Abstract

The fundamental principles of ferroelectric capacitor memory (FECM), a new, solid state technology for implementing non-volatile computer memory, are discussed. The advantages and disadvantages of using FECM are presented. FECM has properties that make it highly desirable for portable, embedded, space-based, and weapon-based computers, including non-volatility, long retention, high reliability, low fatigue rate, and tolerance to ionizing radiation. However, interfacing a microprocessor (μPC) system to current serial-access FECM devices can be difficult. This paper describes an interface that maps a block of FECM into the address space of a μPC such that the μPC can write and read the memory as if it were a fully decoded silicon SRAM. The circuit has been bread boarded and tested and found to be 100% functional. A CMOS VLSI implementation of the interface is feasible and is being considered. Applications for FECMs and the proposed VLSI interface chip in portable, embedded, space-based, and weapon-based computers are discussed.

1: Introduction

Ferroelectric capacitors are a new type of solid state device that can be used to implement non-volatile write/read memory for computers and other digital systems [1]. The concept of ferroelectric capacitor memory (FECM) is not new, although practical devices that utilize ferroelectric principles have only recently become available [2]. It should be noted that the only link between ferroelectrics and ferromagnetics is the fact that the Q-V hysteresis loop of a ferroelectric device is similar to the B-H loop of a ferromagnetic device.

The term ferroelectric is deceptive because no iron is actually used in the device. As shown in Fig. 1, ferroelectric capacitors consist of a layer of ferroelectric material, such as lead zirconate titanate (PZT), serving as the dielectric between 2 electrodes. Fig. 1 also illustrates how ferroelectric capacitors can be fabricated on a silicon substrate along with CMOS transistors. This allows both high-density, low-power logic and non-volatile, write/read memory to be fabricated on the same IC.

To store data in a ferroelectric capacitor, a voltage is applied across it. The bipolar molecules in the dielectric align in one direction or the other, depending on the polarity of the applied voltage. The element is read by applying another voltage across the capacitor and observing the charging current, as indicated by Fig. 2. A small current pulse indicates proper alignment of the molecules. A large current pulse indicates that the molecules had to realign in response to the applied voltage. Therefore, a logic 0 or 1 can be determined by the switching current present whenever the sampling reference is applied. It should be noted that reading a FECM destroys the stored data, which must then be refreshed.

The advantages of using FECMs in a computer or digital system include: 1) non-volatility, 2) long retention, 3) high reliability, 4) a low fatigue rate, and 5) tolerance of ionizing radiation. Non-volatility is the biggest single advantage of using FECM. FECMs will retain stored data for more than 10 years without power, making them comparable to magnetic disk memory with respect to retention. However, FECMs have a tremendous reliability advantage over magnetic disks because they have no moving parts. A typical FECM can be written as many as $10^{10}$ times before fatigue makes it vulnerable to errors [3], which is vastly superior to the $10^5$ write cycle limit for typical electrically erasable programmable read only memory (EEPROM), the only other solid state, non-volatile write/read memory technology currently available. Another advantage that FECM has is that it is very tolerant of ionizing radiation [4].

The disadvantages of using FECM in a computer or digital system include: 1) complex interface and control cir-
Figure 1: Ferroelectric capacitor and CMOS transistor.

Figure 2: Ferroelectric capacitor voltage and current characteristics during a read operation.

The objective of the new interface circuit described here is to provide an interface between existing FECM ICs and a typical μPC system. The interface allows the FECM to appear as a normal block of SRAM, without requiring any special actions on the part of the μPC.

2: Interface Design

The μPC-FECM interface must accomplished 4 tasks in order to attain the desired objective. First, the parallel address and data busses of the μPC must be converted to a 1-bit serial bus for write operations. Second, the 1-bit serial bus of the FECM ICs must be converted to a 16-bit parallel data bus for read operations. Third, the asynchronous memory protocol of the μPC must be converted to the synchronous, bit-serial protocol of the FECM. Fourth, the relatively fast μPC memory cycle must be synchronized with the relatively slow memory cycle of the FECM for both write and read operations. The block diagram of the interface, shown in Fig. 3, illustrates how the μPC, the interface, and up to 8 FECM ICs can be incorporated into a system.

