Maxim > Design Support > Technical Documents > Application Notes > 1-Wire® Devices > APP 5042
Maxim > Design Support > Technical Documents > Application Notes > Memory > APP 5042

Keywords: digital counter, nonvolatile, nonresettable, NV, EPROM emulation mode, EEPROM, EPROM, eeprom, eprom

APPLICATION NOTE 5042

Implementing Nonvolatile, Nonresettable Counters for Embedded Systems

By: Bernhard Linke, Principal Member Technical Staff
Jun 08, 2011

Abstract: For warranty reasons it is desirable to count certain events such as power cycles, operating time, hard (pushbutton) resets, and timeouts. Although digital counters are easy to build, it is not that obvious how to make them nonvolatile and nonresettable. This article explains how to use the EPROM-emulation mode of readily available serial EEPROMs together with an unusual coding scheme to solve the problem.

A similar version of this article appeared in the May 15, 2011 issue of *EE Times* magazine.

Motivation

It is sometimes desirable for warranty reasons to count and record certain events such as power cycles, operating time, hard (pushbutton) resets, and timeouts. The traditional electronic counters for this purpose are built from flip-flops, using a binary code such as the one shown in Figure 1. The maximum count is reached when all flip-flops are set, so the size of the counter is determined by the maximum number of events to be counted during the interval of interest.

Bit Number										
n+1	n	...	8	7	6	5	4	3	2	1
2^n	2^{n-1}	...	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Figure 1. In natural binary code, the place value doubles from one bit to the next.

Addressing the Requirements

Although a counter based on flip-flops is easily built, it has major drawbacks. It rolls over to zero (resets itself) when it reaches the counting limit. Its operation is also volatile—it requires power to maintain the state of its count. The first issue can be addressed by freezing the counter when it reaches its limit; you can resolve the second issue by attaching a battery to maintain power on the counter. But these measures may not be acceptable, because they add cost and work only for a limited time.

There is an alternative. You can back up the count value in EEPROM or some other form of nonvolatile (NV) storage. On the next power-up, the counter can be preloaded with the value stored in this NV memory. Unless the NV memory is embedded in another chip like a microcontroller or FPGA, however, the memory contents are not secure because the memory chip can easily be removed, reprogrammed (reset), and replaced on the circuit board. This method, therefore, does not meet the requirement of being nonresettable.

A Trick from the Past

EPROM is another type of nonvolatile memory that requires no battery. EPROM became popular in the 1970s with the advent of the microprocessor. Fresh from the factory, all EPROM bytes read FFh. Data is stored by programming individual bits from 1 (erased) to 0 (programmed); the programming requires pulses of 12V to 13V. Before loading new data, you must erase the entire memory by exposing the chip to high-intensity ultraviolet light through a window built into the package for that purpose. One-time-programmable (OTP) EPROMs do not have a window and, therefore, cannot be erased. Because of these inconveniencies, the popularity of EPROMs has dwindled in favor of EEPROMs and high-density flash memories, both of which can be operated and programmed at 5V or less. Nonetheless, by combining the one-way programming (1 to 0) and lack of erasability of an OTP EPROM with modern EEPROM technology, you can attain a new feature called EPROM-emulation mode. EPROM-emulation mode is a key technique for realizing nonvolatile, nonresettable counters.

EPROM-Emulation Mode

A common feature of serial-interface EEPROMs is a page-write buffer, which lets you program an entire memory page in a single stroke. On receiving a write command, the system automatically loads the page-write buffer with data from the addressed memory page. For EPROM-emulation mode, the write buffer is implemented as a shift register (**Figure 2**). The incoming new data (D-IN) feeds into an AND gate that combines it with buffer data (S-OUT) before it enters the buffer (S-IN). Thus, the AND gate ensures that a memory bit cannot be changed to 1 after it has been programmed to 0. After a full cycle through all bits of the page, data in the buffer is again aligned with data in the memory page. Now the write cycle can be started, which copies the entire buffer back to the nonvolatile EEPROM.

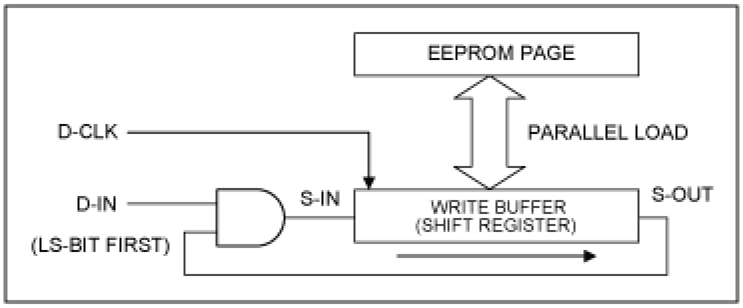

Figure 2. EPROM emulation writes the bitwise logical AND of new data and existing data back to the memory.

Counting in EPROM

Because EPROM bits can be changed only in one direction, a conventional counting code does not work. Instead, you must regard the entire memory array as a single entity of n bits. Initially, none of the n bits are programmed (all are 1). To count an event, you must change one of the unprogrammed bits to 0. You could simply select the next bit to be programmed at random, but the scheme in **Figure 3** is easier to implement. It starts with the least significant bit and continues in sequence until all the bits of that byte are programmed. It then programs the next byte bit by bit, and so on. In EPROM-emulation mode, a memory chip with 1024 bits is good for counting 1024 events.

Count Value	Hex Code	Bit Number							
		8	7	6	5	4	3	2	1
0	FFh	1	1	1	1	1	1	1	1
1	FEh	1	1	1	1	1	1	1	0
2	FCh	1	1	1	1	1	1	0	0
3	F8h	1	1	1	1	1	0	0	0
4	F0h	1	1	1	1	0	0	0	0
5	E0h	1	1	1	0	0	0	0	0
6	C0h	1	1	0	0	0	0	0	0
7	80h	1	0	0	0	0	0	0	0
8	00h	0	0	0	0	0	0	0	0

Figure 3. Counting in EPROM requires a code for which every bit has the same place value.

