>e</sub> 字段	初始访问数据中的值字段的第2个字节和最后一个字节(表示被读的字节数)

### 8.3.3.3 长度 = '5'

在初始访问数据中找到的值由执行的命令APDU组成。当执行时，该命令在其响应数据字段中提供初始数据串。

### 8.3.4 卡发行者数据

该数据对象是任选的并且为可变长度。结构和编码由卡发行者进行定义。该数据对象通过'5Y'来引入。

### 8.3.5 预先发行的数据

该数据对象是任选的并且为可变长度。结构和编码在本规范本部分中不予定义。它可以用来表示：

- 卡制造商
- 集成电路类型
- 集成电路制造商
- ROM掩模版本

—操作系统版本

该数据对象通过‘6Y’来引入。

8.3.6 卡能力

该数据对象是任选的并且为可变长度。其值字段由第1个软件功能表，或者由前面的2个软件功能表，或者由3个软件功能表组成。

该数据对象通过‘71’，‘72’或‘73’来引入。

表85 第1个软件功能表

<u>b8</u>	<u>b7</u>	<u>b6</u>	<u>b5</u>	<u>b4</u>	<u>b3</u>	<u>b2</u>	<u>含义</u>
<u>b1</u>							
<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>DF选择</u>
<u>—</u>							<u>—通过全DF名称</u>
<u>—</u>							<u>—通过部分DF名称</u>
<u>-</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>—通过路径</u>
<u>—</u>							<u>—通过文件标识符</u>
<u>-</u>	<u>-</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>—隐式地</u>
<u>—</u>							<u>EF管理</u>
<u>-</u>	<u>-</u>	<u>-</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>—所支持的短EF标识符</u>
<u>—</u>							<u>—所支持的记录号</u>
<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>—所支持的记录标识符</u>
<u>—</u>							
<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>1</u>	<u>-</u>	
<u>—</u>							
<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>1</u>	
<u>—</u>							
<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	
<u>—</u>							
<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	

表86示出了第2个软件功能表，它是数据编码类型。该数据编码类型也可以作为带有标记‘82’的文件控制参数中的第2个数据元而存在(见5.1.5中的表2)。

表86 第2个软件功能表  
(数据编码类型)

<u>b8</u>	<u>b7</u>	<u>b6</u>	<u>b5</u>	<u>b4</u>	<u>b3</u>	<u>b2</u>	<u>含义</u>
<u>b1</u>							
<u>-</u>	<u>x</u>	<u>x</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>写功能的行为</u>
<u>—</u>							<u>—一次写</u>
<u>-</u>	<u>0</u>	<u>0</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>—专有</u>
<u>—</u>							<u>—写‘或’</u>
<u>-</u>	<u>0</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>—写‘和’</u>
<u>—</u>							
<u>-</u>	<u>1</u>	<u>0</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	
<u>—</u>							
<u>-</u>	<u>1</u>	<u>1</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	
<u>—</u>							
<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	<u>-</u>	



<u>- - - - - × ×</u> <u>×</u>	<u>数据单元长度〔以4位字节 Nibble位单位〕</u> <u>(幂为2, 例如, 001 = 2 Nibble) (默认值 = 1个字节 byte)</u>
<u>- - - × × - -</u> <u>-</u>	<u>0...00...(其他值为RFU)</u>

表87示出了第三个软件功能表

表87 第三个软件功能表

<u>b8 b7 b6 b5 b4 b3 b2</u> <u>b1</u>	<u>含义</u>
<u>X - - - - -</u>	<u>0(1为RFU)</u>
<u>- 1 - - - - -</u>	<u>—扩充的L<sub>c</sub>和L<sub>e</sub>字段</u>
<u>- - - - -</u>	<u>0(1为RFU)</u>
<u>- - X - - - -</u>	<u>逻辑信道管理</u>
<u>- - - - -</u>	<u>—通过卡</u>
<u>- - - - X X - -</u>	<u>—通过接口设备</u>
<u>- - - - -</u>	<u>—无逻辑信道</u>
<u>- - - 1</u>	<u>0(1为RFU)</u>
<u>- - - 1</u>	<u>逻辑信道的最大数( = 2X + Y + 1)</u>
<u>- - - 0 0 - -</u>	
<u>- - - - - X -</u>	
<u>- - - - - X</u>	
<u>Y</u>	

#### 8.4 状态信息

状态信息由3个字节组成: 卡生存状态(1个字节)和2个状态字节SW1-SW2.

卡生存状态的值‘00’表示没有卡生存状态被提供, 值‘80’至‘FE’为专有的, 所有其他值为RFU.

SW1-SW2的值‘9000’表示按5.4.5定义的进行正常处理.

SW1-SW2的值‘0000’表示该状态未予表示.

如果种类指示符的值为‘80’, 则状态信息可以呈现在压缩TLV数据对象中. 在这种情况下, 标记号为‘8’. 当长度为‘1’时, 则值为卡生存状态. 当长度为‘2’时, 则值为SW1-SW2. 当长度为‘3’时, 则值为卡生存状态后紧跟着SW1-SW2. 长度的其他值被保留供ISO用.

#### 8.5 DIR数据引用

如果种类指示符为‘10’, 则后随字节为DIR数据引用. 该字节的编码及含义超出了本规范本部分的范围.

