1 概 述

1.1 产品简介

非接触式IC芯片产品FMXC50由射频通讯接口、安全控制单元和8K位EEPROM组成，符合ISO/IEC14443-A国际标准。适用于各种证件、电子钱包、自动收费系统和公共交通自动售检票系统等领域。

1.2 产品特性

- 射频通讯接口
  - 符合 ISO/IEC14443-A 标准
  - 无线传送数据和能量，无需电池
  - 采用半双工通讯方式
  - 操作距离可达 10cm（需要天线配合）
  - 操作频率 13.56MHz±7KHz
  - 数据传输速率 106kbps
  - 抗冲突：配合应用系统，支持处理同一时间多张卡进入场区的情况
  - 典型票务交易时间小于 100ms，包括备份管理

- 存储器
  - 8k 位 EEPROM
  - EEPROM 分为 16 个扇区，每扇区分为 4 个块，每个块包含 16 字节，每个字节包含 8 位
  - 用户可对每个块单独定义访问权限
  - 数据保持时间至少 10 年
  - 数据擦写次数至少 10 万次

- 安全特性
  - 数据完整性：包含 16 位 CRC、奇偶校验、位编码和位计数
  - 三重交互安全认证
  - 每个扇区均可独立设置两个密钥用于支持带层次密钥的多应用
  - 每个芯片均有唯一序列号
  - 芯片交付时有传输密钥对 EEPROM 访问进行保护

1.3 产品形式

- Wafer
- 芯片
- 模块
  - COB 封装
  - XOA2 封装

1.4 管脚信息

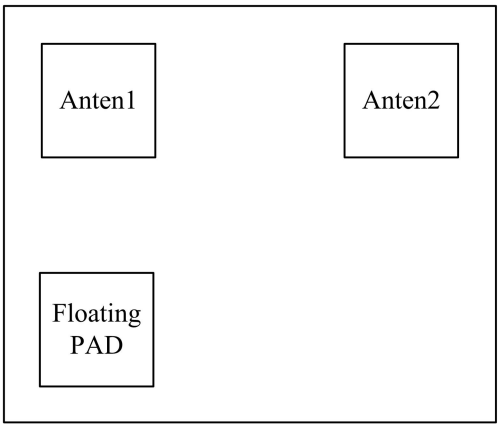

图 1 产品管脚图

管脚	名称	类型	描述
1	Anten1	I/O	外接天线端 1
2	Anten2	I/O	外接天线端 2
3	Floating PAD	NC	Floating PAD

2 技术参数

2.1 功能模块描述

FMXC50是一款非接触式IC卡芯片产品，其内部结构主要由模拟部分、数字部分和EEPROM存贮单元电路三个部分组成，如下图所示：

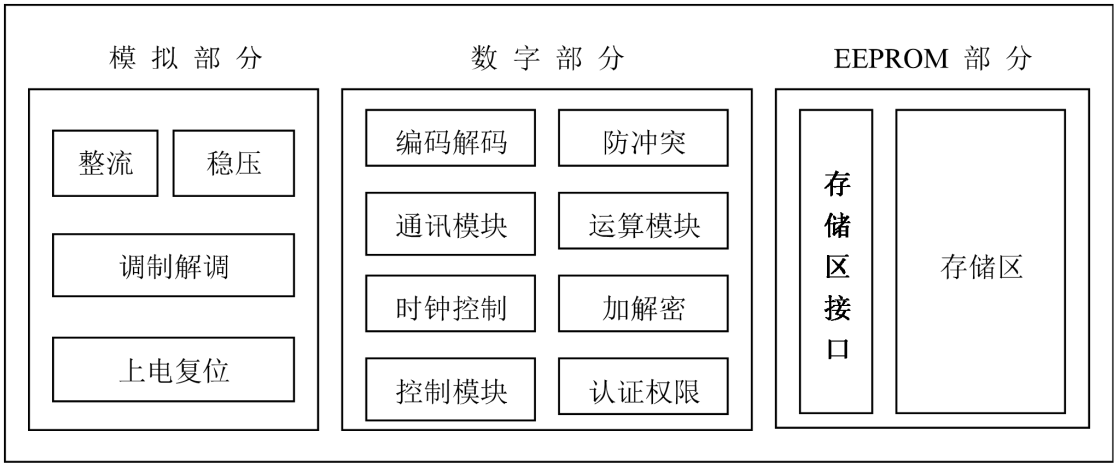

图 2 FMXC50 非接触式 IC 卡内部结构

模拟部分主要有四个功能，一是给IC卡内部各部分电路提供工作时所需要的能量,通过电源产生电路完成；二是从载波中提取电路正常工作时需要的时钟，由时钟恢复电路完成；三是对进出IC卡的数据进行调制解调，由数据调制解调电路完成；四是上电复位，由复位电路完成。

