How to Use Catalyst Secure Access Serial E²PROMs

Applications Staff

INTRODUCTION

This application note is intended to be a tutorial on the use of CAT35C704A/35C804A Secure Access Serial E²PROMs. Device operation and typical applications for the device are shown as well as examples for each of the instructions available.

DEVICE OPERATION

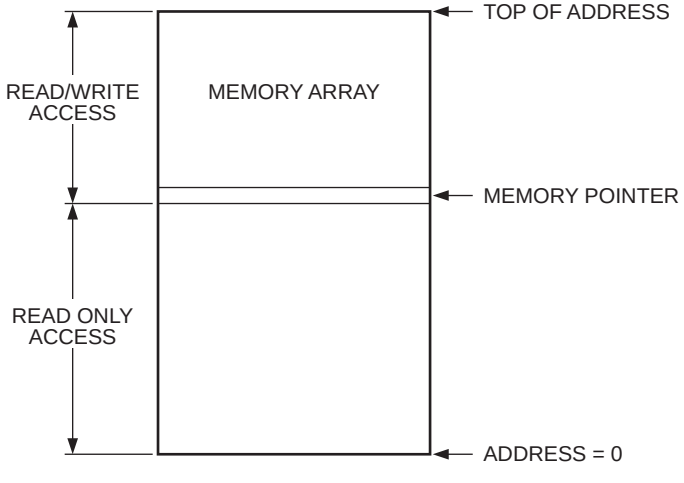
The CAT35C704A/35C804A is a 4K bit Secure Access Serial E²PROM that can be used in applications that require nonvolatile memory storage and a need to protect the contents of that memory from unauthorized access. Two basic modes of operation are available, protected and unprotected. In the unprotected mode, with the memory pointer set to "0", the device operates like a standard E²PROM, allowing full read/write access to the entire array.

Using the memory pointer the user can determine how much memory needs protection. With the WMPR command, a pointer value can be set to split the memory array into two blocks. Addresses above the pointer value offer full Read/Write access; addresses below and including the pointer are Read only (see Figure 1).

In the protected mode, up to 8 bytes of password security are available. Once the password has been set and a disable access (DISAC) command (or power down) has been executed, the device becomes inaccessible with only the portion of the array not protected by memory pointer readable (see Figure 2). Upon power up, the correct password must be sent to the device before any writing or moving of the memory pointer can be done. This scheme lends itself to applications where users are allowed to view only those portions of memory that is intended for them to see. For example, an application where data is uncovered in the array (by moving the memory pointer) to make available to the user certain features/options that they require, as in the cable TV industry (see Figure 3).

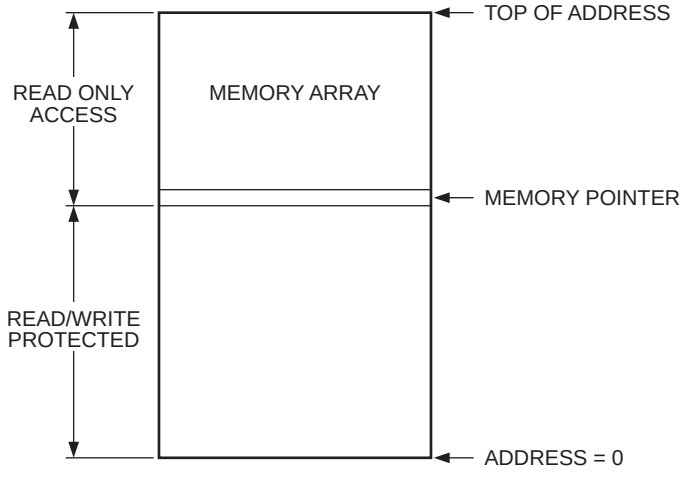
Among the 19 instructions available with the CAT35C704A/35C804A is a Read Status Register (RSR) instruction, which lets a system interrogate the device and determine its working status. The 8 bit status register displays information regarding parity errors, instruction errors and RDY/$\overline{\text{BUSY}}$ status. An organization instruction (ORG) is also available for organizing the memory into either 512x8 or 256x16 configurations depending on the application.