### 9 与应用无关的卡服务

#### 9.1 定义和范围

本章描述了与应用无关的卡服务, 其在下面的文本中被称作“卡服务”. 其目的

是提供在卡和接口设备之间的交换机制，它们(卡和接口设备)两者除了都遵循本规范外，它们彼此互不了解。

卡服务可通过下列内容的任何组合来支持。

—历史字节

—一个或多个保留EF的内容

—行业间命令的序列。

命令使用CLA='00' (见本部分规范5.4.1)，即，没有安全报文交换和基本逻辑信道。

只要一个应用在卡内已经被标识和选择，就没有必要遵循本章。应用该使用与本规范本部分兼容的其他机制来获得类似的功能。因此，这种解决方法可能不保证交换。

已定义了下列卡服务。

—卡标识服务—该服务允许接口设备标识卡以及如何处理。

—应用选择服务—该服务允许接口设备了解什么应用在卡(如果有)内活动以及如何选择和起内在卡的应用。

—数据对象检索服务—该服务允许检索在本规范本部分或其他部分中定义的数据对象。本章描述了仅用于行业间数据对象的标准机制。

—文件选择服务—该服务允许选择无名的DFs和EF。

—文件I/O服务—该服务允许访问存储在EF中的数据。

## 9.2 卡标识服务

该功能由卡根据其逻辑内容以及所有应用可能感兴趣的某些一般数据对象(例如，行业间数据对象)提供给外界的信息组成。称作“卡标识数据”的信息可由卡按历史字节以及可能直接在复位应答之后隐式选择的文件来给出。

对该文件的访问在初始访问数据信息中进行表示(见本部分规范8.3.3)。

如果历史字节的初始访问数据不指示读命令，则对执行命令的响应包含有卡标识数据。

## 9.3 应用选择服务

一个应用可在卡内被隐式地选择或通过其名称被显式地选择。

### 9.3.1 隐式应用选择

当应用在卡内被隐式地选择时，按本规范第5部分定义的应用标识符应在卡标识数据中进行表示。如果该标识符在卡标识数据中不存在，则它应存在于ATR文件中。

### 9.3.2 直接应用选择

多应用环境的卡应能实际地响应由SELECT FILE 命令所执行的直接应用选择，而该SELECT FILE 命令规定了应用标识符作为DF名称。

应用标识符应在命令APDU中完整地予以提供。在通过部分DF名称的应用选择的情况下，与所建议的名称相匹配的下一个应用可以被选择，并且全DF名称象带有标记'84'的文件控制参数那样可用于文件命令的响应报文(见5.1.5的表2)。执行的命令APDU如下。

表88 直接应用选择用的命令编码

<u>CLA</u>	<u>'00' (见5.4.1)</u>
<u>INS</u>	<u>'A4'</u>
<u>P1-P2</u>	<u>'0400'</u>
<u>L<sub>c</sub>字段</u>	<u>数据字段的字节长度</u>
<u>数据字段</u>	<u>全或部分DF名称</u>
<u>L<sub>e</sub>字段</u>	<u>存在, 仅包含了“0”</u>

#### 9.4 数据对象检索服务

与应用无关的国际交换所使用的数据对象在本规范本部分和其他部分中进行定义。

对那些数据对象的检索依赖于下列方法之一或两者:

—在卡标识数据中存在数据对象

—在DIR文件(路径 = '3F002F00')中存在数据对象或在ATR文件(路径 = '3F002F01')中存在数据对象。

通过间接的方法检索数据对象所必需的信息在本规范第6部分中进行定义。

#### 9.5 文件选择服务

当EF的路径为已知时, 被发出的SELECT FILE 命令数等于路径长度除以2, 减1(路径总是以当前DF开始)。

如果路径长度大于4个字节, 则直到路径的所有有效DF标识符都已被使用为止, 一个或多个SELECT FILE 命令应使用下列命令APDU来执行。

表89 使用文件标识符选择DF的命令的编码

<u>CLA</u>	<u>'00' (见5.4.1)</u>
<u>INS</u>	<u>'A4'</u>
<u>P1-P2</u>	<u>'0100'</u>
<u>L<sub>c</sub>字段</u>	<u>'02'</u>
<u>数据字段</u>	<u>DF标识符(来自路径的字节3和4)</u>
<u>L<sub>e</sub>字段</u>	<u>空</u>

最后一个选择并且可能是唯一的选择是带有下列命令APDU的EF选择

表90 选择EF的命令的编码

<u>CLA</u>	<u>'00' (见5.4.1)</u>
<u>INS</u>	<u>'A4'</u>
<u>P1-P2</u>	<u>'0200'</u>
<u>L<sub>c</sub>字段</u>	<u>'02'</u>
<u>数据字段</u>	<u>EF标识符(路径的最后2个字节)</u>
<u>L<sub>e</sub>字段</u>	<u>空</u>

#### 9.6 文件I/O服务

一旦用于行业间交换的文件已经被选择, 与交换相关的内容应通过下列命令APDU之一加以返回。

如果第1个软件功能表不存在, 或者不指示支持面向记录的命令, 则下列命令应予执行。

表91 读透明文件的命令的编码

<u>CLA</u>	<u>'00' (见5.4.1)</u>
<u>INS</u>	<u>'B0'</u>
<u>P1-P2</u>	<u>'0000'</u>
<u>L<sub>c</sub></u>	<u>字段空</u>
<u>数据字段</u>	<u>空</u>
<u>L<sub>s</sub>字段</u>	<u>存在, 仅包含了“0”</u>

· 如果第1个软件功能表指示了支持面向记录的命令, 则下列命令应予执行.

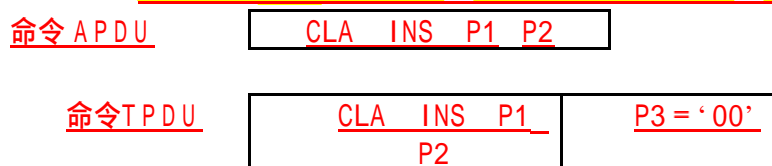
表92 读面向记录文件的命令的编码

<u>CLA</u>	<u>'00' (见5.4.1)</u>
<u>INS</u>	<u>'B2'</u>
<u>P1-P2</u>	<u>'0005'</u>
<u>L<sub>c</sub>字段</u>	<u>空</u>
<u>数据字段</u>	<u>空</u>
<u>L<sub>s</sub>字段</u>	<u>存在, 仅包含了“0”</u>

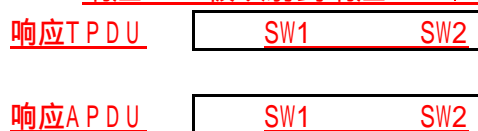
附 录 A  
(标准的附录)  
通过T = 0传输APDU报文

A.1 情况 1

通过将值‘00’分配给P3，把命令APDU映射到T = 0命令TPDU.



