AN1158
APPLICATION NOTE

Uniform vs. Boot Block Flash Architectures

CONTENTS

INTRODUCTION

The choosing between a Uniform Block and a Boot Block can trouble designers far more than the choice merits. Many applications using Flash do not make use of Flash's internal Block Architecture and therefore ignore it.

This application note aims to categorize the data types stored in Flash memory and help the designer choose which type of Block Architecture they require.

BLOCKS

The difference between EEPROM and Flash, from a system view-point, is the way that the two types are erased. EEPROM can be erased one byte at a time, whereas Flash can only be erased one Block at a time. Typically a Block contains 64 Kbytes, though other Block sizes exist.

Because Flash can only be erased in Blocks, changing a single memory location in the Block is not straight forward. There are many different algorithms for storing data in Flash that allow single parameters to be updated without reading the entire Block, erasing it and re-programming it.

Read and program operations are independent of Blocks. Bytes (or words) can be read individually, regardless of Block boundaries. Erased bytes (or words) can be programmed individually too.

UNIFORM BLOCK ARCHITECTURE

All the Blocks in a Uniform Block Flash memory are the same size. Table 1 lists some typical parts, together with the number of Blocks in them and their Block sizes.

Table 1. Typical Uniform Block Flash

Flash	Size	Number of Blocks	Block Size
M29F010B, M29W010B	128Kb x8	8	16 Kbytes
M29F040B, M29W040B	512Kb x8	8	64 Kbytes
M29F080A	1024Kb x8	16	64 Kbytes
M29F016B	2048Kb x8	32	64 Kbytes

Uniform Block Architecture Flash memories are usually used in applications that use all Blocks for the same purpose, for example for a Flash File System.

BOOT BLOCK ARCHITECTURES

Boot Block Flash memory usually contains four different sizes of Block. Usually there are one Boot Block, two Parameter Blocks and a small Main Block as well as the normal Main Blocks. The Boot Block, Parameter Blocks and small Main Block can be located at the top or bottom of the Flash's memory space, depending on whether a Top Boot Block or Bottom Boot Block part is chosen. Table 2 gives the Block address locations in the M29W004BB Boot Block Memory.

Table 2. M29W004BB Block Addresses

Size (Kbytes)	Address Range	Block Type
64	70000h-7FFFFh	Main Block
64	60000h-6FFFFh	Main Block
64	50000h-5FFFFh	Main Block
64	40000h-4FFFFh	Main Block
64	30000h-3FFFFh	Main Block
64	20000h-2FFFFh	Main Block
64	10000h-1FFFFh	Main Block
32	08000h-0FFFFh	Small Main Block
8	06000h-07FFFh	Parameter Block
8	04000h-05FFFh	Parameter Block
16	00000h-03FFFh	Boot Block

The Boot Block of the M29W004BB is positioned at the bottom of the Flash's address space because many microprocessors have their Reset vector at 0000h. If the Flash is mapped to address 0000h of the microprocessor's address space then the microprocessor can boot from the Flash's Boot Block. There are many microprocessors that boot from the bottom of their memory space, such as the ST10, Motorola 68000 series, Power PC etc.

The Top Boot Block equivalent, the M29W004BT, has the Boot Block at the top of the Flash's address space. Figure 3 shows the Block address locations for the M29W004BT.

Table 3. M29W004BT Block Addresses

Size (Kbytes)	Address Range	Block Type
16	7C000h-7FFFFh	Boot Block
8	7A000h-7BFFFh	Parameter Block
8	78000h-79FFFh	Parameter Block
32	70000h-77FFFh	Small Main Block
64	60000h-6FFFFh	Main Block
64	50000h-5FFFFh	Main Block
64	40000h-4FFFFh	Main Block
64	30000h-3FFFFh	Main Block
64	20000h-2FFFFh	Main Block
64	10000h-1FFFFh	Main Block
64	00000h-0FFFFh	Main Block

Top Boot Block Flash memory are intended for microprocessors that begin execution by reading from the top of their memory space. Typical examples include Intel i960, 80x86 etc.

DATA CATEGORIZATION

Some applications have no need to make use of the block architecture. Designs using Flash purely for code storage (as EPROM or ROM is used) need only erase the whole memory and reprogram it. The Flash offers advantages over EPROM in that the hardware can be built and tested with a test program prior to the application program being loaded. Using Flash alleviates the need to remove any components since the application can be programmed onto the board without removing the Flash. Two or more products may share the same hardware, but include different features that are dependent on the application software.

Many applications do, however, make good use of the erase and reprogram facilities of Flash. The information that is stored in the Flash can be categorized in to Boot Code, Application Code, User Parameters and User Data.

Boot Code is the code and data that the microprocessor requires to Boot. If this is to be included in the Flash then ideally a Boot Block Flash would be used. Boot code rarely exceeds the 16 Kbyte size of the typical Boot Block. The Boot Block should never be erased and many products will protect their Boot Block to ensure this. The boot code could include functionality to replace the application code. If the application code becomes corrupt then the boot code can replace it. If the boot code becomes corrupt the system may become unusable and the Flash may need to be removed from the board for reprogramming.

Application Code is the code that the user sees running. Often the application code will not be run from the Flash, but will be copied into another memory, such as DRAM, before being run. Many systems have 32-bit wide DRAM, but only 8-bit wide Flash, so running from DRAM is faster. The boot code is often responsible for copying the application code before running it. The application code does not need different Blocks, it is usually replaced all at once, so one erase operation is sufficient.

User Parameters configure the system as the user likes it. Many systems do not have very many options, and a 64 Kbyte block is overkill for most user settings. The Parameter Blocks provide an ideal storage solution for user settings. Since there are two Parameter Blocks one can be erased while the other stores the current parameters. Parameter Blocks erase quickly, since they are small, so the user does not have to wait long for the Parameter Blocks to erase.

User Data can be any data the user wishes to store. It may include Flash File Systems, Databases, etc. Uniform blocks are best for User Data storage because then any erased block can be used to store the data from another block when it is changed.

As an example consider how a PC BIOS might use a Flash memory. A Top Boot Block memory would be chosen to boot the 80x86 microprocessor. The BIOS would take the place of the Application Code, being shadowed into DRAM before execution. The BIOS settings would be stored in the Parameter Blocks so that, in the event of the battery backed-up SRAM failing, the BIOS settings would still be valid. Finally the remaining Flash (all Main Blocks, so uniform in size) would be used for a Flash File System that contains the operating system and other applications.

By contrast a PCMCIA Flash Card would use Uniform Block Flash in order to simplify the implementation of a Flash File System.

CONCLUSION

Choosing between Uniform Block and Boot Block Flash memory requires knowledge of what data will be stored in the Flash memory. Many applications do not make use of the Block Architecture, and therefore they can use Uniform or Boot Block parts. Other applications use different areas of the Flash for different purposes, Boot Block architectures are advantageous to them. Finally some applications treat all Blocks the same and are therefore more suited to Uniform Block Architectures.

If you have any questions or suggestion concerning the matters raised in this document please send them to the following electronic mail address:

ask.memory@st.com (for general enquiries)

Please remember to include your name, company, location, telephone number and fax number.

littlediode
COMPONENTS FOR ALL

Littlediode supplies new, hard to find or obsolete electronic components and semiconductors all over the world.

With over two million different components listed you are sure to find the part you need.

Feel free to visit us today at our online store:

LittleDiode.com

Looking forward to providing you with the best possible service.