Figure 1. Access Control Using No Access Code



5195 FHD F02

Figure 2. Access Control Using Access Code



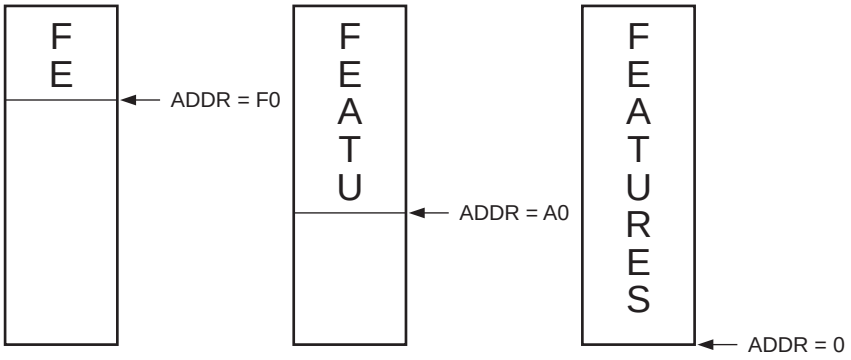
5195 FHD F03

In addition, to allow for reading multiple words from the memory and minimize the overhead of repeated Read instructions, a Read Sequential (RSEQ) instruction allows you to specify a starting location and then continuously shift out data to the end of the array.

The security code is entered/modified by sending the Modify Access Code (MACC) instruction followed by the length of the access code (1 to 8 bytes), the old access code (if needed), and then the new access code twice (for verification). Once power has been removed (or the DISAC instruction sent), the Enable Access (ENAC) instruction, followed by the correct access code, must be sent to the device or the memory array's protected portion cannot be accessed, and the memory contents above the pointer remain Read only.

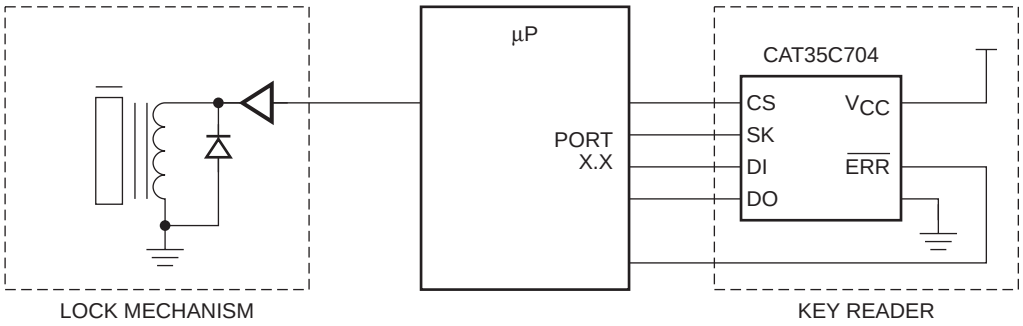
A simple interface is shown in Figure 4 where the CAT35C704 is used in an electronic key application. The device interfaces directly to a microprocessor and is used as the security portion of a door lock mechanism. The lock is only activated when the key's (hotel key, car key, etc.) access code matches the one stored in the CAT35C704.

Figure 3. Using CAT35C704A/35C804A for Protected Features



5195 FHD F04

Figure 4. CAT35C704 in an Electronic Key



5195 FHD F05

INSTRUCTION SET

The following section describes the 19 instructions available for the device and examples of each.