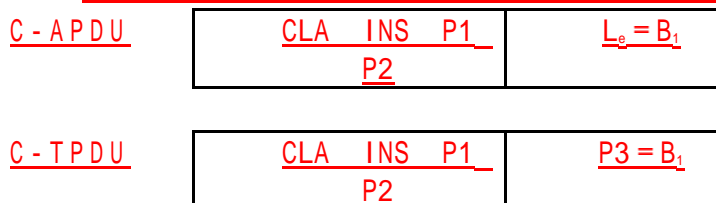
响应TPDU被映射到响应APDU，而没有任何变化.



A.2 情况 2 短的

在该情况下，L<sub>e</sub>的值从1至256，并且按字节B<sub>1</sub>进行编码(B<sub>1</sub> = ‘00’意味着最大值，即L<sub>e</sub> = 256).

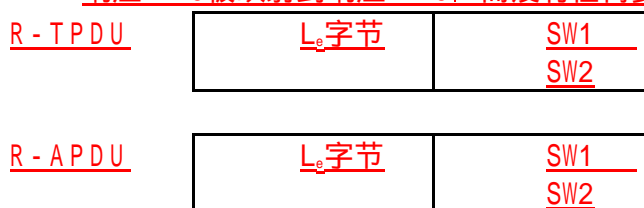
命令APDU被映射到T = 0命令TPDU，而没有任何变化.



按照接受的L<sub>e</sub>和按照处理的命令，响应TPDU被映射到响应APDU.

情况2 S . 1 —L<sub>e</sub> 被接受

响应TPDU被映射到响应APDU，而没有任何变化.

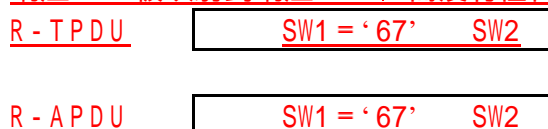


情况2 S . 2 —L<sub>e</sub> 明确地不被接受

如果长度是错误的，L<sub>e</sub>不能被不支持提供数据服务的卡所接受.

来自卡的响应TPDU表示卡放弃该命令被放弃是由于错误的长度:(SW1) = ‘67’.

响应TPDU被映射到响应APDU，而没有任何变化.



情况2 S.3 -  $L_e$  不被接受,  $L_a$  被指出

$L_e$  不能被卡所接受, 并且卡指出了有效长度  $L_a$ .

来自卡的响应TPDU指示该命令是由于错误的长度引起的, 并且指示正确的长度为  $L_a$ : (SW1) = '6C', 以及SW2编码了  $L_a$ .

如果传输系统不支持重新发出同一命令的服务, 它应将响应TPDU映射到响应APDU, 而没有任何变化.

R - TPDU 

SW1 = '6C'	SW2 = $L_a$
------------	-------------

R - APDU 

SW1 = '6C'	SW2 = $L_a$
------------	-------------

如果传输系统支持重新发出同一命令的服务, 它应重新发出将值  $L_a$  分配给参数 P3 的同一命令TPDU.

TPDU 

CLA	INS	P1	P3 = SW2
	P2		

响应TPDU由  $L_e$  字节后面紧跟着2个状态字节组成.

如果  $L_a$  小于  $L_e$  或等于  $L_e$ , 则响应TPDU被映射到响应APDU, 而没有任何变化.

R - TPDU 

$L_e$ 字节	SW1
	SW2

R - APDU 

$L_e$ 字节	SW1
	SW2

如果  $L_a$  大于  $L_e$ , 则响应TPDU被映射到主体的前面的  $L_e$  字节并且被映射到状态字节 SW1 - SW2.

R - TPDU 

$L_e$ 字节	SW1
	SW2

R - APDU 

$L_e (< L_a)$ 字节	SW1
	SW2

情况2 S.4 - SW1 - SW2 = '9XYZ', '9000' 除外

响应TPDU被映射到响应APDU, 而没有任何变化.

### A.3 情况 3 短的

在该情况下,  $L_c$  的值从1至255, 并且按字节  $B_1$  进行编码 ( $B_1$  '00').

命令APDU被映射到T = 0命令TPDU, 而没有任何变化.

C - APDU 

CLA	INS	P1	$L_c =$	$L_c$ 字
	P2		$B_1$	节

C - TPDU 

CLA	INS	P1	P3 =	$L_c$ 字
	P2		$B_1$	节

响应TPDU被映射到响应APDU, 而没有任何变化.

R - TPDU 

SW1 SW2
---------

R - APDU 

SW1 SW2
---------

#### A.4 情况 4 短的

在该情况下， $L_c$ 的值从1至255，并且按字节 $B_i$ 进行编码， $L_e$ 的值从1至256，并且按字节 $B_i$ 进行编码( $B_i = '00'$ 意味着最大值，即， $L_e = 256$ )。命令APDU被映射到截断主体的最后一个字节的 $T = 0$ 命令TPDU。

C - APDU 

CLA INS P1	$B_i =$	$L_e$ 字	$B_i$
P2	$L_e$	节	

C - TPDU 

CLA INS P1	$P3 =$	$L_e$ 字
P2	$B_i$	节

##### 情况4 S. 1—命令不被接受

来自卡的第1个响应TPDU表示卡放弃的命令:SW1 = '6X'，'61'除外。响应TPDU被映射到响应APDU，而没有任何变化。

R - TPDU 

SW1 = '6X' SW2
----------------

R - APDU 

SW1 = '6X' SW2
----------------

##### 情况4 S. 2—命令被接受

来自卡的第1个响应TPDU表示卡执行的命令:SW1-SW2 = '9000'。传输系统应通过分配值 $L_e$ 给参数P3将GET RESPONSE 命令TPDU发送给卡。

