# Programming Specification for AT17F(A) Series FPGA Configuration Memories

## 1. The FPGA Configurator

The AT17Fxx(A) Configurator is a serial flash memory device generally used to program FPGA type devices with their functional bit stream. This document describes the procedures for erasing, writing, reading and performing device identification while in program mode (i.e., when SER_EN- is driven low).

There are many ways to program an AT17F(A) Series Configurator:

- Using a third-party programmer
- Using Atmel's Configurator Programming Kit ATDH2200E
- Using Atmel's Configurator Programming Cable ATDH2225
- Using a custom programming solution

For details of other programming options, please refer to "Introducing Atmel Configurators" application note, available on the Atmel web site (www.atmel.com).

## 2. Serial Bus Overview

The serial bus is a Two-Wire Interface (TWI) bus. One wire (CLOCK) functions as a clock and is provided by the programmer, the second wire (DATA) is a bi-directional signal and is used to provide data and control information.

Information is transmitted on the serial bus in messages. Each MESSAGE is preceded by a Start Condition and is ended with a Stop Condition. The message consists of an integer number of bytes, each byte consisting of 8 bits of data, followed by a 9th Acknowledge Bit. This Acknowledge Bit is provided by the recipient of the transmitted byte. This is possible because devices may only drive the DATA line Low. The system must provide a pull-up resistor (4.7 kΩ) for the DATA line.

The MESSAGE FORMAT for read and write instructions consists of the bytes shown in "Bit Format" on page 2

While writing, the programmer is responsible for issuing the instruction and data. While reading, the programmer issues the instruction and acknowledges the data from the Configurator as necessary.

Again, the Acknowledge Bit is asserted on the DATA line by the receiving device on a byte-by-byte basis.

AT17F(A) Series FPGA Configuration Flash Memory

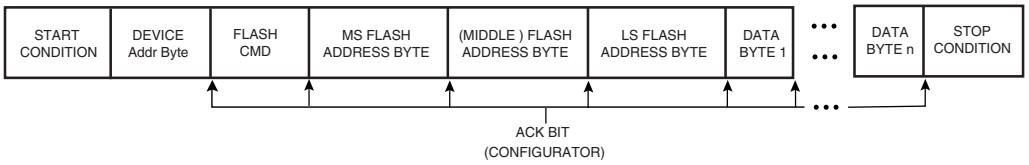
Application Note

3018E–CNFG–1/10

## 3. Bit Format

Data on the DATA pin may change only during the CLOCK Low time; whereas Start and Stop Conditions are identified as transitions during the CLOCK High time.

### Write Instruction Message Format




## 4. Start and Stop Conditions

The Start Condition is indicated by a high-to-low transition of the DATA line when the CLOCK line is High. Similarly, the Stop Condition is generated by a low-to-high transition of the DATA line when the CLOCK line is High, see Figure 7-1.

The Start Condition will return the device to the state where it is waiting for a Device Address (its normal quiescent mode).

Since the write cycle typically completes in less than $t_{WR}$ seconds, we recommend the use of “polling” as described in later sections. Input levels to all other pins should be held constant until the write cycle has been completed.

## 5. Acknowledge Bit

The Acknowledge (ACK) Bit shown in Figure 1 is provided by the Configurator receiving the byte. The receiving Configurator can accept the byte by asserting a Low value on the DATA line, or it can refuse the byte by asserting (allowing the signal to be externally pulled up to) a High value on the DATA line. All bytes from accepted messages must be terminated by either an Acknowledge Bit or a Stop Condition. Following an ACK Bit, when the DATA line is released during an exchange of control between the Configurator and the Programmer, the DATA line may be pulled High temporarily due to the open-collector output nature of the line. Control of the line must resume before the next rising edge of the clock.

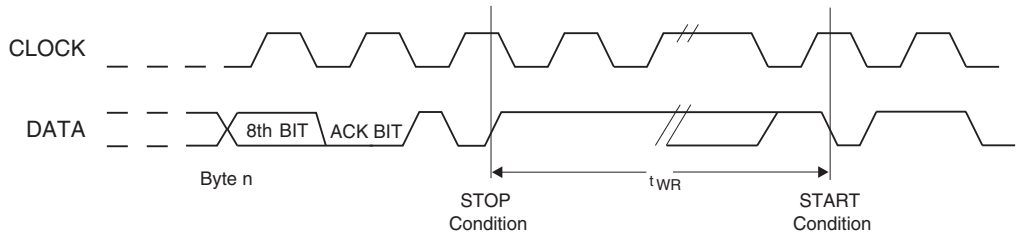
## 6. Bit Ordering Protocol

The most significant bit is the first bit of a byte transmitted on the DATA line for the Device Address Byte and the Flash Address Bytes. It is followed by the lesser significant bits until the eighth bit, the least significant bit, is transmitted.

## 7. Device Address Byte

The contents of the Device Address Byte are shown below, along with the order in which the bits are clocked into the device. The A2 bit is provided to allow multiple Configurators to share a common bus. When programming a Configurator, the A2 pin on the Configurator must be forced to a logic “0” or “1” level. It is recommended that this pin be connected to 0V (GND) using a 4.7 kΩ pull-down resistor – thereby matching the default setting of Atmel’s AT17 Configurator Programming System (CPS). Thus, the A2 bit may be used as an Address Bit among two Configurators, or as a chip-enable mechanism for In-System Programming programming employing more than two Configurators.

