WHITEPAPER

**Girija V Chougala,**
***Senior Applications Engineer***
*Cypress Semiconductor Corp.*

# Using Cypress nvSRAM as Program Memory

**Abstract**

This white paper describes the advantages of using Cypress nvSRAM as program memory when compared with other options.

## Introduction

Because of the increasing speed of CPUs, parallel NOR Flash devices often are too slow to "execute in place" (XIP) program code stored in them. However, SRAMs offer access times below 10 ns. As a result, you often must "shadow" code stored in Flash into the RAM; that is, code must be copied from Flash into RAM before execution, so that the CPU may access it at full speed. In compact board designs where PCB form factor is fixed, the two-chip solution becomes difficult to implement because it requires additional board area. The two-chip solution also requires a special bootup subroutine as the system has to configure the memory controllers first, followed by the rest of the bootup sequence, after program code is copied to the SRAM from the Flash memory. Cypress nvSRAM offers a simple and compact solution because it combines high-speed random access with non-volatility in a single chip.

## Program Memory

A few key attributes of the program memory devices are as follows:

### Firmware or Bootup Code

All embedded controllers used in such applications as hard disk drives (HDD), Ethernet controllers, DSL modems, and wireless network devices need to have their firmware or bootup code stored in a nonvolatile memory. One common solution is to have this code stored in an external Flash or EEPROM devices. In most applications where firmware code size is large and the firmware is upgraded periodically, bootup from an external memory is preferred over on-chip Flash memory as a way to store the program firmware. The reason is that manufacturing a Flash memory cell, along with other standard logic cells, is expensive; a Flash cell requires additional masks, making the manufacturing flow complicated and expensive. Therefore, most vendors prefer to support bootup from external memory, as opposed to large on-chip Flash memory capable of meeting density retirements across all applications.

### Execute in Place

An execute in place (XIP) file system enables certain sections of programs to be stored as read-only in an area other than the main system memory. As a result, code can run from the same location at which it is permanently stored. For systems with small amounts of memory, XIP support allows many instances of a large program to co-exist. This is because each instance uses only system RAM for its stack, BSS (Block Started by Symbol), and data segments, while the shared text segment remains in read-only long-term storage. The

BSS segment is that part of the data segment filled with zero valued data initially. The text segment that contains the code can reside in Flash memory and does not need to be copied to the system RAM.

## *Advantages of Cypress nvSRAMs as Program Memory*

Current nonvolatile memory solutions for program memory, such as Flash or EEPROM, are limited by their access speeds. In applications where the controller runs at a speed not less than a few hundreds of megahertz, you cannot run codes from Flash or EEPROM at processor speeds. As a result, the code stored in the nonvolatile memory is first transferred into a RAM so the processor may access it at full speed. When clocked at 20 MHz, 16 Mbits of data takes more than 800 ms for such a transfer. That time puts the entire system into a wait state because the transfer of this boot code delays the bootup of the controller. These wait states can be induced often, whenever the controller needs to run the XIP codes.

The Cypress nvSRAM proves a suitable fit for the application because it is truly nonvolatile and can be accessed exactly like a fast SRAM with access speed up to 20 ns. Unlike other nonvolatile RAMs, the Cypress nvSRAM has a nonvolatile storage element in each memory cell. Therefore, it does not need batteries and provides the most reliable solution with 20 years of data retention. During power up, the nvSRAM automatically recalls the contents from the nonvolatile part into the SRAM within 20 ms, after which it can be accessed at processor speeds. Therefore, the processor can run with no wait states and without special interface circuitry or glue logic.

Modifying the program or bootup code is as simple as writing into an SRAM followed by a Software Store command. Unlike other memories, modifying runtime code does not require costly manual operations or a return of the hardware to the vendor.

Another advantage of using nvSRAM is that it doubles as RAM space. When used to store XIP or bootup code, the controller can use the nvSRAM just like a RAM to store temporary data after executing the code. If the controller wants to run the stored code again, it must be preceded only with a software recall command, which places the code again into the SRAM array and the processor can execute it. When used to store  firmware, if it does not occupy the entire address space, the rest of the memory space can be used by the controller as RAM space for data.

The Cypress nvSRAM also is available with the option of a fully featured real-time clock (RTC) in a reliable, monolithic IC. The watchdog timer is designed to interrupt or reset the processor if the program hangs in a loop and does not respond in a timely manner.