C - TPDU 

CLA INS = GET RESPONSE	$P3 =$
P1 P2	$B_i$

依赖于来自卡的第2个响应，传输系统应象上述情况2S.1，2S.2，2S.3和2S.4那样作出反应。

##### 情况4 S. 3—命令被接受，同时信息被增加

来自卡的第1个响应TPDU表示卡执行的命令，并且给出关于有效数据字节的长度:SW1 = '61'，并且SW2编码 $L_x$ 。

传输系统应通过分配最小 $L_x$ 和 $L_e$ 给参数P3将GET RESPONSE 命令TPDU发送给卡。

C - TPDU 

CLA INS = GET RESPONSE	$P3 = \min(L_e,$
P1 P2	$L_x)$

第2个响应TPDU被映射到响应APDU，而没有任何变化。

R - TPDU 

P3字节	SW1
	SW2

R - APDU 

P3字节	SW1
	SW2

情况4 S. 4—SW1-SW2 = '9XYZ'，'9000'除外  
响应TPDU被映射到响应APDU，而没有任何变化。

### A.5 情况 2 扩充的

在该情况下， $L_e$  的值从1至65536，并且按3个字节进行编码： $(B_1) = '00'$ ， $(B_2 B_3) =$ 任何值( $B_2$ 和 $B_3$ 的值为'0000'意味着最大值，即 $L_e = 65536$ )。

C - APDU	<table border="1"> <tr> <td>CLA</td> <td>INS</td> <td>P1</td> </tr> <tr> <td colspan="3">P2</td> </tr> </table>	CLA	INS	P1	P2			$B_1 = '00'$ $B_2 B_3 = L_e$
CLA	INS	P1						
P2								

情况2 E.1— $L_e \leq 256$ ， $B_1 = '00'$ ， $B_2 B_3$ 从'0001'至'0100'。

命令APDU应通过分配 $B_3$ 的值给参数P3而被映射到命令TPDU。传输系统的处理应按照情况2S进行。

C - TPDU	<table border="1"> <tr> <td>CLA</td> <td>INS</td> <td>P1</td> </tr> <tr> <td colspan="3">P2</td> </tr> </table>	CLA	INS	P1	P2			$P3 = B_3$
CLA	INS	P1						
P2								

情况2 E.2— $L_e > 256$ ， $B_1 = '00'$ ， $B_2 B_3 = '0000'$ 或从'0101'至'FFFF'。

命令APDU应通过分配值'00'给参数P3而被映射到命令TPDU。

C - TPDU	<table border="1"> <tr> <td>CLA</td> <td>INS</td> <td>P1</td> </tr> <tr> <td colspan="3">P2</td> </tr> </table>	CLA	INS	P1	P2			$P3 = '00'$
CLA	INS	P1						
P2								

a)如果来自卡的第1个响应TPDU表示卡放弃的命令是由于错误的长度( $SW1 = '67'$ )引起的，则响应TPDU应被映射到响应APDU，而没有任何变化。

R - TPDU	$SW1 = '67'$ $SW2$
----------	--------------------

R - APDU	$SW1 = '67'$ $SW2$
----------	--------------------

b)如果来自卡的第1个响应TPDU表示被放弃的命令是由于错误的长度引起的，并且表示正确的长度为 $L_s$ ( $SW1 = '6C'$ 和 $SW2 = L_s$ )，则传输系统应象情况2S.3描述的那样完成处理。

c)第1个响应TPDU是256个数据字节后面紧跟着 $SW1 - SW2 = '9000'$ ，这就意味着卡具有不大于256个数据字节，和/或不支持GET RESPONSE命令。则传输系统应将响应TPDU映射到响应APDU，而没有任何变化。

R - TPDU	256字节	$SW1 = '90'$ $SW2 = '00'$
----------	-------	---------------------------

R - APDU	256字节	$SW1 = '90'$ $SW2 = '00'$
----------	-------	---------------------------

d)如果来自卡的第1个响应TPDU或后续的响应TPDU为 $SW1 = '61'$ ，则 $SW2$ 编码 $L_x$ ，而该 $L_x$ 是从卡得到的额外字节量( $SW2$ 的值为'00'表示256个额外字节或更多)，传输系统应计算 $L_m = L_e -$ (先前收到的响应TPDU(s)的主体长度之和)，以获得被卡检索的其余字节量。

如果 $L_m = 0$ ，则传输系统应将所有收到的响应TPDU的主体与最后收到的响应TPDU的尾标一起并置到响应APDU。

如果 $L_m > 0$ ，则传输系统应通过分配最小 $L_x$ 和 $L_m$ 给参数P3来发出GET RESPONSE命令。来自卡的相应响应TPDU应进行处理：

—如果 $SW1 = '61'$ ，根据情况d)。



—如果SW1 = '90'，当L<sub>c</sub> = 0时，如上所述。

### A.6 情况 3 扩充的

在该情况下，L<sub>c</sub>的值从1至65536，并且按3个字节进行编码：(B<sub>1</sub>) = '00'，

(B<sub>2</sub> B<sub>3</sub>) = '0000'。

C - APDU	CLA	INS	P1	B <sub>1</sub> = '00' B <sub>2</sub> B <sub>3</sub> = L <sub>c</sub>	L <sub>c</sub> 字节
U		P2			

情况3E.1—0 < L<sub>c</sub> < 256，B<sub>1</sub> = '00'，B<sub>2</sub> = '00'，B<sub>3</sub> = '00'。

命令APDU可通过分配B<sub>3</sub>的值给参数P<sub>3</sub>而被映射到命令TPDU。

C - APDU	CLA	INS	P1	P3 = B <sub>3</sub>	L <sub>c</sub> 字节
U		P2			

在该情况下，L<sub>c</sub>的值从1至255，并且按1个字节进行编码。

响应TPDU被映射到响应APDU，而没有任何变化。

R - TPDU	SW1	SW2
U		

R - APDU	SW1	SW2
U		

情况3E.2—L<sub>c</sub> > 255，B<sub>1</sub> = '00'，B<sub>2</sub> = '00'，B<sub>3</sub> = 任何值。