**Figure 7-1.** Start and Stop Conditions



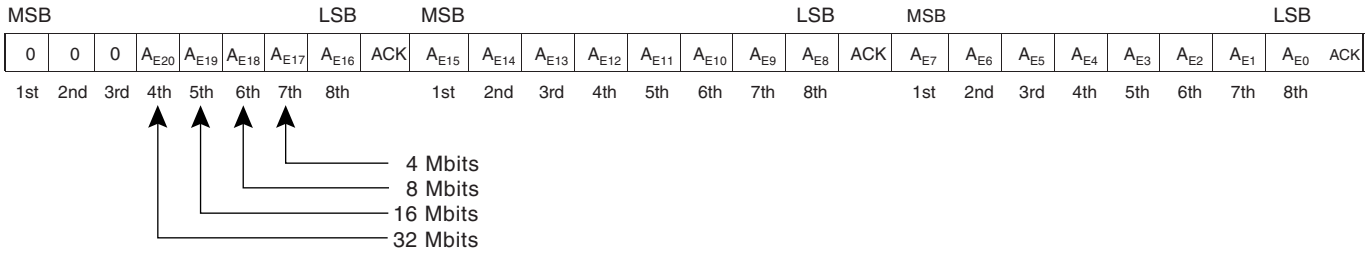

**Device Address Byte**

| MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | $A_2$ | 1 | 1 | R/$\overline{W}$ |
| 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th |

Where:R/$\overline{W}$=1 Read
= 0 Write
A2 = 1 if A2 pin of target Configurator is connected to $V_{CC}$ through a pull-up resistor
= 0 if A2 pin of target Configurator is connected to GROUND through a pull-down resistor

## 8. Flash Address

The Flash Address consists of three bytes. Each Address Byte is followed by an Acknowledge Bit (provided by the Configurator). These bytes define the normal address space of the Configurator, as described below. The order in which each byte is clocked into the Configurator is also indicated. Unused bits in an Address Byte must be set to “0”.




The AT17F device stores data as 16 bit words in its memory array, hence only 217 address locations are required to store 4 Mbits of data. The same reasoning follows for the 8M, 16M and 32M devices.

## 9. Programmer Instructions and Procedures

The following programmer instructions and procedures are supported while the Configurator is in programming mode (i.e., when SER_EN- is driven low).

## 9.1 Memory Write Procedure

Writing data into memory[1] is accomplished by providing a 3-byte Flash Address, followed by an even number of data bytes. Logic internal to the Configurator takes care sequentially writing data into the memory array. Although data is received as byte packets via the Two-Wire serial bus, the data is internally formatted and stored as words into the Flash memory array. The first byte of serial data received in a byte pair is stored as the most significant byte of the word, while the second byte received forms the least significant byte of the word.

Note: 1. Note: Prior to attempting memory writes the device must be erased. See the procedure on Erasing Memory. Memory write times (twr) are specified in the AT17F datasheet.

| Most Significant Byte[1] | | | | | | | | Least Significant Byte[1] | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th |

Note: 1. Bits are provided serially in left to right ordering from the programmer (ack' bits are not shown).

A Write Procedure consists of the following steps:

```
A Start Condition
A Device Address Byte with R/W- = 0
(An Acknowledge Bit from the Configurator)
The Flash Write Command (02h) Byte
(An Acknowledge Bit from the Configurator)
Most Significant Byte of Flash Address
(An Acknowledge Bit from the Configurator)
Middle Byte of Flash Address
(An Acknowledge Bit from the Configurator)
Least Significant Byte of Flash Address
(An Acknowledge Bit from the Configurator)
One or more Data Byte pairs(1)
(Each Byte Acknowledged by the Configurator)
A Stop Condition
```

Note: 1. The determination of a completed write cycle is accomplished by monitoring (polling) the Ack's provided by the Configurator. If a data byte is not immediately acknowledged, the programmer must resend the byte and check again for the Ack. This sequence should be repeated for a time equal to the worst case write cycle time of the AT17F device ($T_{wr}$) See the respective AT17F device datasheet for write cycle times. Note: Flowcharts with exact details of each of the program mode procedures are provided in following sections.

## 9.2 Memory Read Procedure

Read instructions are initiated similarly to write instructions, but the R/$\overline{W}$ bit in the Device Address is set to one. There are two variants of the read instruction: random read and sequential read.

### 9.2.1 Random Read

A Random Read is a Current Address Read preceded by an aborted write instruction. The write instruction is only initiated for the purpose of loading the FLASH Address Bytes. Once the Device Address Byte and the FLASH Address Bytes are clocked in and acknowledged by the Configurator, the programmer immediately initiates a Current Address Read.

A Random Read Procedure consists of the following steps:

```
A Start Condition
A Device Address Byte with R/W- = 0
(An Acknowledge Bit from the Configurator)
The Flash Read Command (01h) Byte
(An Acknowledge Bit from the Configurator)
Most Significant Byte of Flash Address
(An Acknowledge Bit from the Configurator)
Middle Byte of Flash Address
(An Acknowledge Bit from the Configurator)
Least Significant Byte of Flash Address
(An Acknowledge Bit from the Configurator)
A null byte (00h)
(An Acknowledge Bit from the Configurator)
A Stop Condition
A Start Condition
A Device Address Byte with R/W- = 1
(An Acknowledge Bit from the Configurator)
A Data Byte from the Configurator
(An Acknowledge Bit from the programmer)
```

### 9.2.2 Sequential Read

Sequential Reads follow a Random Address Read. After the programmer receives a Data Byte, it may respond with an Acknowledge Bit. As long as the Configurator receives an Acknowledge Bit, it will continue to increment the Data Byte address and serially clock out sequential Data Bytes until the memory address limit is reached. The Sequential Read instruction is terminated when the programmer does not respond with an Acknowledge Bit, but instead generates a Stop Condition following the receipt of a Data Byte.

## 9.3 Memory Erase Procedure

There are two variants of the Erase instructions: Sector Erase and Chip Erase. The Erase instructions are initiated in a manner similar to a Memory Read. The data bytes read from the Configurator provide status as to how the erase sequence is progressing (i.e. polling). When the byte(s) read equal FFh, erasure is complete. Erase[(1)] cycle times (tec) are specified in the AT17F datasheet. The Erased state of the memory bits is logic 1.

