Programming Atmel's Family Of Flash Memories

Introduction

Atmel offers a diverse family of Flash Memory devices ranging in density from 256 K to 4M-bits. These devices read and program with a single voltage supply. The nominal supply voltage is 5V for the AT29Cxxx, 3.3V for the "low voltage" AT29LVxxx, and 3V for the "*Battery-Voltage*™" AT29BVxxx Flash family. The entire Flash family is designed to allow users to have one common programming algorithm for all three Flash voltage families. Therefore, upgrading from one density to another and from a higher voltage to a lower voltage device is simplified.

This application note describes the design benefits of Atmel's Flash architecture as well as how the device ID feature is used to adjust for varying densities and supply voltages. In addition, Atmel's Software Data Protection (SDP) feature, which prevents inadvertent writes, is described. An example is given to illustrate the ease with which the programming software can be written to accommodate four different 4M-bit Flash devices, the AT29C040, AT29LV040, AT29C040A, and AT29LV040A.

Hardware and software have been developed to demonstrate the relevant design issues. The demo uses an AT89C51 Flash-based microcontroller (which has the same pinout and instruction set as an 80C51) as the host processor and a "C" language program for the software. The software automatically adjusts the amount of time required for programming the varying voltage versions of the 4M-bit Flash devices in addition to accommodating for their different sector sizes.

The AT89C51, a member of Atmel's growing family of Flash microcontroller devices, features 4K bytes of in-system reprogrammable Flash memory (see Atmel application note "AT89C51 In-Circuit Programming" for additional information). Current and future versions of Atmel's microcontroller family incorporate from as little as 1K bytes of Flash memory to as much as 128K bytes, providing many density options for different applications. Other versions will also include special architectures such as a combination of Flash and parallel EEPROM memory on board.

Programming Flash Devices

Unlike Atmel's Flash Memories, previous generations of Flash memories had large kilobyte sectors and required that an entire sector be erased prior to programming. Generally, the sector erase cycle time was hundreds or thousands of milliseconds and could be as long as 30 seconds for the entire memory array. In addition, a separate high voltage supply was required for a write and erase operation. Atmel's Flash family has simplified usage by having only one supply voltage, reducing the sector size, having the programming similar to an SRAM write operation, and decreasing significantly the total programming time.

Small sector sizes reduce the amount of system resources necessary for programming. When only a few bytes in a Flash memory need to be altered, a RAM image of the Flash sector must be created. The RAM must then be altered with the new data, and the image transferred back into the Flash device. Because Atmel's Flash devices have

0510A-B–12/97

small sector sizes (from 64 to 512 bytes, depending on the memory density), the RAM requirements are much less than those of large sector Flash devices. The latter generally have 4 K to 128K byte sector sizes.

A second advantage of Atmel's Flash is that an entire sector can be updated during a single program operation, instead of the byte-by-byte programming of previous generation Flash memories. This saves significant programming time when updating an entire sector, especially when comparing Atmel's small sector devices with large sector devices. In addition, Atmel's devices do not require a sector erase prior to writing, thus saving additional programming time. The maximum sector program time is 10 ms and 20 ms for the AT29Cxxx and AT29LVxxx/AT29BVxxx families respectively.

AT29C040 and AT29C040A Architecture

The AT29C040 provides operation similar to a byte-wide SRAM. The device has eight data lines and 19 address lines. The familiar three input control lines are also present ($\overline{CE}$, $\overline{OE}$, $\overline{WE}$). Read operations are identical to an SRAM, but write operations are somewhat different due to the write cycle time (t_{WC}) requirements of all Flash memories. Flash write operations take several milliseconds to complete, compared to the nanosecond writes of SRAM devices. It should be noted that Atmel's Flash Memories require only a write operation; the erase operation is automatically performed internally in the device.

Data is loaded into the AT29C040 one sector at a time, with each sector consisting of 512 bytes. The sector chosen for modification is defined by the upper order address bits (A9-A18). The entire sector must be loaded during the write operation. Any byte not loaded during the sector load will contain FFH after the write operation has completed. Address lines A0 through A8 define the location of the bytes within a sector. All data must be loaded into the same sector (A9 through A18 must remain constant) and can be randomly loaded within that sector.

The AT29C040A is identical to the AT29C040 except for the sector size and the Device ID Code (the Device ID Code is described later). The AT29C040A has a 256 byte sector (instead of a 512 byte sector) which is defined by address lines A8 through A18; the bytes within the sector are determined by address lines A0 through A7.