如果传输系统不支持ENVELOPE 命令，则它应返回差错响应APDU，意味着长度是错误的：SW1 = '67'。

R - TPDU	SW1 = '67'	SW2
U		

R - APDU	SW1 = '67'	SW2
U		

如果传输系统支持ENVELOPE 命令，它应将APDU分解成小于256的长度段，并且将这些连续的段发送到连续ENVELOPE 命令TPDUs的主体。

C - TPDU	CLA	INS = ENVELOPE	P1	P/3	P3字节
U		P2			

来自卡的第1个响应TPDU表示卡不支持ENVELOPE 命令(SW1 = '6D')，该TPDU应被映射到响应APDU，而没有任何变化。

R - TPDU	SW1 = '6D'	SW2
U		

R - APDU	SW1 = '6D'	SW2
U		

如果来自卡的第1个响应TPDU表示卡支持ENVELOPE 命令(SW1-SW2 = '9000')，传输系统应按需要发送进一步的ENVELOPE 命令。

R - TPDU	SW1-SW2 = '9000'
----------	------------------

U

--

C - T P D

<u>CLA</u> <u>INS = ENVELOPE</u> <u>P1</u>		<u>P3</u>	<u>P3字</u>
<u>P2</u>			<u>节</u>

对应于最后一个ENVELOPE 命令的响应TPDU被映射到响应APDU，而没有任何变化。

R - T P D

<u>SW1</u>	<u>SW2</u>
------------	------------

R - A P D

<u>SW1</u>	<u>SW2</u>
------------	------------

### A.7 情况 4 扩充的

在该情况下， $L_c$ 的值从1至65536，并且按3个字节进行编码： $(B_1) = '00'$ ， $(B_2 B_3) = '0000'$ ，以及 $L_c$ 的值从1至65536，并且按2个字节进行编码： $(B_{L-1} B_L) =$ 任何值 $(B_{L-1}$ 和 $B_L$ 的值为‘0000’意味着是最大值，即， $L_c = 65536)$ 。

C - A P D

<u>CLA</u> <u>INS</u> <u>P1</u>	<u><math>B_1 = '00'</math> <math>B_2 B_3 = L_c</math></u>	<u><math>L_c</math>字</u>	<u><math>B_{L-1} B_L</math></u>
<u>P2</u>		<u>节</u>	<u><math>= L_c</math></u>

情况4E.1— $L_c < 256$ ， $B_1 = '00'$ ， $B_2 = '00'$ ， $B_3 = '00'$ 。

命令APDU可通过截断最后2个字节 $B_{L-1}$ 和 $B_L$ 以及通过分配 $B_3$ 的值给参数P3而被映射到命令TPDU。

C - T P D

<u>CLA</u> <u>INS</u> <u>P1</u>	<u><math>P3 = B_3</math></u>	<u><math>L_c</math>字节</u>
<u>P2</u>		

在该情况下， $L_c$ 的值从1至255字节，并且按1个字节进行编码。

a) 如果在来自卡的第1个响应TPDU中 $SW1 = '6X'$ ，则响应TPDU被映射到响应APDU，而没有任何变化。

R - T P D

<u>SW1 = '6X'</u>	<u>SW2</u>
-------------------	------------

R - A P D

<u>SW1 = '6X'</u>	<u>SW2</u>
-------------------	------------

b) 如果在来自卡的第1个响应TPDU中 $SW1 = '90'$ ，则，如果 $L_c < 257$  ( $B_{L-1} B_L$ 的值为‘0001’至‘0100’)则传输系统应通过分配 $B_L$ 的值给参数P3来发送GET RESPONSE 命令TPDU。传输系统的后续处理应按照上述情况2S.1、2S.2、2S.3和2S.4进行。

如果 $L_c > 256$  ( $B_{L-1} B_L$ 的值从‘0000’或大于‘0100’)，则传输应通过分配值‘00’给参数P3来发送GET RESPONSE 命令TPDU。传输系统的后续处理应按照上述情况2E.2进行。

c) 如果在来自卡的第1个响应TPDU中 $SW1 = '61'$ ，则传输系统应如上述情况2E.2 d)规定的那样继续进行。

情况4E.2— $L_c > 255$ ,  $B_1 = '00'$ ,  $B_2 = '00'$ ,  $B_3 = \text{任何值}$ .

传输系统应按照上述情况3E.2继续进行,直到已经完整地将命令APDU发送到卡为止.然后,它应如上述情况4E.1 a)和c)规定的那样继续进行.

## 附 录 B

(标准的附录)

### 通过T=1传输APDU报文

—

#### B.1 情况 1

命令APDU被映射到 块的信息字段,而没有任何变化.

命令APDU 

CLA	INS	P1	P2
-----	-----	----	----

信息字段 

CLA	INS	P1	P2
-----	-----	----	----

在响应中收到的 块的信息字段被映射到响应APDU，而没有任何变化。

信息字段 

SW1	SW2
-----	-----

命令APDU 

SW1	SW2
-----	-----

### B.2 情况 2(短的和扩充的)

命令APDU被映射到 块的信息字段，而没有任何变化。

C - APDU 

CLA	INS	P1	P2	L <sub>c</sub> 字段
-----	-----	----	----	-------------------

信息字段 

CLA	INS	P1	P2	L <sub>c</sub> 字段
-----	-----	----	----	-------------------

响应APDU由：

—在响应中收到的 块的信息字段组成，

——或者在响应中收到的连续 块的顺序连接的信息字段组成。这些块应予以链接。

信息字段 

数据字段	SW1-SW2
------	---------

或者顺序连接的信息字段

字段数据 .....
------------

.....
-------

..... 字 段	SW1-S W2
--------------	-------------

R - APDU 

数据字段	SW1-S W2
------	-------------

### B. 3 情况 3(短的和扩充的)

命令APDU没有任何变化地被映射到：

—某一 块的信息字段，

——或应链接的连续 块的顺序连接的信息字段。

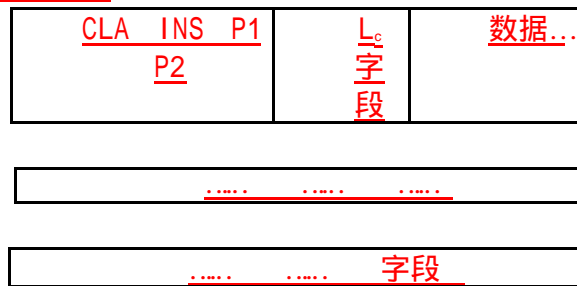
C - APDU 

CLA	INS	P1	P2	L <sub>c</sub> 字段	数据字 段
-----	-----	----	----	-------------------	----------