数字部分由主控制模块、通讯模块、信息安全模块等部分组成。各模块在主控制模块的控制下，对读卡器的指令进行响应。

EEPROM存贮单元电路用来存储关键数据，它通过EEPROM接口电路与数字部分进行通讯。为数字部分提供必要的数据或数据读写指令执行的结果。

2.2 交易流程

当FMXC50非接触式IC卡接收到读卡器的指令后，经过指令译码，通过控制逻辑进行数据处理，并返回相应的处理结果。

当FMXC50非接触式IC卡位于读卡器的有效工作范围之外时，卡上芯片处于无电状态，不能进行任何操作。

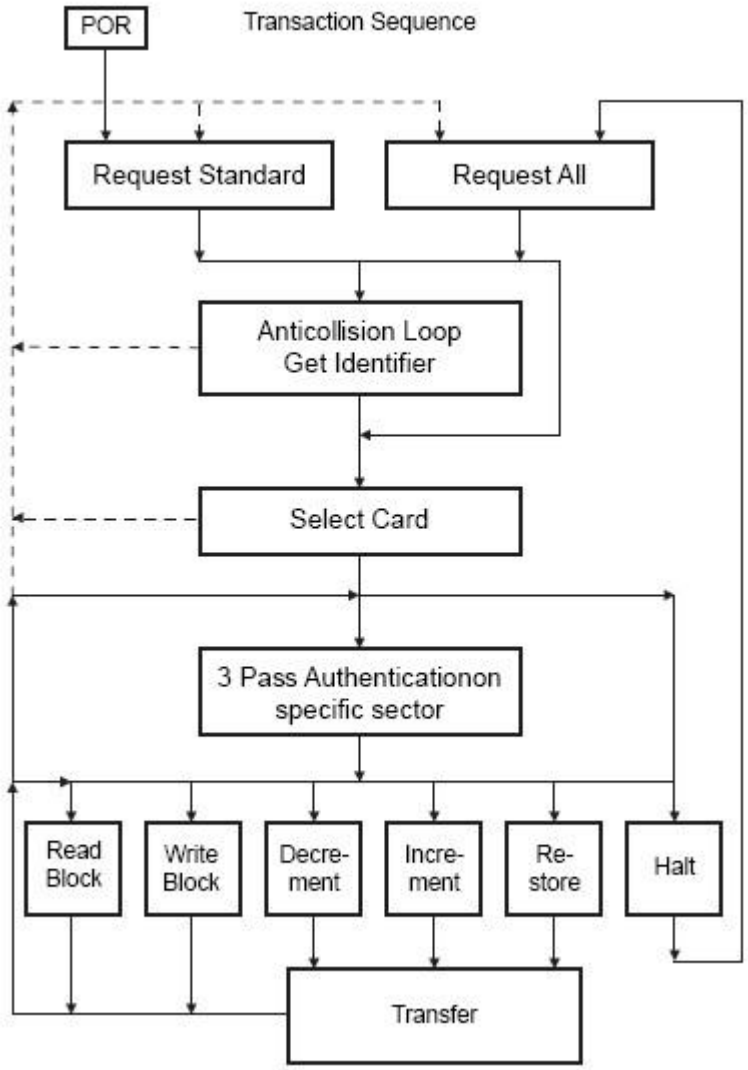

图 3　　交易流程

3 FMXC50 的状态

下面的状态图描述了FMXC50所有可能出现的状态，其中的缩写的意义如下：

AC：ANTICOLLISON Command (matched UID)

nAC: ANTICOLLISON Command (not matched UID)

SELECT: SELECT Command (matched UID)

nSELECT: SELECT Command (not matched UID)