The Cypress nvSRAM offers the following advantages:

- Simple single-chip solution
- Fast and symmetrical write and read access from memory
- No wait states for the processor
- No additional interface circuitry required
- Easy in-circuit code changes
- Can be multiplexed as RAM space
- Optional watchdog timer feature
- Optional RTC and alarms feature

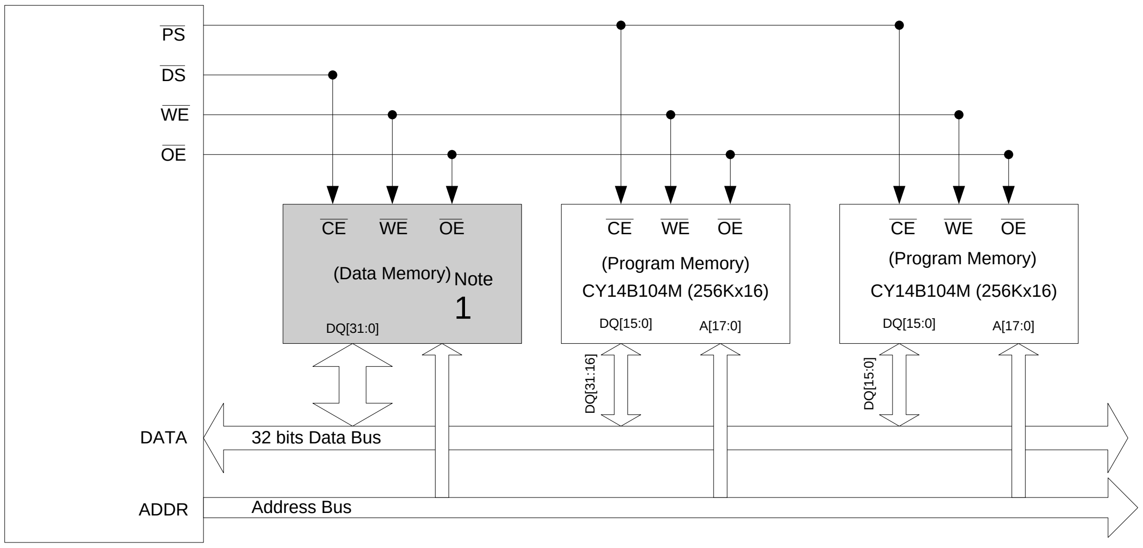
## *Implementing Cypress nvSRAM as Program Memory*

Loading the program code into nvSRAM is similar to writing it into an SRAM. There are two methods of loading the nvSRAM with the boot code or the program code for the processor.

1. Use the controller on board to write the code into the nvSRAM. However, an initial boot code is required for the processor to be up and the memory controller to be active. The nvSRAM has to be preloaded with this boot code. When the processor is up, it can modify or load the entire nvSRAM with the firmware and perform a Software Store. This loads the nvSRAM with the desired program code. Loading the nvSRAM with the code is a onetime activity; during subsequent power cycles, the nvSRAM contains the program code that the processor can access at full speed. This is simple and requires no additional interface compared with Flash or EEPROMs, which are programmed, generally using JTAG pins. Use the second method if it is not feasible to preload the nvSRAM with the initial boot code.

2. When a blank nvSRAM is used on the board (with no preloaded initial boot code), all the data, address, and control lines of the nvSRAM must be brought on to a header on the board. Then, an appropriate jig is designed that writes into the nvSRAM through this connector. This is similar to programming the EEPROM or Flash with JTAG pins.

Figure 1. Interfacing nvSRAM as Program Memory for a 32-Bit Controller



**Note 1** SRAM data memory may be optional, and it can use nvSRAM instead.

Regardless of the way in which nvSRAM is written, you must perform the following actions to load the program code into the nvSRAM:

- Disable AutoStore. nvSRAM comes from the factory with AutoStore enabled. During power down, the contents of the SRAM array are stored into the nonvolatile parts so that it is recalled during the next power up. Because the program code must be permanently stored and recalled each time during power up, AutoStore must be disabled or the program code in the nonvolatile part could get corrupted. To disable AutoStore, the controller must perform sequential $\overline{CE}$ or $\overline{OE}$ controlled reads from six specific address locations. For more details on disabling AutoStore, refer to datasheets on the Cypress website.
- Write the code into the nvSRAM. This is similar to writing into an SRAM.

  Perform a Software Store. When the code is written into the nvSRAM, it must be stored into the nonvolatile part. This is accomplished using the Software Store command. To perform Software Store, the controller must perform sequential $\overline{CE}$ or $\overline{OE}$ controlled reads from six specific address locations. For more details on performing Software Store, refer to datasheets on the Cypress website.

- When the code is stored in the nvSRAM, during each power up the SRAM array holds the code ready for the processor to access it in 20 ns or faster. You can modify code easily by just writing into the nvSRAM followed by a Software Store.

## *Summary*

The Cypress nvSRAM offers a simple and efficient solution for program memory. By replacing the combination of EEPROM/Flash and SRAM used in most embedded applications, the nvSRAM saves board space, cost, and power.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com