SECURITY OR WRITE PROTECT INSTRUCTIONS

1. **MACC**–Modify Access Code

This instruction allows the user to issue a new password to the device or modify an existing one. The password is issued in the following manner:

1101	[Length of new pswrd]	[old pswrd]	[new pswrd]	[new pswrd]

For example, to issue the device a password for the first time, send the following:

1101	3	AA 55 D2	AA 55 D2
Instruction Code	*Length*	*3-Byte Pswrd*	*Repeat 3-Byte Pswrd*

The device now has a 3 byte password of AA 55 D2. To change this password to a 5 byte password, send the following:

1101	5	AA 55 D2	01 02 03 04 05	01 02 03 04 05
Instruction Code	*Length of New Pswrd*	*Old Pswrd*	*New Pswrd*	*Repeat New Pswrd*

The device now has a 5 byte password equal to 01 02 03 04 05. This password can be modified in the same manner to any length password you choose, up to 8 bytes.

Finally, to modify the password back to a 0 length (no password), send the device the following instruction:

1101	0	01 02 03 04 05
Instruction Code	*Length of New Pswrd*	*Old Pswrd*

The device is now in the unprotected mode.

2. **ENAC/DISAC**—Enable/Disable Access

These two instructions permit the user to turn on or off the password protection to the device. To disable any access to the device, send the following instruction:

1000 | 1000
Instruction Code

The device will give no indication that it has been disabled other than you now cannot Read or Write to the array.

To enable the device operation, send the following instruction:

1100 | 0101	[Access Code]

For example, to enable access to a device that has an 8 byte password stored in the access code register, send the following:

1100 | 0101	01 02 03 04 AA BB CC DD
Instruction Code	*8-Byte Pswrd*

Again, the device gives no indication that you have entered the correct password, however you now have full Read/Write capabilities.

3. **WMPR**—Write Memory Pointer Register

The Write Memory Pointer Register instruction allows you to modify the contents of the memory pointer register. The value of the register determines what portion of the memory array is protected from byte-writes during unprotected operation and what portion you are allowed to read during protected operation.

For example, if there is no password protection and the memory pointer is set to 00AA, then no byte-writes from address 0000 to address 00AA are allowed (unless the OVMPR instruction has been entered previously). In the protected mode, with the memory pointer set to the same value (00AA), a Read Sequential (RSEQ) instruction from address 0000 will not allow the user to read any of the array. The RSEQ instruction must begin at 00AA and will then allow read access from 00AA to the end of the array. Note: The memory pointer contents will not block Erase All or Write All operations.

To change the contents of the register, send the following:

1100 | 0100	[A15–A8] [A7–A0] *x8* [A7–A0] *x16*

This instruction is operational only after an ENAC instruction (if a password has been set) and an EWEN (see EWEN section) instruction have been sent to the device. Once this is done, you can modify the register contents to the desired value. For example, to change the contents from 0000 to 0123, send the following:

1100 | 0100	0123
Instruction Code	*New Memory Pointer Value*

Stock No. 21041-01 2/98

The memory pointer register now has a value of 0123 (x8).

4. **RMPR—**Read Memory Pointer Register

The Write Memory Pointer Register instruction allows you to read the location in memory where the memory pointer resides. This tells you which portions of the memory are divided between read only and full read/write access. To read the value of the register, send the following:

1100	1010
Instruction Code	

The device will then return the hex value of the memory pointer location.

5. OVMPR—Override Memory Pointer Register

This instruction allows the user to write data to a protected area of memory on a one time basis, without having to uncover that area with the memory pointer. For example, to write data to an area protected by the memory pointer the OVMPR instruction would be issued, followed immediately by a write instruction. After the write has been completed, the area of memory is again protected.

1000	0011
Instruction Code	

READ/WRITE/ERASE INSTRUCTIONS

1. **READ—**Read Memory

This instruction outputs the data from memory at the specified location.

1100	0101	[A15–A8] [A7–A0] *x8* [A7–A0] *x16*
Instruction Code		*Address*

For example, to Read the contents of address 1AH, send the following:

1100	1001	00011010

The device then outputs data located at this address on the DO pin.