Error: transmission error detect

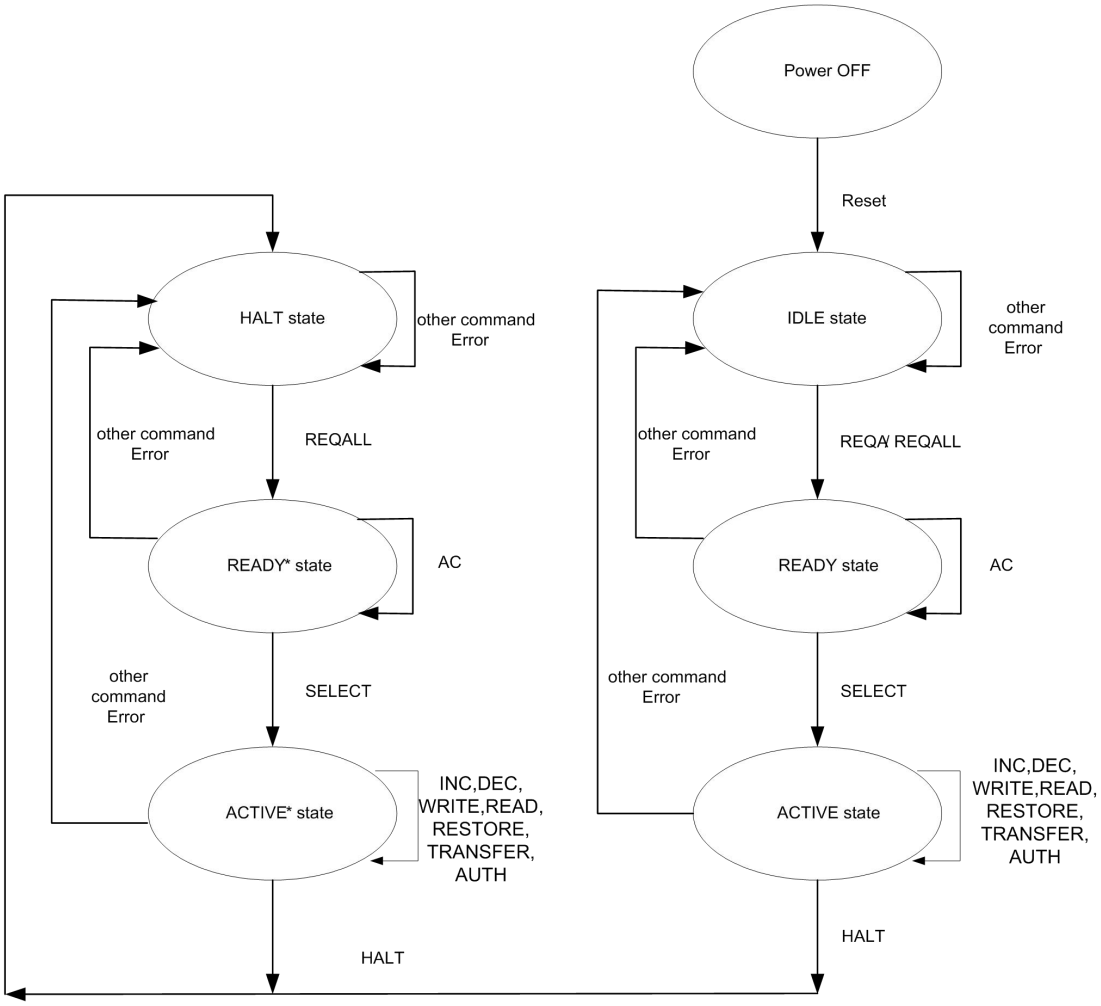

图 4　　FMXC50 状态图

3.1 Power OFF 状态

- **状态描述**

在Power OFF状态中，由于缺少载波能量，FMXC50不能被激活。

- **状态的退出和转换**

如果FMXC50进入有效工作场强，应该在5ms内进入IDLE状态。

3.2 IDLE 状态

- **状态描述**

在IDLE状态，FMXC50被加电，并且能够识别REQA和REQALL指令。

- **状态的退出和转换**

当FMXC50接收到有效的REQA和REQALL指令并以ATQA响应后，FMXC50进入READY状态。当FMXC50接收到其它指令或错误时，保持在IDLE状态。

3.3 READY 状态

- **状态描述**

在READY状态，可以进行抗冲突操作，得到FMXC50所有的UID，并能够识别SELECT指令。

- **状态的退出和转换**

当使用 FMXC50 完整的 UID 进行选卡操作且 FMXC50 被选中后，FMXC50 进入 ACTIVE 状态。当 FMXC50 接收到其它指令或错误时，回到 IDLE 状态。

3.4 ACTIVE 状态

- **状态描述**

在ACTIVE状态，FMXC50能够处理它认为合法的应用信息。

- **状态的退出和转换**

当FMXC50接收到有效的HALT指令后，FMXC50进入HALT状态。当FMXC50接收到其它指令或错误时，

回到IDLE状态。

3.5 HALT 状态

- **状态描述**

在HALT状态，FMXC50只能响应REQALL指令。

- **状态的退出和转换**

当FMXC50接收到有效的REQALL指令并以ATQA响应后，FMXC50进入READY状态。

3.6 READY*状态

- **状态描述**

在 READY*状态，可以进行抗冲突操作，得到 FMXC50 所有的 UID，并能够识别 SELECT 指令。

- **状态的退出和转换**

当使用完整的 UID 进行选卡操作且 FMXC50 被选中后，FMXC50 进入 ACTIVE*状态。
当 FMXC50 接收到其它指令或错误时，回到 HALT 状态。

3.7 ACTIVE*状态

- **状态描述**

在 ACTIVE*状态，FMXC50 能够处理合法的应用指令：包括认证、读写、加减、存储、传输、暂停。

- **状态的退出和转换**

当接收到有效的 HALT 指令后，FMXC50 进入 HALT 状态。
当 FMXC50 接收到其它指令或错误时，回到 HALT 状态。

4 操作指令

4.1 询卡指令：REQA (26H)和 REQALL (52H)