Software Data Protection (SDP)

One concern of systems designers when using nonvolatile programmable memories is the possibility of inadvertent write operations that can be caused by noise or by power-up and power-down sequences. Atmel's Flash memories provide a feature called Software Data Protection (SDP) that addresses this issue. The user can enable SDP upon receipt of the device from Atmel, and its usage is highly recommended. Data can be written into a sector with or without SDP enabled. However, once SDP has been enabled, the device requires that all subsequent write operations perform a series of "dummy" write operations before loading the chosen sector with data. The "dummy" writes consist of loading three known data values into three predefined addresses. This three-byte sequence preceding a write operation virtually eliminates the chance of inadvertent write operations. The sequence is described below.

1. Load Data AAH into Address 05555H
2. Load Data 55H into Address 02AAAH
3. Load Data A0H into Address 05555H
4. Load desired sector with data
5. Pause t_{WC} (device write cycle time)
6. The device is returned to standard operating mode

If SDP is enabled, any attempt to write to the device without the three-byte command sequence will start a write cycle. However, no data will actually be written to the device, and during this "write" cycle time (t_{WC}), valid data cannot be read from the Flash.

Product and Manufacturer ID

Atmel's Flash memory devices allow the user to access both device and manufacturer information. This feature allows a system to determine exactly which Flash memory is being used. Once this is known, the host system can choose different algorithms for write operations in order to accommodate for differences in device density, V_{CC} requirements, sector size, and required write cycle time.

Product and manufacturer ID information is determined with the Software Product Identification procedure, which is similar to the Software Data Protection sequence. The sequence is described below.

1. Load Data AAH into Address 05555H
2. Load Data 55H into Address 02AAAH
3. Load Data 90H into Address 05555H
4. Pause t_{WC} (device write cycle time)
5. Read Address 00000H
   Data read is the Manufacturer Code

6. Read Address 00001H
   Data read is the Device ID Code
7. Load Data AAH into Address 05555H
8. Load Data 55H into Address 02AAAH
9. Load Data F0H into Address 05555H
10. Pause t_{WC} (device write cycle time)
11. The device is returned to standard operating mode

The following table uses the 4M-bit Flash as an example to illustrate the pertinent device information than can be determined once the Device ID Code is known.

Device	ID	V_{CC}	Sector Size	t_{WC}
AT29C040	5B	5.0V ± 10%	512 bytes	10 ms
AT29C040A	A4	5.0V ± 10%	256 bytes	10 ms
AT29LV040	3B	3.3V ± 0.3V	512 bytes	20 ms
AT29LV040A	C4	3.3V ± 0.3V	256 bytes	20 ms
AT29BV040	3B	3.0V ± 10%	512 bytes	20 ms
AT29BV040A	C4	3.0V ± 10%	256 bytes	20 ms

Hardware Description

The demo hardware consists of a 12 MHz AT89C51 Flash-based microcontroller with 4K bytes of on-board Flash memory. The internal AT89C51 Flash memory is used for boot code, and the external 8K x 8 SRAM and the AT29C040 are mapped as data memory. The AT29C040 is also mapped as program memory to facilitate off-chip program execution. The AT89C51 can only access a maximum of 64K bytes of data memory space, while the AT29C040 has 512K bytes of storage capacity. To solve this size mismatch, the AT29C040 is bank switched into the AT89C51 data memory map in 8K byte blocks. The bank switching is performed with six general purpose I/O port bits on the AT89C51. The system address map is shown below.

System Address Map

AT89C51 Microcontroller	0000-1FFF	Internal program memory
8K x 8 Static RAM	2000-3FFF	Data memory
AT29C040 Flash	4000-5FFF	Program and data memory

Software Description

The software (available from Atmel's BBS 408-436-4309) demonstrates how the Device ID Code can be used to allow a single program to work with different Atmel Flash memories. The program uses Atmel's 4M-bit Flash (AT29C040, AT29LV040, AT29C040A, and AT29LV040A) as an example, but the software can be easily adapted to accommodate other device densities.

In order to program the Flash memory, the software must first determine which Flash device is being used. This is accomplished by first putting the device into the Software Product Identification mode by executing a three-byte command sequence (described in the "Product and Manufacturer ID" section of this application note). The program subsequently reads the Device ID Code and executes another three-byte command sequence to return the Flash to the standard operating mode. Using the Device ID Code, the program then determines the appropriate sector size and write cycle time ($t_{WC)}$ for the particular 4M-bit Flash being used.