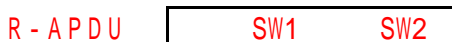
信息字  
段 

CLA	INS	P1	P2	L <sub>c</sub> 字段	数据字 段
-----	-----	----	----	-------------------	----------

或者顺序连接的信息字段



在响应中收到的 块的信息字段被映射到响应APDU，而没有任何变化。

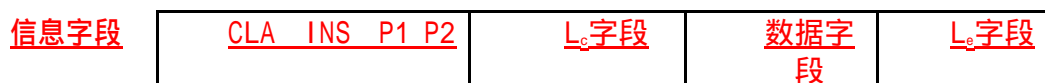


**B.4 情况 4(短的和扩充的)**

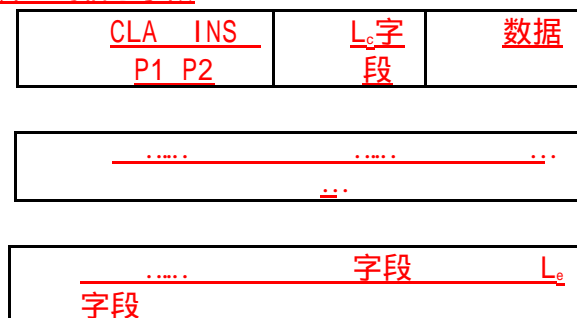
命令APDU没有任何变化地被映射到：

—某一 块的信息字段，

—或者应链接的连续 块的顺序连接的信息字段。



或者顺序连接的并置的信息字段



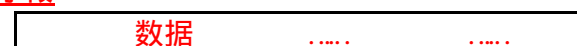
响应APDU由

—在响应中收到的 块的信息字段组成，

—或者在响应中收到的连续 块的顺序连接的信息字段组成。这些块应予以链接。



或者并置的信息字段



.....
-------

.....	字段	SW1-SW2
-------	----	---------

R-APDU

数据字段	SW1-SW2
------	---------

附录 C  
(提示的附录)  
记录指针管理

C.1 情况 1

情况1涉及在选择功能(显式的或隐式的)之后所发生的第1个命令。当前记录指针(CP)是未定义的。

<u>命令</u> <u>(READ RECODE)</u>	<u>记录</u> <u>(在响应中)</u>	<u>CP的位置</u> <u>(在命令之后)</u>
<u>下一个(id = aa)</u>	<u>第1个带有id = aa</u> <u>如果未找到, 则差错</u>	<u>记录读出</u> <u>未定义</u>
<u>先前一个(id = bb)</u>	<u>最后一个带有id = bb</u> <u>如果未找到, 则差错</u>	<u>记录读出</u> <u>未定义</u>
<u>第1个(id = cc)</u>	<u>第1个带有id = cc</u> <u>如果未找到, 则差错</u>	<u>记录读出</u> <u>未定义</u>
<u>最后一个(id = dd)</u>	<u>最后一个带有id = dd</u> <u>如果未找到, 则差错</u>	<u>记录读出</u> <u>未定义</u>
<u>下一个(id = 00)</u>	<u>第1个</u>	<u>记录读出</u>
<u>先前一个(id = 00)</u>	<u>最后1个</u>	<u>记录读出</u>
<u>第1个(id = 00)</u>	<u>第1个</u>	<u>记录读出</u>
<u>最后一个(id = 00)</u>	<u>最后一个</u>	<u>记录读出</u>
<u>记录 # 00</u>	<u>差错</u>	<u>未定义</u>
<u>记录 # ee</u>	<u># ee</u> <u>如果未找到, 则差错</u>	<u>未定义</u> <u>未定义</u>
<u>P1 = '00', P2 = XXXX</u> <u>X101</u>	<u>差错</u>	<u>未定义</u>
<u>v = '00', P2 = XXXX X110</u>	<u>差错</u>	<u>未定义</u>
<u># jj, P2 = XXXX X101</u>	<u># jj至最后一个</u> <u>如果 # jj未找到, 则差错</u>	<u>未定义</u> <u>未定义</u>
<u># kk, P2 = XXXX X110</u>	<u>最后一个至 # kk</u> <u>如果 # kk未找到, 则差错</u>	<u>未定义</u> <u>未定义</u>

C2 情况 2

情况2涉及后续命令。当前记录指针(CP)是被定义的。

<u>命令</u> <u>(READ RECODE)</u>	<u>记录</u> <u>(在响应中)</u>	<u>CP的位置</u> <u>(在命令之后)</u>
<u>下一个(id = aa)</u>	<u>下一个带有id = aa</u> <u>如果没有下一个, 则差错</u>	<u>记录读出</u> <u>未变化</u>
<u>先前一个(id = bb)</u>	<u>先前一个带有id = bb</u>	<u>记录读出</u>

	<u>如果没有下一个，则差错</u>	<u>未变化</u>
<u>第1个(id = cc)</u>	<u>第1个带有id = cc</u> <u>如果未找到，则差错</u>	<u>记录读出</u> <u>未变化</u>
<u>最后一个(id = dd)</u>	<u>最后1个带有id = dd</u> <u>如果未找到，则差错</u>	<u>记录读出</u> <u>未变化</u>
<u>下一个(id = 00)</u>	<u>CP + 1</u> <u>如果CP = 最后一个，则差错</u>	<u>先前CP + 1</u> <u>未变化</u>
<u>先前的一个(id = 00)</u>	<u>CP - 1</u> <u>如果CP = 第1个，则差错</u>	<u>先前CP - 1</u> <u>未变化</u>
<u>第1个(id = 00)</u>	<u>第1个第1个记录</u>	<u>读出</u>
<u>最后1个(id = 00)</u>	<u>最后1个最后一个</u>	<u>记录</u>
<u>记录 # 00</u>	<u>CP</u>	<u>未变化</u>
<u>记录 # ee</u>	<u># ee</u> <u>如果未找到，则差错</u>	<u>未变化</u> <u>未变化</u>
<u>P1 = '00'， P2 = XXXX</u> <u>X101</u>	<u>CP至最后1个</u>	<u>未变化</u>
<u>P1 = '00'， P2 = XXXX</u> <u>X110</u>	<u>最后1个至CP</u>	<u>未变化</u>
<u># jj， P2 = XXXX X101</u>	<u># jj至最后1个</u> <u>如果 # jj未找到，则差错</u>	<u>未变化</u> <u>未变化</u>
<u># kk， P2 = XXXX X110</u>	<u>最后1个至 # kk</u> <u>如果 # kk未找到，则差错</u>	<u>未变化</u> <u>未变化</u>