The data path required for converting the parallel data and address from the μPC to a serial bit stream, and the 1-bit serial data from the memory to 16-bit parallel data, is shown in Fig. 4. It consists of 2, 8-bit shift registers with both parallel and serial inputs, and both parallel and serial outputs. Each shift register interfaces to a single serial data and address bus that can connect to up to 4 FECM ICs. The odd and even bytes of the 16-bit μPC data word require 2 separate banks of FECM because each bank is restricted to a 1-bit word width, and because there may be times when the μPC will want to read or write a single byte of data. Each bank contains 4, 256-word by 1-bit, FECM ICs, and has a total storage capacity of 1,024 bytes. Between the 2 banks, total storage capacity is 2,048 bytes.

Control of the data path is rather complex because of the sequential and rigid timing requirements of the FECM, and also because of the large difference in memory cycle times between the μPC and the FECM. The FECM controller is shown in Fig. 5. It consist of a finite state machine (FSM), a counter, and 2 serial bit-pattern generators. The FSM controls all functions of the interface and the FECM ICs, in response to signals from the μPC. The counter is used to count the number of bits that have been shifted into or out of the parallel-to-serial or serial-to-parallel conversion shift registers. The bit-pattern generators are used to generate the appropriate serial bit patterns that must be transmitted to the FECMs.
3: Interface Implementation

The µPC-FECM interface is implemented on a wire-wrapped prototyping board along with 8 FECM ICs, an 8086 microprocessor [6], and some associated support logic. The implementation required a total of 58 small-scale and medium-scale TTL ICs, including 10 erasable programmable logic devices (EPLDs). The number of logic gates required is few enough such that it could be easily implemented on a single CMOS VLSI chip. This would greatly reduce the power consumption, weight, and physical size of the interface. The development of a VLSI interface chip would allow FECM to be used in µPC systems where power consumption, weight, and size are important design concerns, such as in portable, embedded, space-based, and weapon-based, computers [7].

The working prototype interface is capable of operating at a maximum memory bus bandwidth of 27.5K bits per second. This bandwidth is significantly lower than the maximum memory bandwidth that the µPC is capable of. The low bandwidth is attributed to the long, serial write and read cycles of the FECM.

4: Applications

Currently available FECM ICs are not intended to replace Si RAM because of their relatively long write and read cycle times. As the speed and density of FECM improves, it may eventually become competitive with Si RAM. However, at the present level of development, FECM is the preferred method for implementing small blocks of non-volatile, high-reliability, write/read memory in any computer or digital system. FECM should eventually replace EEPROM, and in fact, it already has in many new designs. Applications for FECM include the storage of configuration parameters and boot routines in turn-key systems, the storage of main programs and data constants in cold standby systems, and the storage of programs and data in any system where memory integrity must be maintained during power interruption.

A study is currently being conducted on the feasibility of using FECMs in spacecraft, especially in satellites. The long retention, high reliability, and tolerance of ionizing radiation make FECM ideal for storing semi-permanent programs and data, and for on-board back up of high-speed Si RAM that can be susceptible to permanent damage and single event upsets from ionizing radiation. To conserve power, the FECM can be kept in a power down state except when being written with new programs or data, or when being used to reload Si RAM after a failure or a software update. Its low fatigue rate

Figure 3: Block diagram of microprocessor-FECM interface.
would allow a block of FECM to be uploaded as often as is necessary with new programs and data.

5: Conclusions

The described µPC-FECM interface is an elegant solution to the problem of interfacing a block of FECM to a µPC. The interface allows the µPC to address the FECM as if it were just another block of normal Si RAM. The interface handles the problems of parallel-to-serial and serial-to-parallel data conversion, interface signal differences, and the extreme disparity in write and read cycle times between the FECM and the µPC. The complexity of the interface is such that it can be easily implemented on a single VLSI CMOS IC, allowing the use of FECM in µPC systems where power consumption, weight, and size are important design concerns, such as in portable, embedded, space-based, and weapon-based, computers.

Acknowledgements

The assistance of National Semiconductor Corporation in this project is greatly appreciated. This research was sponsored by the NPS Research Administration through a Research Initiation Grant. The support of the NPS and of the U.S. Navy is gratefully acknowledged.
Figure 5: Microprocessor-FECM interface controller.

References