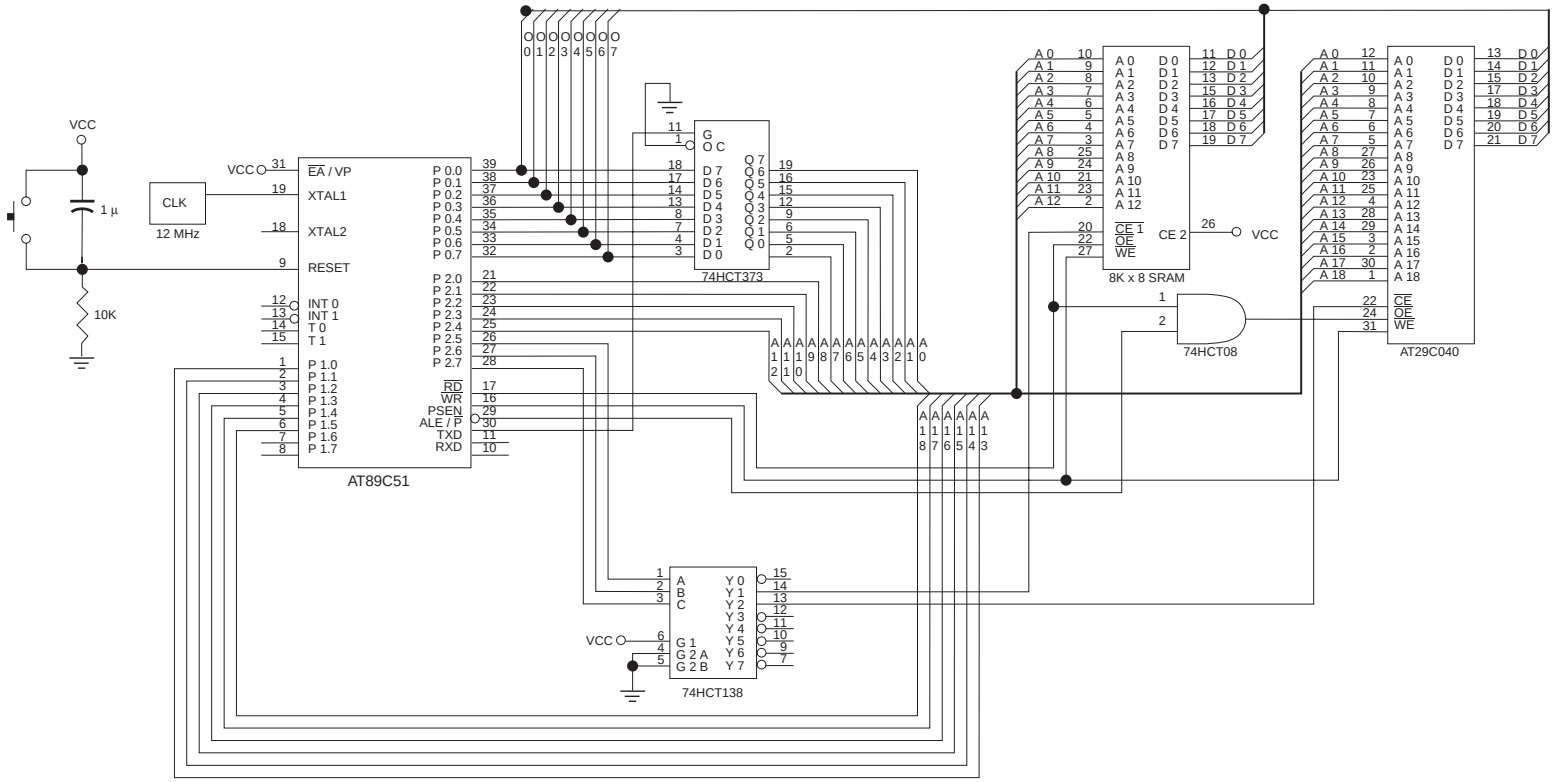
To demonstrate a sector write, the program proceeds to load the SRAM with "dummy" data. After the data has been loaded, the program transfers the data from the SRAM to a predefined sector (within one of the mapped 8K byte blocks) of the 4M-bit Flash. After pausing the required write cycle time ($t_{WC),}$ the sector that was just written is transferred back to the SRAM buffer.

Summary

Atmel's Flash Memories are designed to allow all densities and device configurations to be programmed using the same programming algorithm. The user has to simply determine the Device ID Code and set the appropriate sector size and write cycle time. This operation need only be performed once provided the sector size and write cycle information is saved. If only one density or configuration will ever be used, then reading of the Device ID Code can be eliminated, and the sector size and write cycle information can be predefined in the software.

As demonstrated, programming Atmel's Flash is a simple process, similar to loading an SRAM. Architectural and circuit features within the devices minimize software and system overhead while simplifying programming procedures. Atmel's Flash Memories require only about one-tenth of the typical software, buffer memory, and performance overhead of previous generation Flash, thus providing substantial system cost savings.

Figure 1. Atmel AT29C040DA Demo Circuit



Note: If the Flash is to be used as external program memory, then pin 31 ($\overline{EA}$/ V_{PP}) of the AT89C51 should be connected to ground.