Note: 1. A memory location must be in the erased state prior to being written. A Sector or Chip Erase is required before the memory location can be written.

## 9.4 Chip Erase

In the case of a Chip Erase, the 3 memory address bytes are not required since every address in the memory array is targeted for erasure.

A Chip Erase Procedure consists of the following steps:

```
A Start Condition
A Device Address Byte with R/W- = 0
(An Acknowledge Bit from the Configurator)
The Chip Erase Command (03h) Byte
(An Acknowledge Bit from the Configurator)
A null byte (00h)
A Stop Condition
A Start Condition
A Device Address Byte with R/W- = 1
(An Acknowledge Bit from the Configurator)
A Data Byte from the Configurator
(An Acknowledge Bit from the programmer)
The programmer continues reading bytes until it receives a FFh byte, the
indication that erasure is complete.
```

### 9.4.1 Sector Erase

In the case of a sector erase a 3 byte memory address must be provided. Specifying any address within the address range of the sector is sufficient to erase the sector. Address range and sector information for the AT17F's can be found in the Sector Address Tables, see “Sector Address Tables” on page 14

A Sector Erase Procedure consists of the following steps:

```
A Start Condition
A Device Address Byte with R/W- = 0
(An Acknowledge Bit from the Configurator)
The Sector Erase Command (04h) Byte
(An Acknowledge Bit from the Configurator)
Most Significant Byte of Flash Address
(An Acknowledge Bit from the Configurator)
Middle Byte of Flash Address
(An Acknowledge Bit from the Configurator)
Least Significant Byte of Flash Address
(An Acknowledge Bit from the Configurator)
A null byte (00h)
A Stop Condition
A Start Condition
A Device Address Byte with R/W- = 1
(An Acknowledge Bit from the Configurator)
A Data Byte from the Configurator
(An Acknowledge Bit from the programmer)
The programmer continues reading bytes until it receives a FFh byte, the
indication that erasure is complete.
```

### 9.5 Device ID Check

The Device ID Instruction is initiated in a manner similar to a Memory Read. Executing the Device ID Check provides determination of the manufacturer ID, memory density and the on/off setting of the internal oscillator. (aka., support byte).

AT17F devices have a 32-bit device ID stream which must be read out as 4 data bytes.

See “Program Mode Flow Chart: Device ID Check” on page 13 for a list of Device ID's for AT17F devices.

```
A Chip Erase Procedure consists of the following steps:
A Start Condition
A Device Address Byte with R/W- = 0
(An Acknowledge Bit from the Configurator)
The Device ID Command (05h) Byte
(An Acknowledge Bit from the Configurator)
A null byte (00h)
A Stop Condition
A Start Condition
A Device Address Byte with R/W- = 1
(An Acknowledge Bit from the Configurator)
A Data Byte from the Configurator
(An Acknowledge Bit from the programmer)
The programmer continues reading bytes and providing ack's bits until it
receives the fourth data byte,
A Stop Condition
```

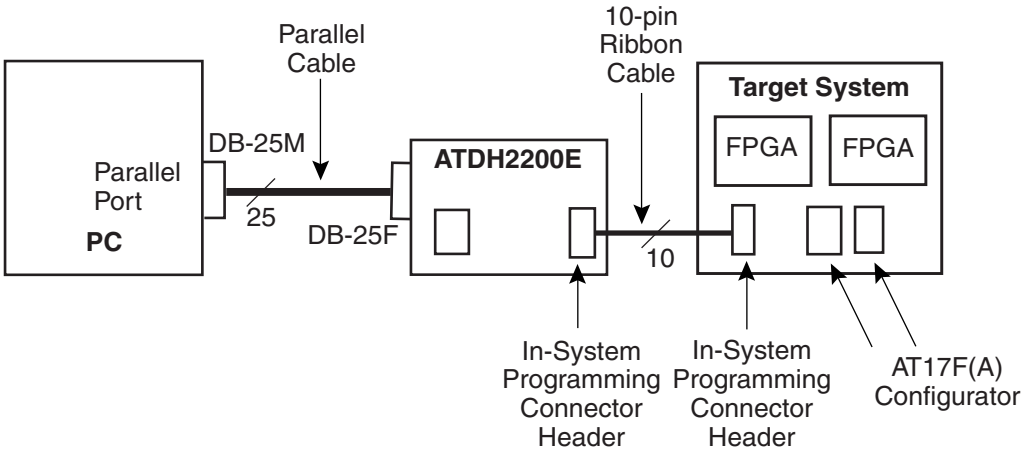
Flowcharts with exact details of each of the program mode procedures are provided in following sections.