附 录 D  
(提示的附录)  
使用ANS.1基本编码规则

**D.1 BER-TLV 数据对象**

每个BER-TLV数据对象(见ISO8825)应由2或3个连续字段组成。

—标记字段T由一个或多个连续字节组成，它编码了类别、类型和编号。

—长度字段由一个或多个连续字节组成，它编码了整数L。

—如果L不为空，则值字段V由L个连续字节组成。如果L为空，则数据对象为空：没有值字段。

本规范既不使用‘00’作为标记值，也不使用‘FF’作为标记值。

注：在BER-TLV数据对象之前、之间或之后，没有任何含义的‘00’或‘FF’字节可以出现(例如，由于擦除的或修改的TLV编码数据对象所引起)。

**D.2 标记字段**

标记字段中的引导字节的位b8和b7应编码标记类别，即，数据对象的类别。

—b8-b7 = 00引入全局类别的标记。

—b8-b7 = 01引入应用类别的标记。

—b8-b7 = 10引入上下文特定类别的标记。

—b8-b7 = 11引入专用类别的标记。

标记字段中的引导字节的位b6应编码标记类型，即，数据对象的类型。

—b6 = 0引入原始数据对象。

—b6 = 1引入结构化数据对象。

如果引导字节的位b5至b1不是都置为“1”，则它们应编码等于标记编号的一个整数，而该标记编号位于从0至30的范围内。然后标记字段由单个字节组成。

另一种方法(在引导字段中b5到b1置为“1”)标记字段应继续1个或多个后续字节。

—每个后续字节的b8应置为“1”，除非它是最后一个后续字节。

—第1个后续字节的位b7至b1应不是都置为“0”。

—第1个后字节的位b7至b1，后面紧跟着每个进一步的后续字节的位b7至b1，一直到并包括最后一个后续字节的位b7至b1都应编码等于标记编号的一个整数(因此是精确的正数)。

**D.3 长度字段**

在短形式中，长度字段由单个字节组成，其中位b8应置为“0”，并且位b7至b1



应编码等于值字段中的字节数的一个整数，因此从0至127的任何长度可以利用1个字节来编码。

在长形式中，长度字段由一个引导字节组成，其中位b8应置为“1”，并且位b7至b1应不是全相同，因此，编码的一个正整数等于在长度字段中的后续字节数。那些后续字节应编码等于在值字段中的字节数的一个整数。因此在APDU限制(高达65536)范围内的任何长度可以利用3个字节来编码。

注:本规范不使用ASN.1基本编码规则所规定的不定长度(见ISO8825)。

#### **D.4 值字段**

在本规范本部分中，某些原始BER-TLV数据对象的值字段由0个、1个或多个简单TLV数据对象组成。

任何其他原始BER-TLV数据对象的值字段由通过数据对象规范所确定的0个、1个或多个数据元组成。

每个结构化BER-TLV数据对象的值字段由0个、1个或多个BER-TLV数据对象组成。

附 录 E  
(提示的附录)  
卡轮廓的举例

E.1 引言

本附录定义了许多卡轮廓，以指导应用设计者选择命令用于其应用中。这些轮廓也可用来帮助规定卡内要求的特征。卡轮廓可以进行组合。

E.2 轮廓 M

该轮廓的卡至少具有如下特征和命令。

—文件结构

- 透明结构
- 带有固定长度记录的线性结构

—命令

- READ BINARY和UPDATE BINARY，同时  
P1, b8 = 0,  
长度高达256个字节。
- READ RECODE和UPDATE RECORD，同时  
P2, b8至b4 = 0,  
P2, b3 = 1,  
P2, b3 b2 b1 = 000, 001, 010或011, 并且P1 = 0.
- SELECT FILE，同时  
P1-P2 = '0000'.
- VERIFY，同时  
P1-P2 = '0001' 或'0002'.
- INTERNAL AUTHENTICATE，同时  
P1-P2 = '0000'.

E.3 轮廓 N

该轮廓和M相同，加上在SELECT FILE 命令中的附加选项P = '04'.

E.4 轮廓 O

该轮廓的卡至少具有如下特征和命令。

—文件结构

- 透明结构

- 带有固定长度记录的线性结构。
- 带有可变长度记录的线性结构。
- 带有固定长度记录的循环结构。

#### —命令

- READ BINARY、WRITE BINARY和UPDATE BINARY，同时  
P1, b8 = 0,  
长度高达256个字节。
- READ RECODE、WRITE RECORD和UPDATE RECORD，同时  
P2, b8至b4 = 0,  
P2, b3 = 1,  
P2, b3 b2 b1 = 000, 001, 010或011, 并且P1 = 0.
- APPEND RECORD，同时  
P1-P2 = '0000'.
- SELECT FILE，同时  
P1 = '00'、'01'、'02'、'03'、'04'或'09'，  
P2 = '00'.
- VERIFY，同时  
P1-P2 = '0001'或'0002'.
- INTERNAL AUTHENTICATE，同时  
P1-P2 = '0000'.
- EXTERNAL AUTHENTICATE，同时  
P1-P2 = '0000'.
- GET CHALLENGE，同时  
P1-P2 = '0000'.