2. **WRITE—**Write Memory

The Write instruction writes an 8 or 16 bit data word into a specified address of memory. Once the instruction, address, and data have been entered, the self-timed program/erase cycle will start. The addressed memory location is erased before data is written. For example, to write the data 5A2D Hex to address C8, send the following:

1100	0001	11001000	0101101000101101
Instruction Code		*Address (x16)*	*Data (x16)*

After the specified Program/Erase pulse width, the data 5A2D is written to address C8 Hex.

3. **ERASE—**Clear Memory

The Erase instruction clears the specified memory location by setting all cells to a logic “1”. Once the instruction and address have been entered, the self-timed erase cycle will start. For example, to erase the data located at address 1234 Hex, send the following:

1100	0000	0001001000110100
Instruction Code		*Address (x8)*

After the specified Erase pulse width, the contents of address 1234 Hex will be FF Hex.

4. **ERAL—**Erase All

The Erase All instruction clears the data from all locations in the memory. To erase the entire device, send the following:

1000	1001	1000	1001
Instruction Code		*Instruction Code*	

The code is required to be sent twice (to protect against inadvertent chip clear) and, once sent, clears all locations to the FF Hex state.

5. **WRAL—**Write All

The Write All instruction is used to write the same data byte to all locations in the memory. For example, to write the data AA Hex to all locations, send the following:

1000	1001	1100	0011	10101010
Instruction Code		*Instruction Code*		*Data (x8)*

After the specified Program/Erase pulse width, all locations in the device will now have AA Hex written to them.

6. **RSEQ—**Read Sequential

The Read Sequential instruction allows the user to sequentially clock out data starting at a specified address continuing until the end of memory or Chip Select

is brought low. For example, to read memory starting at address 4D Hex continuing to the end of the array, send the following:

1100	1011	01001101
Instruction Code		*Address (x16)*

The device will now clock out (SK pin must be clocked by user) the contents of memory starting at address 4D and continuing to the end of memory.

STATUS AND CONTROL INSTRUCTIONS

1. **EWEN—**Erase/Write Enable

This instruction is required to be entered before any program/erase instruction will be carried out. Once it is entered, it remains valid until a power down or a EWDS instruction is sent. To enable the device for writing/erasing, send the following:

1000	0001
Instruction Code	

The device is now ready to be erased or written to.

2. **EWDS—**Erase/Write Disable

This instruction disables all writing or erasing of the device. Once sent, the device must be sent an EWEN instruction before any erase/write instruction will be performed. To disable erase/write instructions, send the following:

1000	0010
Instruction Code	

The device is now protected from any erase or write instructions.

3. **ORG—**Select Memory Organization

This instruction allows the user to select a x16 or x8 memory organization. For example, to configure the device with a word length of 8 bits, send the following:

1000	0110
Instruction Code	

To configure the device with a word length of 16 bits, send the following:

1000	0111
Instruction Code	

4. **RSR—**Read Status Register

The Read Status Register instruction allows the user to determine the state of the device. To determine if the device is in an error condition, send the following:

1100	1000
Instruction Code	

The device then responds with an 8 bit status word that gives the following information:

10100000 - the device is operating normally
10110000 - the device has a parity error
10101000 - the device has an instruction error
10100100 - the device is in the program/erase cycle

5. **DISBSY—**Disable Busy

The Disable Busy instruction disables the RDY/$\overline{\text{BUSY}}$ status on the DO (data out) pin. To disable the RDY/$\overline{\text{BUSY}}$ function, send the following:

1000	0101
Instruction Code	

The RDY/$\overline{\text{BUSY}}$ status is now no longer available on the DO pin.

6. **ENBSY—**Enable Busy

The Enable Busy instruction enables the RDY/$\overline{\text{BUSY}}$ status on the DO pin. To enable this status, send the following:

1000	0100
Instruction Code	

The RDY/$\overline{\text{BUSY}}$ status is now enabled on the DO pin. This allows the user to tell if the device is in the program/erase cycle (DO low) or has completed it (DO high).

7. **NOP—**No Operation

The NOP instruction leaves the device in an idle mode; no operation is executed.