## 10. In-System Programming Applications

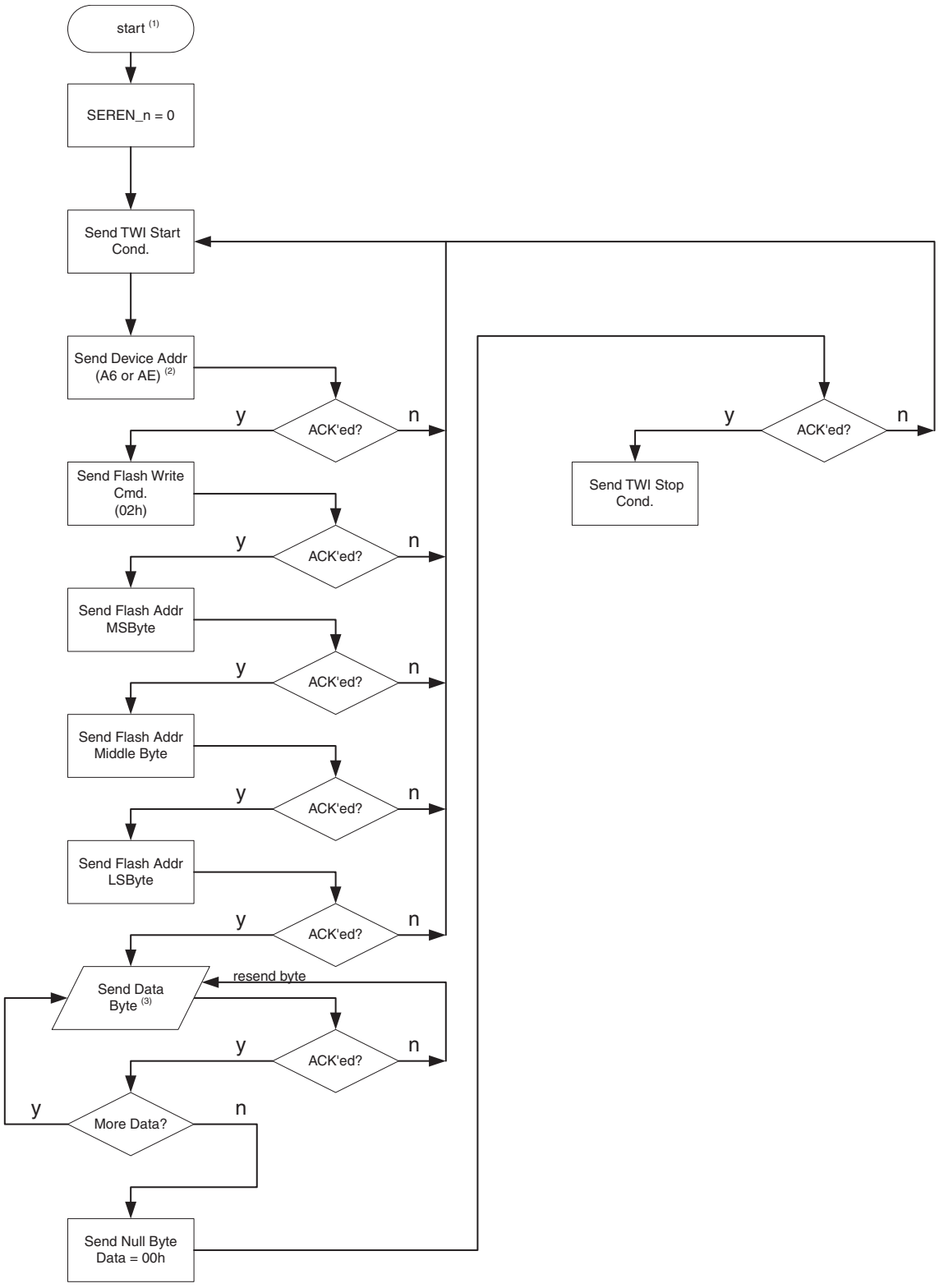
The AT17F Series Configurator is a (re)programmable device that can be programmed by the Atmel ATDH2200E programming system and a number of other 3rd party programmers. Additionally, they are In-System-Programmable (ISP), allowing the user to perform all programming procedures described in this document.

As shown in Figure 10-1, ISP can be performed with the aid of the Atmel ATDH2200E programming system or by means of those described in applications notes on the Atmel web site. Applications notes with circuit schematics for Atmel, Xilinx®, and Altera® FPGA's can be found on the Atmel web site, at http://www.atmel.com/dyn/products/app_notes.asp?family_id=625