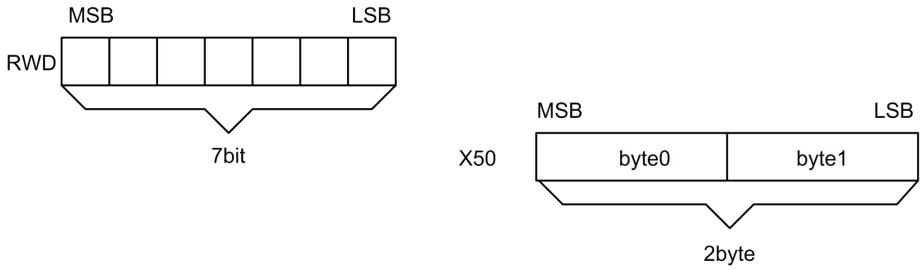

图 5 询卡指令

当FMXC50处于IDLE状态时，能够响应REQA或REQALL指令，并发送回2byte的固定数据；当FMXC50处于HALT状态时，只能够响应REQALL指令，并发送回2byte的固定数据。

4.2 抗冲突指令 AC（93H 20H）和选卡指令 SELECT（93H 70H + UID）

当FMXC50处于READY状态时，如果接收到抗冲突指令或选卡指令，FMXC50会以相应的数据响应。

这些命令在抗冲突环期间使用。AC和SELECT命令由下列内容组成：

- 选择代码SEL（1个字节）
- 有效位的数目NVB（1个字节）
- 根据NVB的值，对应UID （卡号）的0到40个数据。

NVB规定了读写器所发送的数据的有效位的数目。

注：只要NVB没有规定40个有效位，若卡保持在寻卡响应后的READY状态中，该命令就被称为抗冲突（Ant）命令。

如果NVB规定了UID的40个数据位（NVB=‘70’），则应添加CRC码。该命令称为选卡（Sel）命令。

如果卡已发送了完整的UID，则它从READY状态转换到ACTIVE（卡被选上）状态并在其SAK响应中指出

UID完整。

在流程中，一旦收到SELECT指令，并且SELECT指令中所包含的UID (即卡号) 和自己不一致，必须退回到IDLE状态。

关于抗冲突的具体规范请参见ISO/IEC14444-3。

4.3 认证指令：Authentication (60H/61H)

选定要处理的卡之后，读卡器与卡之间要进行三次相互认证，只有都通过了认证，读卡器才能对卡进行操作。

三次认证的一般过程如下图所示：

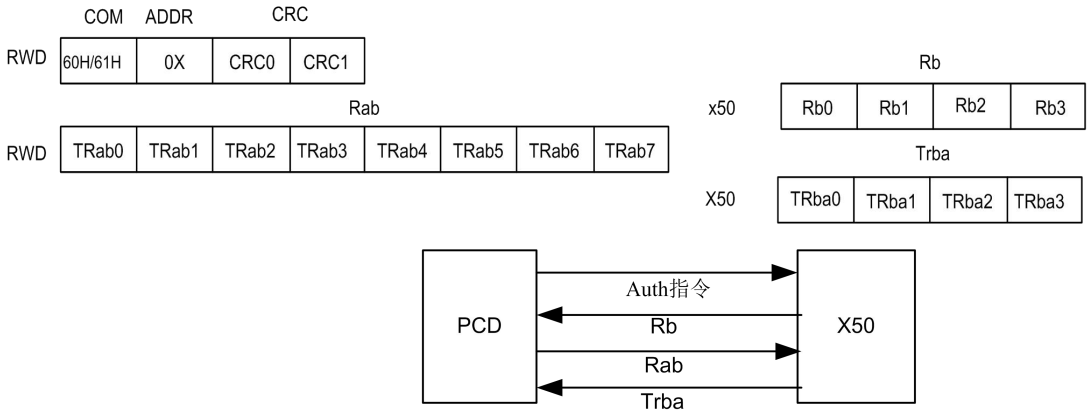

图 6　　三次认证过程图

FMXC50与读卡器之间的认证遵循ISO/IEC9798-2中规定的三重认证。

1） 读卡机首先向FMXC50发送认证指令。

2） FMXC50接收到认证指令时，首先返回四个字节的随机数Rb。

3） 读卡机接收到Rb后，用加密算法和相应的密钥加密产生8个字节的Rab，发送给FMXC50。

4） FMXC50接收到Rab后，用加密算法与相应的密钥解密，只要发现随机数Rb与解密结果一致，FMXC50就认为认证通过，同时发送四个字节的经相应的密钥加密的随机数Trba给读卡机。

注：当认证指令为60时，用KEYA进行认证；

当认证指令为61H时，用KEYB进行认证。

4.4 读指令: READ (30H)