### E.5 轮廓 P

该轮廓的卡至少有下列特征和命令。

#### —文件结构

- 透明结论

#### —历史字节

- 卡服务数据( = '3188')。
- 初始访问数据( = '4164')。

#### —命令

- READ BINARY和UPDATE BINARY，同时  
P1, b8 = 0,  
长度高达64个字节。
- SELECT FILE，同时  
P1-P2 = '0400'.
- VERIFY，同时  
P1-P2 = '0001'或'0002'.
- INTERNAL AUTHENTICATE，同时  
P1-P2 = '0000'.

### E.6 轮廓 Q

该轮廓的卡至少具有如下特征和命令。

—历史字节

· 初始访问数据( = ‘45’ -得到)。

· 卡能力( = ‘7180’ )。

—安全报文交换

—命令

· GET DATA和PUT DATA，同时

标记在P1-P2中

· SELECT FILE，同时

P1-P2 = ‘0401’、‘0402’ 或‘0403’。

· VERIFY，同时

P1 = ‘00’

· INTERNAL AUTHENTICATE

· EXTERNAL AUTHENTICATE

· GET CHALLENGE。

附 录 F  
( 提示的附录)  
用的安全报文交换

F.1 缩略语

下列编略语适用于本附录。

CC 密码检验和

CG 密码

CH 命令循环(CLA INS P1 P2)

CR 控制引用

FR 文件引用

KR 密钥引用

L 长度

PB 填充字节(‘80’后面紧跟着0至K-1次‘00’，其中K为块长度)

P1 填充指示符字节

PV 简明值

RD 响应描述符

T 标记

\_\_\_\_\_ 并置

对于所有举例，CLA表示通过合适的值(‘0X’、‘8X’、‘9X’或AX)使用的安全报  
文交换，其中CLA的b4置为“1”(见本部分规范5.4.1和表9)。

F.2 使用密码校验和

为了表4和图4中定义的4种情况而示出了使用的密码校验和(见本部分规范  
5.6.3.1)。

—情况1—没有数据，没有数据

命令数据字段 = Tcc Lcc CC

CC(CLA中b3 = 1)所覆盖的数据 = 第1个并且是唯一的一个数据块 = CH PB

情况1的命令被交换成情况3的命令。

—情况2—没有数据，有数据

命令数据字段 = Tcc Lcc CC

CC(CLA中b3 = 1)所覆盖的数据 = 第1个并且是唯一的一个数据块 = CH PB

响应数据字段 = T<sub>PV</sub>(b1 = 1) L<sub>PV</sub> PV Tcc Lcc CC

CC所覆盖的数据 = 数据块 = T<sub>PV</sub> (b1 = 1) L<sub>PV</sub> PV PB

—情况3 . a—有数据，没有数据

命令数据字段 = T<sub>PV</sub>(b1=1) L<sub>PV</sub> PV T<sub>CC</sub> L<sub>CC</sub> CC

CC(CLA中b3 = 0)所覆盖的数据 = 数据块 = T<sub>PV</sub>(b1 = 1) L<sub>PV</sub> PV PB

—情况3 . b—有数据，没有数据

命令数据字段 = T<sub>PV</sub>(b1 = 0) L<sub>PV1</sub> PV1 T<sub>PV2</sub>(b1 = 1) L<sub>PV2</sub> PV2 T<sub>CC</sub> L<sub>CC</sub> CC

CC(CLA中b3 = 1)所覆盖的数据 = 数据块 = CH PB T<sub>PV2</sub>(b1 = 1) L<sub>PV2</sub> PV2 PB

—情况4—有数据，有数据

命令数据字段 = T<sub>PV</sub>(b1 = 1) L<sub>PV</sub> PV T<sub>CC</sub> L<sub>CC</sub> CC

CC(CLA中b3 = 0)所覆盖的数据 = 数据块 = T<sub>PV</sub>(b1 = 1) L<sub>PV</sub> PV PB

响应数据字段 = T<sub>PV</sub>(b1 = 1) L<sub>PV</sub> PV T<sub>CC</sub> L<sub>CC</sub> CC

CC所覆盖的数据 = 数据块 = T<sub>PV</sub>(b1 = 1) L<sub>PV</sub> PV PB

### **F.3 使用密码**

密码的使用〔见本部分规范5.6.4〕如下所示，有填充与不填充两种情况。

—情况a—没有在BER-TV中编码的简单数据

命令数据字段 = T<sub>CG</sub> L<sub>CG</sub> PI CG

CG传送的数据 = 数据块 = 没有BER-TV编码的数据和填充字节，如果在PI中被指

出

—情况b—在BER-TV中编码的简单数据

命令数据字段 = T<sub>CG</sub> L<sub>CG</sub> CG

CG传送的数据 = 隐藏字节串 = BER-TV数据对象〔按运算法则填充和它的操作模

式〕

### **F.4 使用控制引用**

控制引用的用法〔见本部分规范5.6.5.1〕如下。

命令数据字段 = T<sub>CR</sub> L<sub>CR</sub> CR

此处CR = T<sub>FR</sub> L<sub>FR</sub> FR T<sub>KR</sub> L<sub>KR</sub> KR

### **F.5 使用应答描述符**

应答描述符的用法〔见本部分规范5.6.5.2〕如下。

命令数据字段 = T<sub>RD</sub> L<sub>RD</sub> RD

此处RD = T<sub>PV</sub> '00' T<sub>CC</sub> '00'

数据字段响应 = T<sub>PV</sub> L<sub>PV</sub> PV T<sub>CC</sub> L<sub>CC</sub> CC

### **F.6 使用 ENVELOPE 命令**

ENVELOPE 命令的用法〔见本部分规范 7.2〕如下。

命令数据字段 = T<sub>CG</sub> || L<sub>CG</sub> || PI || CG

CG 传送的数据 = 由 CH 开始的命令 APDU 和根据 PI 的填充字节

数据字段响应 = T<sub>CG</sub> || L<sub>CG</sub> || PI || CG

CG 传送的数据 = 响应 APDU 和根据 PI 的填充字节