**Figure 10-1.** Typical System Setup using the ATDH2200E Programmer



## 10.1 Program Mode Flow Chart: Write to Device

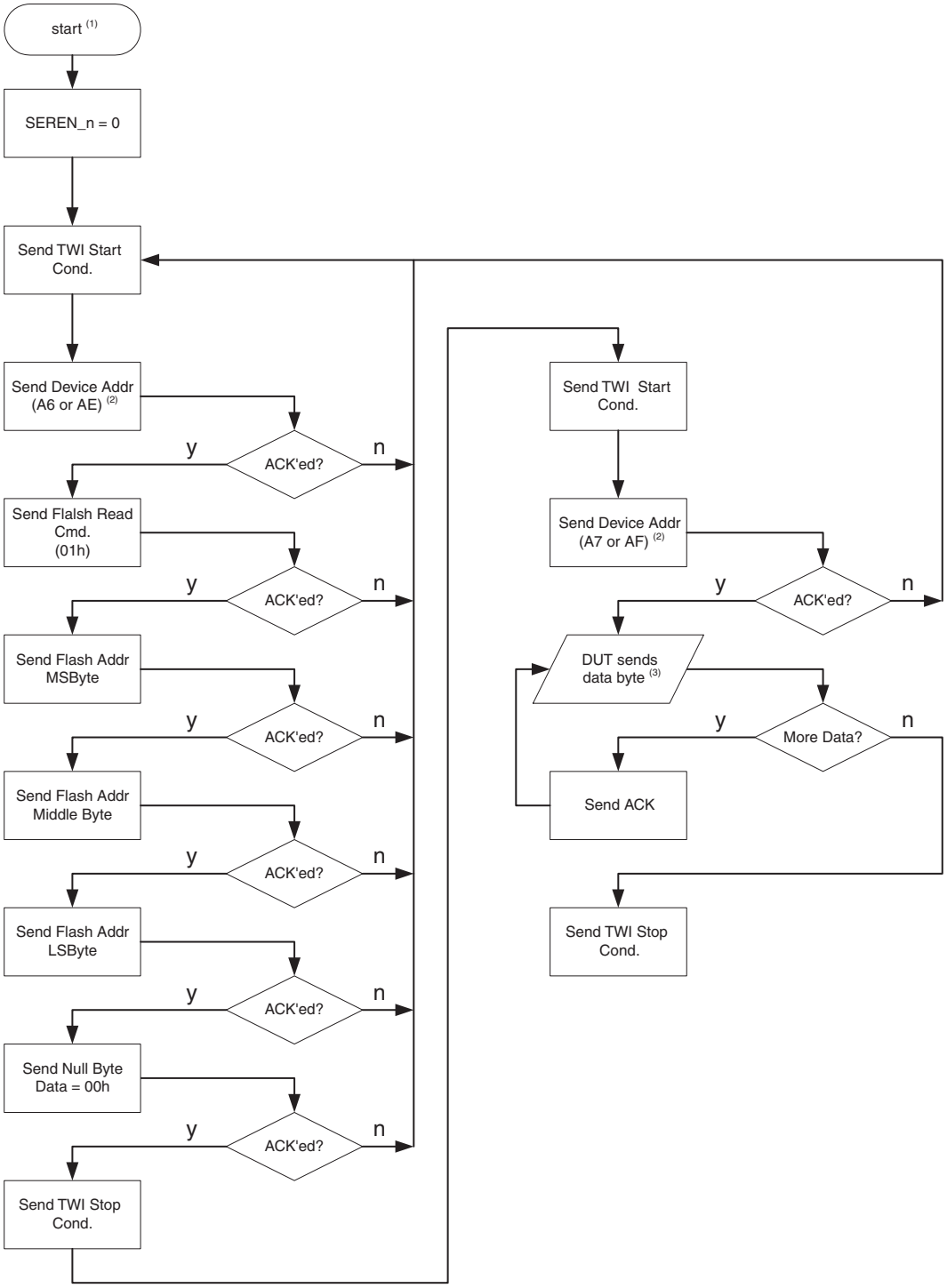



Notes: 1. Pull-up resistor required on DATA line (4.7 kΩ).

2. Pull-up $\overline{CEO}$/A2 pin for AE/h. Pull-down $\overline{CEO}$/A2 pin for A6/h (4.7 kΩ).

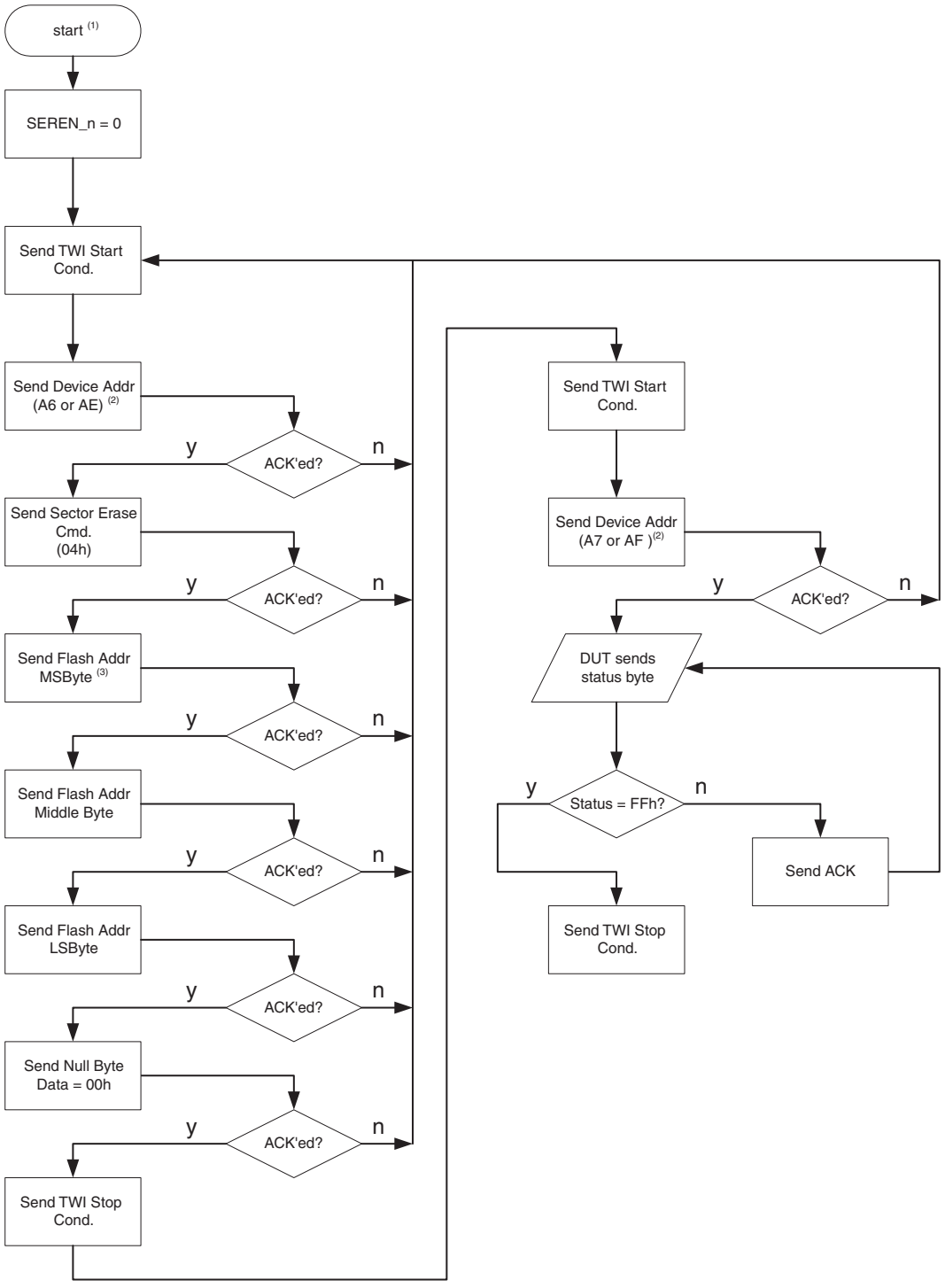
3. Total number of bytes sent must be an even number.