Chips Supporting EPROM-Emulation Mode

Although the EPROM-emulation mode is easy to implement, only Maxim offers such products at the time of this article. Memory densities of 1Kb (DS2431, DS28CN01, and DS28E01), and 20Kb (DS28EC20) are available. All these chips come with a unique factory-programmed identification number. Except for the DS2431 and DS28EC20, the products with EPROM-emulation mode are secure memories; only a master that generates a message-authentication code based on a device-specific secret can gain write access.

Example Flowchart

As an example, consider a DS2431 memory chip in which memory page 0 is configured to implement a 256-bit counter in EPROM-emulation mode. The 32-byte page can be updated in blocks of eight bytes each, using a 64-bit scratchpad as intermediate storage. The algorithm in **Figure 4** detects the first block with unprogrammed bits, increments the count value, and writes the block back to EEPROM.

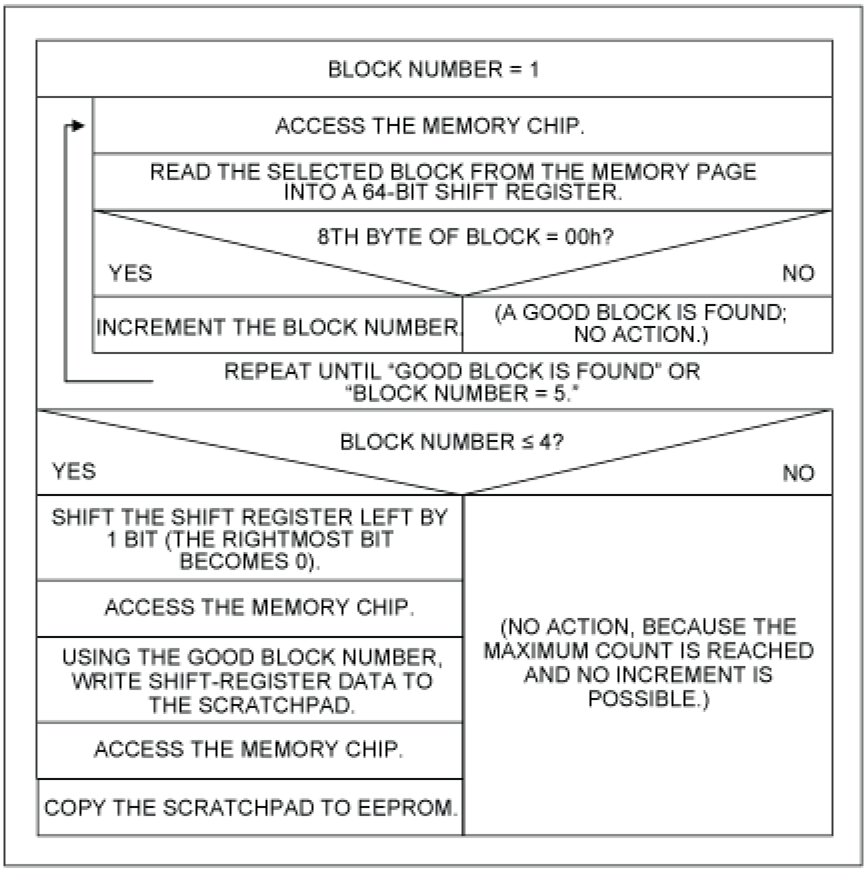
BLOCK NUMBER = 1
ACCESS THE MEMORY CHIP.
READ THE SELECTED BLOCK FROM THE MEMORY PAGE INTO A 64-BIT SHIFT REGISTER.
8TH BYTE OF BLOCK = 00h?
YES
NO
INCREMENT THE BLOCK NUMBER.
(A GOOD BLOCK IS FOUND; NO ACTION.)
REPEAT UNTIL "GOOD BLOCK IS FOUND" OR "BLOCK NUMBER = 5."
BLOCK NUMBER ≤ 4?
YES
NO
SHIFT THE SHIFT REGISTER LEFT BY 1 BIT (THE RIGHTMOST BIT BECOMES 0).
ACCESS THE MEMORY CHIP.
USING THE GOOD BLOCK NUMBER, WRITE SHIFT-REGISTER DATA TO THE SCRATCHPAD.
ACCESS THE MEMORY CHIP.
COPY THE SCRATCHPAD TO EEPROM.
(NO ACTION, BECAUSE THE MAXIMUM COUNT IS REACHED AND NO INCREMENT IS POSSIBLE.)

Figure 4. This algorithm increments the counter in a 32-byte memory page.

Conclusion

EEPROMs that include EPROM-emulation mode are the product of choice for implementing nonvolatile, nonresettable counters. The serial number of the memory chip can be used to detect tampering—that is, replacing the memory chip with one that has a lower count value. To prevent unauthorized incrementing of the counter, one should use a secure memory chip that requires a message-authentication code for write access.

References
Maxim data sheets DS2431, DS28EC20, DS28E01, and DS28CN01.

Related Parts		
DS2431	1024-Bit 1-Wire EEPROM	Free Samples
DS28CN01	1Kbit I²C/SMBus EEPROM with SHA-1 Engine	Free Samples
DS28E01-100	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	Free Samples
DS28EC20	20Kb 1-Wire EEPROM	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 5042: http://www.maximintegrated.com/an5042
APPLICATION NOTE 5042, AN5042, AN 5042, APP5042, Appnote5042, Appnote 5042