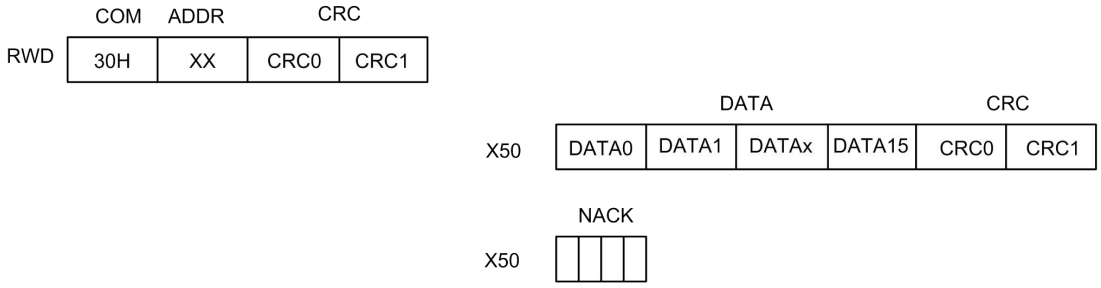

图 7 读指令

当卡处于ACTIVE状态时，如果接收到读指令，卡将根据接收到指令中的地址对此操作进行权限校验，如果权限校验通过，则从EEPROM中读出1BLOCK（16Byte）的数据，经奇偶校验和CRC校验后发送给读卡器。否则，卡将返回错误确认NACK（4bit）。

4.5 写指令：WRITE（A0H）

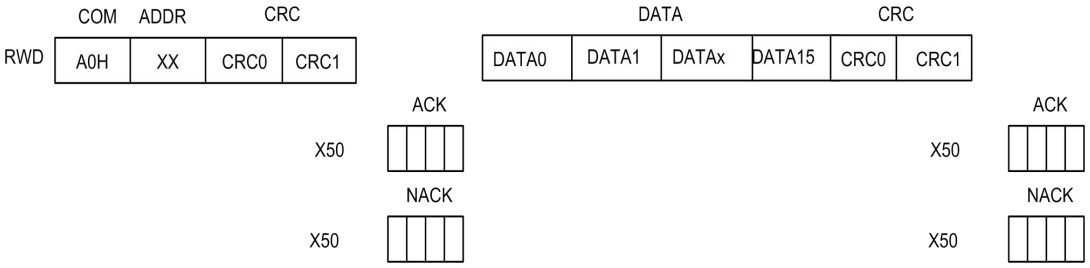

图 8 写指令

当 FMXC50 处于 ACTIVE 或 ACTIVE*状态时，正确接收到此指令的第一帧数据后，FMXC50 将对相应的地址进行权限认证，并返回确认信息。然后当 FMXC50 正确接收此指令的第二帧数据（16Byte）后，把收到的数据 byte0~byte15 写到对应地址 block 的 byte0~byte15 中，并返回确认信息。

4.6 HALT (50H)指令

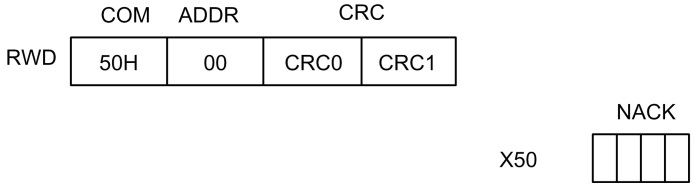

图 9 HALT 指令

当FMXC50处在ACTIVE状态，接收到HALT指令后，进入HALT状态，在HALT状态，只有REQALL（52H）

指令可以将其唤醒；当FMXC50处于其它状态，接收到HALT指令后，FMXC50会返回NACK（4bit）。

4.7 加 INCREMENT (C1H)、减 DECREMENT (C0H)、复制 RESTORE (C2H)

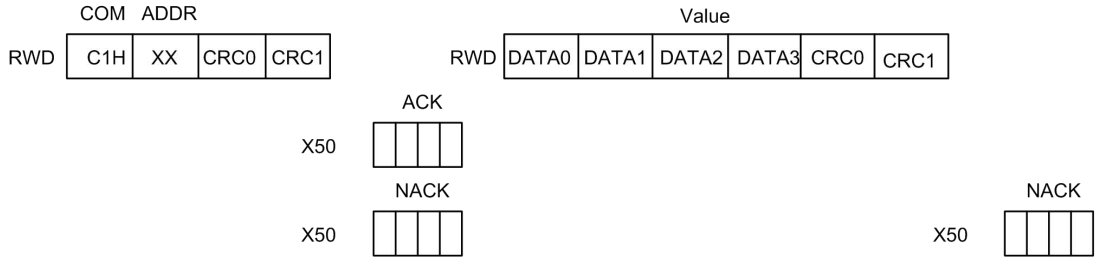

图 10 加/减/复制指令

在进行加、减、复制操作时，如果执行加、减、复制指令正确，FMXC50会返回四个bit的ACK，否则，FMXC50会返回四个bit的NACK；当返回ACK时，FMXC50等待接收四个byte的操作数和它的CRC校验码，同时卡内的运算器开始工作，将操作数与EEPROM内的数据相加减，如果加减结果溢出或者通讯错误，FMXC50会返回四个bit的NACK，否则FMXC50无响应，并等待transfer指令。