## 10.2 Program Mode Flow Chart: Read from Device



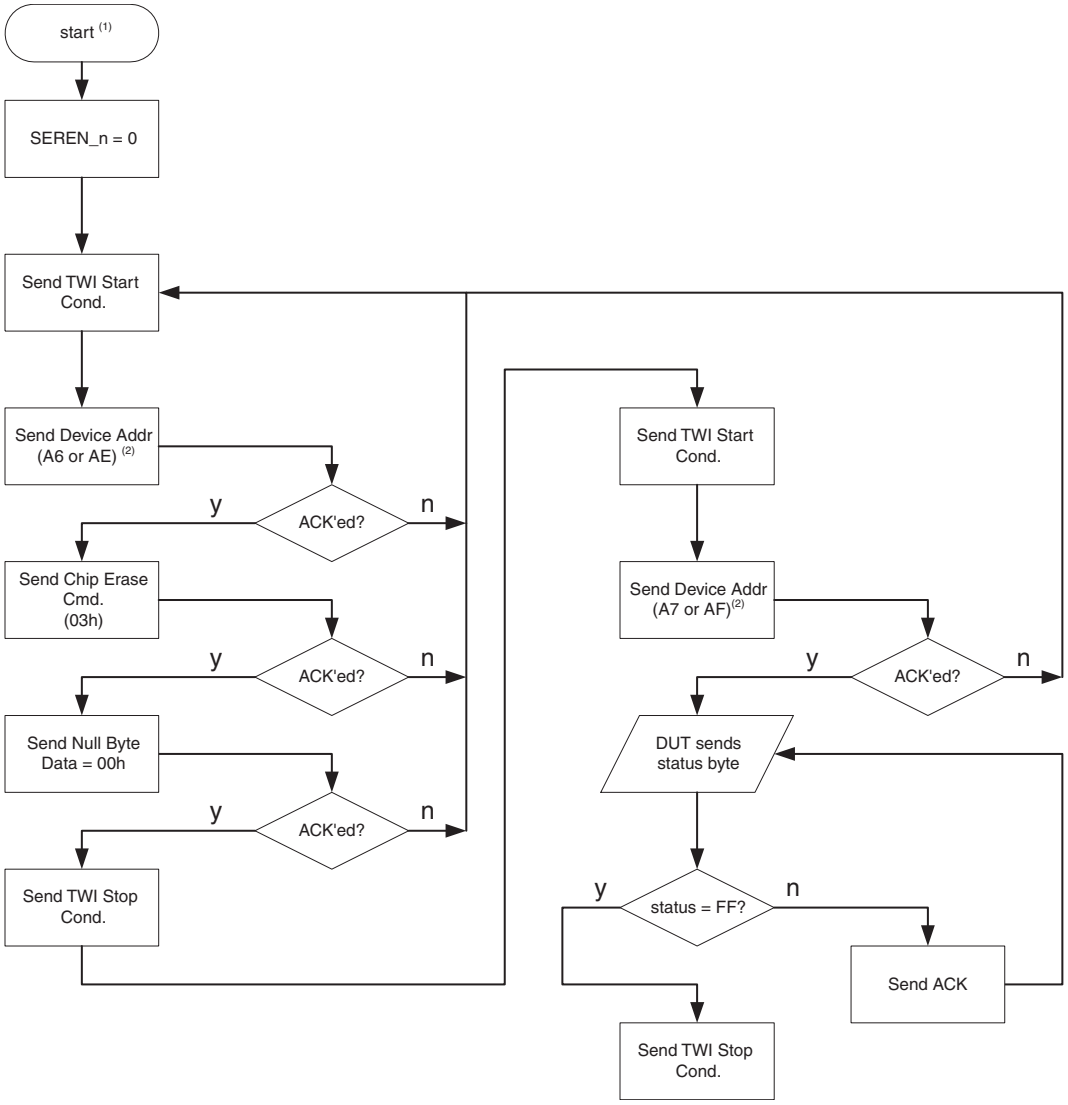

Notes: 1. Pull-up resistor required on DATA line (4.7 kΩ).
2. Pull-up $\overline{CEO}$/A2 pin for AE/h. Pull-down $\overline{CEO}$/A2 pin for A6/h (4.7 kΩ).
3. Total number of bytes sent must be an even number.