FMXC50在下列条件下能够执行加、减、复制操作：

- FMXC50内的block已经通过写操作初始化为“数值块”格式。
- 在通过KEYA和KEYB认证后，FMXC50的权限控制位允许对目标block进行加、减操作。
- 不允许进行加零、减零操作。

与加指令有相同指令格式的指令还有减指令（DECREMENT(C0H)）、复制指令(RESTORE(B0H)) ：

加：加一特定的值到FMXC50的数值块；

减：从FMXC50的数值块减去一特定的值；

复制：执行减零操作；

在无其他指令操作前，计算的结果存储在FMXC50内部的buffer内。Transfer指令不会改变内部的数据。Transfer操作能够遵守数值block的存储格式，将buffer内的数据存储到目标地址的block中。

操作数的长度为4个byte，以原码表示，其最高位在进行加减操作时默认为0。

在卡接收到加、减、复制指令后，如果在加减溢出时，也就是计算结果超出EEPROM所能存储的最高数

值或最低数值时，FMXC50会响应NACK。

4.8 传输指令：TRANSFER (B0H)

每一次执行加、减、复制指令时，其后必须紧跟“transfer”指令，以便将存储在FMXC50内部buffer内的计算结果存储到FMXC50内的EEPROM中。“transfer”指令的目标地址可以与加、减、复制的目标地址相同，也可以是与加、减、复制的目标地址在同一扇区的其它BLOCK的地址。

除“transfer”指令外，其它指令都会影响FMXC50内部buffer所存的数据。

Transfer指令的目标地址必须有transfer的权限，才能将计算结果存储到卡内的EEPROM中。“transfer”指令自动的将加减结果以正确的数据格式存储到目标地址的BLOCK中。

注：transfer指令必须是加、减、复制指令的下一条指令，否则transfer指令无效。

5 FMXC50 的确认编码（acknowledge-code）

FMXC50的确认编码有NACK和ACK：其中NACK为出错确认；ACK为正确确认。编码方式如下：

ACK：1010 B

NACK:

- 0000 B：无权限操作目标地址BLOCK；
- 0001 B：CRC或parity错误；
- 0100 B：对于“transfer”表示溢出；对于其它指令表示无权限操作目标地址BLOCK；
- 0101 B：CRC或parity错误。

6 存储区结构

6.1 存储区结构

FMXC50非接触式IC卡中的EEPROM存储区容量为8 k Bits，分成16个扇区。每个扇区分成4个块，每个块由16个字节组成，每个字节有8位。块是存储区的最小访问单位。

每个扇区有三个块用于存储数据，另一个被称为区尾（Sector Trailer）的块存放该扇区的访问控制条件，从而实现对存储区域的安全保护。存储数据的块分为普通数据块和作为电子钱包的金额块。

		Byte number within a block																	
Sector	block	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	description	block addr
15	3	KEY A						ACCESS BITS				KEY B						SETCTOR TRAILER 15	3FH
	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	3EH
	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	3DH
	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	3CH
14	3	KEY A						ACCESS BITS				KEY B						SETCTOR TRAILER 14	3BH
	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	3AH
	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	39H
	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	38H
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
1	3	KEY A						ACCESS BITS				KEY B						SETCTOR TRAILER 1	07H
	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	06H
	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	05H
	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	04H
0	3	KEY A						ACCESS BITS				KEY B						SETCTOR TRAILER 0	03H
	2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	02H
	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	DATA	01H
	0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Manufacturer DATA	00H

图 11 EEPROM 存储区的结构图

6.2 制造商块

0扇区（sector 0）0块（block 0）是特殊的数据块，用于存放制造商代码，此块只读。

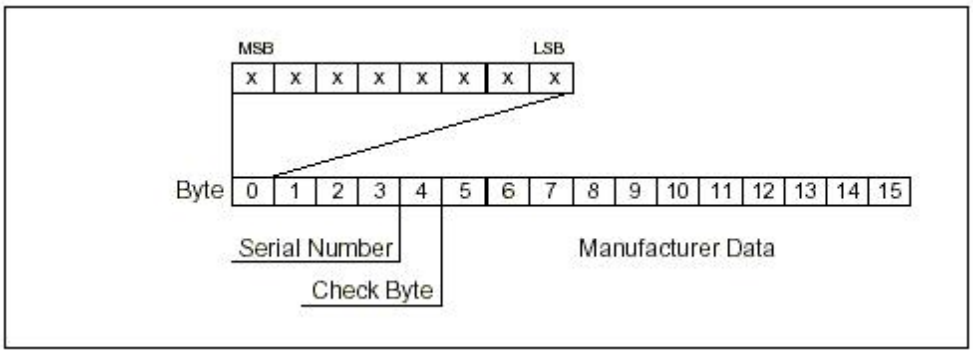

图 12 制造商块存储区的结构图

6.3 普通数据块

普通数据块可通过数据块备份进行数据完整性的管理。一般每块16个字节中有两个用来存放校验码和备份块的地址。

6.4 数值块

数值块（金额块）则采用冗余存储方法以保证其数据的正确性。数值块表示为一个带符号的16进制整数，包括符号位在内，一共为四个字节。每个数据共存储三次，用于错误检测和纠正。剩余的四个字节可以存储一个字节的地址信息（一般为数据备份区的地址），该数据存储四次，以便进行错误的检验。

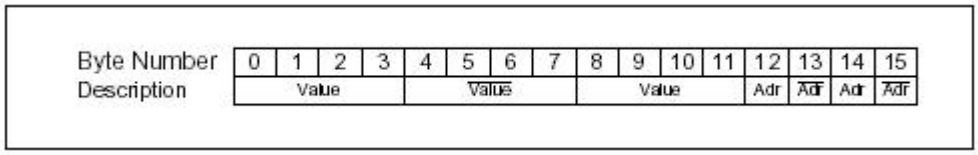

图 13 数值块的数据存储结构

6.5 访问控制块

在每个扇区最后一个块即区尾中存放有密钥A、访问条件、密钥B等内容，它们用于控制对该扇区的操作。其中第0－5字节为48位的密钥A；第6－8字节为访问控制条件；第9字节为备用区可用于存放特殊的应用数据，如存放数据备份区的地址；剩下的6个字节存放密钥B，如用户不需要密钥B，则可用于存放一般的应用数据。

由于区尾中访问条件的数据很重要，因此使用了冗余存储的方法。

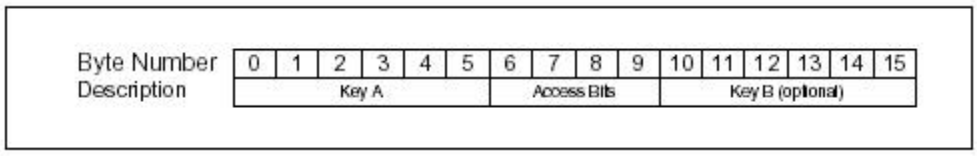

图 14　　每扇区 BLOCK3 的数据存储结构

7　扇区的访问权限

7.1　扇区的访问权限

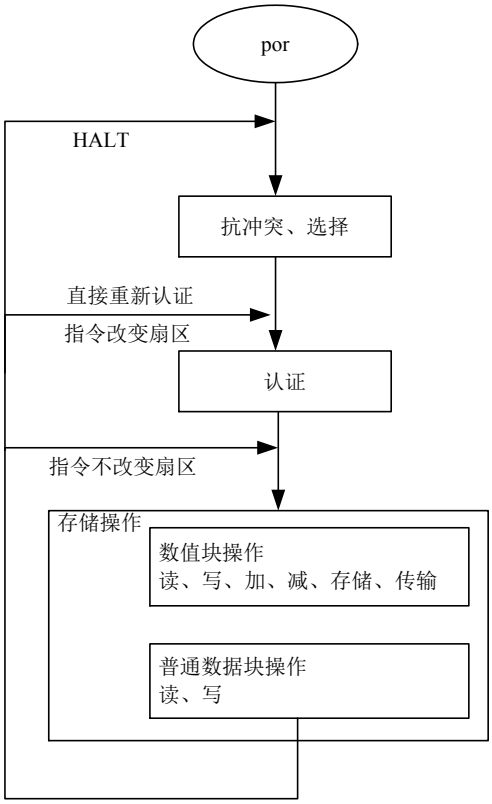

图 15　　扇区访问流程

表 1　扇区中块的访问权限

存储区操作		
操作	**描述**	**视为有效操作的扇区块**
读	读一个存储块	读/写，数值块和区尾
写	写一个存储块	读/写，数值块和区尾
加	与一个块的值相加，并将结果存入内部数据寄存器	数值块
减	减去一个块的值，并将结果存入内部数据寄存器	数值块
传输	将内部数据寄存器的值写入块	数值块
存储	读一个块的值，并存入内部数据寄存器	数值块

7.2 访问条件

对整个扇区的数据块和区尾的操作都由访问条件（3 bits）决定，访问条件决定了对扇区中块访问的四种情况：

- 需要验证密钥 A 或密钥 B
- 需要验证密钥 B
- 需要验证密钥 A
- 不允许

访问条件的3 bits以原码和反码的形式存储在相关扇区的区尾中。

表 2　数据存储条件数据存储方式

访问条件的编码	有效操作	扇区块	描述
$C1_3$　$C2_3$　$C3_3$	读/写	3	区尾
$C1_2$　$C2_2$　$C3_2$	读/写/加/减/传输/存储	2	数据块
$C1_1$　$C2_1$　$C3_1$	读/写/加/减/传输/存储	1	数据块
$C1_0$　$C2_0$　$C3_0$	读/写/加/减/传输/存储	0	数据块