## 10.3 Program Mode Flow Chart: Sector Erase Device



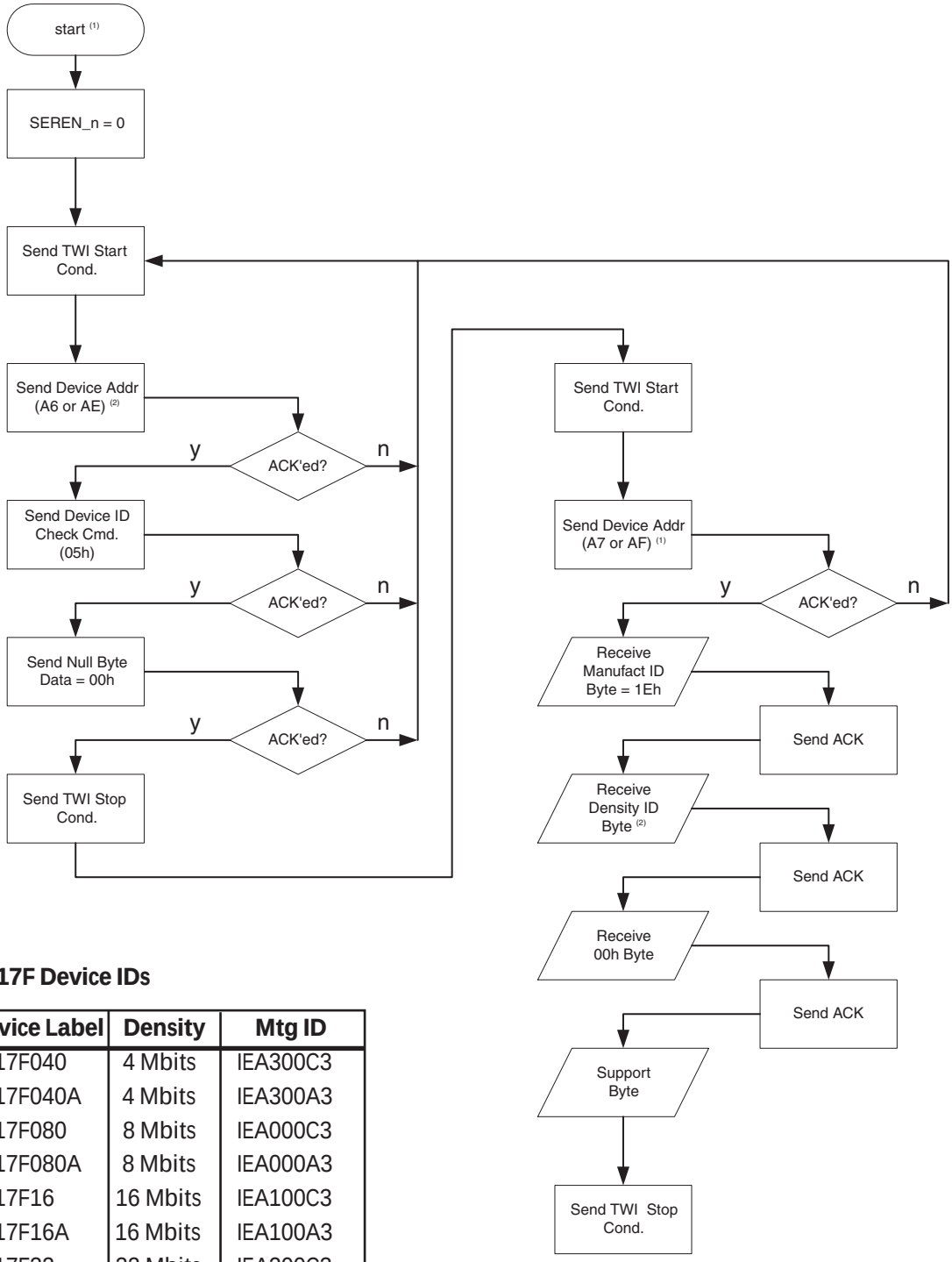

Notes:
1. Pull-up resistor required on DATA line (4.7 kΩ).
2. Pull-up $\overline{CEO}$/A2 pin for AE/h. Pull-down $\overline{CEO}$/A2 pin for A6/h (4.7 kΩ).
3. Total number of bytes sent must be an even number.

## 10.4 Program Mode Flow Chart: Chip Erase Device




Notes:
1. Pull-up resistor required on DATA line (4.7 kΩ).
2. Pull-up $\overline{CEO}$/A2 pin for AE/h. Pull-down $\overline{CEO}$/A2 pin for A6/h (4.7 kΩ).
3. Total number of bytes sent must be an even number.

## 10.5 Program Mode Flow Chart: Device ID Check




**AT17F Device IDs**

| Device Label | Density | Mtg ID |
|---|---|---|
| AT17F040 | 4 Mbits | IEA300C3 |
| AT17F040A | 4 Mbits | IEA300A3 |
| AT17F080 | 8 Mbits | IEA000C3 |
| AT17F080A | 8 Mbits | IEA000A3 |
| AT17F16 | 16 Mbits | IEA100C3 |
| AT17F16A | 16 Mbits | IEA100A3 |
| AT17F32 | 32 Mbits | IEA200C3 |
| AT17F32A | 32 Mbits | IEA200A3 |

Notes: 1. Pull-up resistor required on DATA line (4.7 kΩ).
2. Pull-up $\overline{CEO}$/A2 pin for AE/h. Pull-down $\overline{CEO}$/A2 pin for A6/h (4.7 kΩ).
3. Total number of bytes sent must be an even number.

## 10.6 Sector Address Tables

**Table 10-1.** AT17F040(A) – Sector Address Table

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA0 | 8K | 00000 - 01FFF |
| SA1 | 4K | 02000 - 02FFF |
| SA2 | 4K | 03000 - 03FFF |
| SA3 | 240K | 04000 - 3FFFF |

**Table 10-2.** AT17F080(A) – Sector Address Table

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA0 | 8K | 00000 - 01FFF |
| SA1 | 4K | 02000 - 02FFF |
| SA2 | 4K | 03000 - 03FFF |
| SA3 | 496K | 04000 - 7FFFF |

**Table 10-3.** AT17F16(A) – Sector Address Table

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA0 | 4K | 00000 - 00FFF |
| SA1 | 4K | 01000 - 01FFF |
| SA2 | 4K | 02000 - 02FFF |
| SA3 | 4K | 03000 - 03FFF |
| SA4 | 4K | 04000 - 04FFF |
| SA5 | 4K | 05000 - 05FFF |
| SA6 | 4K | 06000 - 06FFF |
| SA7 | 4K | 07000 - 07FFF |
| SA8 | 32K | 08000 - 0FFFF |
| SA9 | 32K | 10000 - 17FFF |
| SA10 | 32K | 18000 - 1FFFF |
| SA11 | 32K | 20000 - 27FFF |
| SA12 | 32K | 28000 - 2FFFF |
| SA13 | 32K | 30000 - 37FFF |
| SA14 | 32K | 38000 - 3FFFF |
| SA15 | 32K | 40000 - 47FFF |
| SA16 | 32K | 48000 - 4FFFF |
| SA17 | 32K | 50000 - 57FFF |
| SA18 | 32K | 58000 - 5FFFF |
| SA19 | 32K | 60000 - 67FFF |
| SA20 | 32K | 68000 - 6FFFF |
| SA21 | 32K | 70000 - 77FFF |
| SA22 | 32K | 78000 - 7FFFF |