7.3 数据块访问条件编码

数据块所需访问条件的定义如下表所示，其中A/B表示需要通过密钥A或B的验证才可以进行操作，Never表示不允许进行相应的操作。从表中不难看出，所有的操作都可能访问数据块，只是需要满足不同的访问条件。但是，在数据块被锁定后，即访问条件被置成111的情况下，则任何操作都不能访问数据块。

表 3 数据块访问条件

访问条件的编码(原码值)			数据块的访问条件			
C1	C2	C3	读	写	加	减/传输/存储
0	0	0	A/B	A/B	A/B	A/B
0	1	0	A/B	Never	Never	Never
1	0	0	A/B	B	Never	Never
1	1	0	A/B	B	B	A/B
0	0	1	A/B	Never	Never	A/B
0	1	1	B	B	Never	Never
1	0	1	B	Never	Never	Never
1	1	1	Never	Never	Never	Never

7.4 控制块访问条件编码

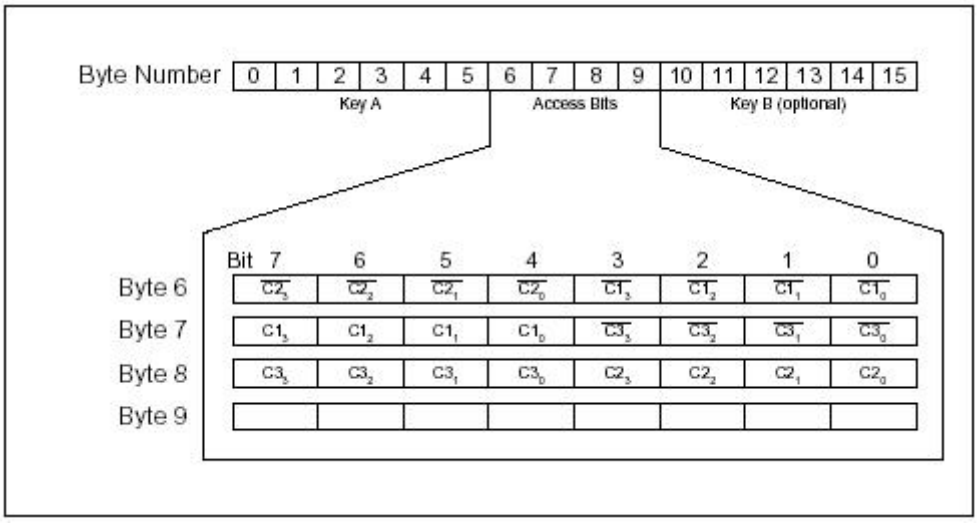

图 16 区尾块存储区的结构图

控制块位于扇区的区尾，需要的访问条件定义如下（FMXC50包含两种访问条件编码的产品FMXC5001和FMXC5002）：

其中FMXC5001的区尾块访问条件遵照表4，FMXC5002的区尾块访问条件遵照表5。

表 4　区尾块访问条件编码（适用于 FMXC5001）

访问条件的编码(原码值)			密钥 A（0~5 字节）		访问条件区（6~9 字节）		密钥 B（10~15 字节）	
C1	C2	C3	读	写	读	写	读	写
0	0	0	Never	A/B	A/B	Never	A/B	A/B
0	1	0	Never	Never	A/B	Never	A/B	Never
1	0	0	Never	B	A/B	Never	Never	B
1	1	0	Never	Never	A/B	Never	Never	Never
0	0	1	Never	A/B	A/B	A/B	A/B	A/B
0	1	1	Never	B	A/B	B	Never	B
1	0	1	Never	Never	A/B	B	Never	Never
1	1	1	Never	Never	A/B	Never	Never	Never

表 5　区尾块访问条件编码（适用于 FMXC5002）

访问条件的编码			**密钥 A（0～5 字节）**		**访问条件区（6～9 字节）**		**密钥 B（10～15 字节）**	
C1	C2	C3	读	写	读	写	读	写
0	0	0	Never	A	A	Never	A	A
0	1	0	Never	Never	A	Never	A	Never
1	0	0	Never	B	A/B	Never	Never	B
1	1	0	Never	Never	A/B	Never	Never	Never
0	0	1	Never	A	A	A	A	A
0	1	1	Never	B	A/B	B	Never	B
1	0	1	Never	Never	A/B	B	Never	Never
1	1	1	Never	Never	A/B	Never	Never	Never

注：如果KEYB能被读出（如表5灰色区域所示），则用KEYB认证后，不能再对存储区进行后续的操作。