**Table 10-3.** AT17F16(A) – Sector Address Table (Continued)

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA23 | 32K | 80000 - 87FFF |
| SA24 | 32K | 88000 - 8FFFF |
| SA25 | 32K | 90000 - 97FFF |
| SA26 | 32K | 98000 - 9FFFF |
| SA27 | 32K | A0000 - A7FFF |
| SA28 | 32K | A8000 - AFFFF |
| SA29 | 32K | B0000 - B7FFF |
| SA30 | 32K | B8000 - BFFFF |
| SA31 | 32K | C0000 - C7FFF |
| SA32 | 32K | C8000 - CFFFF |
| SA33 | 32K | D0000 - D7FFF |
| SA34 | 32K | D8000 - DFFFF |
| SA35 | 32K | E0000 - E7FFF |
| SA36 | 32K | E8000 - EFFFF |
| SA37 | 32K | F0000 - F7FFF |
| SA38 | 32K | F8000 - FFFFF |

**Table 10-4.** AT17F32(A) – Sector Address Table

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA0 | 4K | 00000 - 00FFF |
| SA1 | 4K | 01000 - 01FFF |
| SA2 | 4K | 02000 - 02FFF |
| SA3 | 4K | 03000 - 03FFF |
| SA4 | 4K | 04000 - 04FFF |
| SA5 | 4K | 05000 - 05FFF |
| SA6 | 4K | 06000 - 06FFF |
| SA7 | 4K | 07000 - 07FFF |
| SA8 | 32K | 08000 - 0FFFF |
| SA9 | 32K | 10000 - 17FFF |
| SA10 | 32K | 18000 - 1FFFF |
| SA11 | 32K | 20000 - 27FFF |
| SA12 | 32K | 28000 - 2FFFF |
| SA13 | 32K | 30000 - 37FFF |
| SA14 | 32K | 38000 - 3FFFF |
| SA15 | 32K | 40000 - 47FFF |
| SA16 | 32K | 48000 - 4FFFF |
| SA17 | 32K | 50000 - 57FFF |
| SA18 | 32K | 58000 - 5FFFF |

**Table 10-4.** AT17F32(A) – Sector Address Table (Continued)

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA19 | 32K | 60000 - 67FFF |
| SA20 | 32K | 68000 - 6FFFF |
| SA21 | 32K | 70000 - 77FFF |
| SA22 | 32K | 78000 - 7FFFF |
| SA23 | 32K | 80000 - 87FFF |
| SA24 | 32K | 88000 - 8FFFF |
| SA25 | 32K | 90000 - 97FFF |
| SA26 | 32K | 98000 - 9FFFF |
| SA27 | 32K | A0000 - A7FFF |
| SA28 | 32K | A8000 - AFFFF |
| SA29 | 32K | B0000 - B7FFF |
| SA30 | 32K | B8000 - BFFFF |
| SA31 | 32K | C0000 - C7FFF |
| SA32 | 32K | C8000 - CFFFF |
| SA33 | 32K | D0000 - D7FFF |
| SA34 | 32K | D8000 - DFFFF |
| SA35 | 32K | E0000 - E7FFF |
| SA36 | 32K | E8000 - EFFFF |
| SA37 | 32K | F0000 - F7FFF |
| SA38 | 32K | F8000 - FFFFF |
| SA39 | 32K | 100000 - 107FFF |
| SA40 | 32K | 108000 - 10FFFF |
| SA41 | 32K | 110000 - 117FFF |
| SA42 | 32K | 118000 - 11FFFF |
| SA43 | 32K | 120000 - 127FFF |
| SA44 | 32K | 128000 - 12FFFF |
| SA45 | 32K | 130000 - 137FFF |
| SA46 | 32K | 138000 - 13FFFF |
| SA47 | 32K | 140000 - 147FFF |
| SA48 | 32K | 148000 - 14FFFF |
| SA49 | 32K | 150000 - 157FFF |
| SA50 | 32K | 158000 - 15FFFF |
| SA51 | 32K | 160000 - 167FFF |
| SA52 | 32K | 168000 - 16FFFF |
| SA53 | 32K | 170000 - 177FFF |
| SA54 | 32K | 178000 - 17FFFF |
| SA55 | 32K | 180000 - 187FFF |
| SA56 | 32K | 188000 - 18FFFF |

**Table 10-4.** AT17F32(A) – Sector Address Table (Continued)

| Sector | Size (Words) | Address Range |
|---|---|---|
| SA57 | 32K | 190000 - 197FFF |
| SA58 | 32K | 198000 - 19FFFF |
| SA59 | 32K | 1A0000 - 1A7FFF |
| SA60 | 32K | 1A8000 - 1AFFFF |
| SA61 | 32K | 1B0000 - 1B7FFF |
| SA62 | 32K | 1B8000 - 1BFFFF |
| SA63 | 32K | 1C0000 - 1C7FFF |
| SA64 | 32K | 1C8000 - 1CFFFF |
| SA65 | 32K | 1D0000 - 1D7FFF |
| SA66 | 32K | 1D8000 - 1DFFFF |
| SA67 | 32K | 1E0000 - 1E7FFF |
| SA68 | 32K | 1E8000 - 1EFFFF |
| SA69 | 32K | 1F0000 -1F7FFF |
| SA70 | 32K | 1F8000 - 1FFFF |

## Headquarters

***Atmel Corporation***
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

## International

***Atmel Asia***
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
Tel: (852) 2245-6100
Fax: (852) 2722-1369

***Atmel Europe***
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

***Atmel Japan***
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

## Product Contact

***Web Site***
www.atmel.com

***Technical Support***
configurator@atmel.com

***Sales Contact***
www.atmel.com/contacts

***Literature Requests***
www.atmel.com/literature