

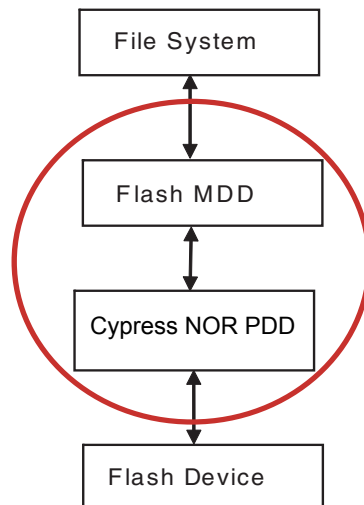
## Designing with the Cypress NOR PDD for Windows® Embedded CE 6.0R2 and Later

AN98545 describes the features and parameters of the Cypress NOR PDD to unlock the maximum potential for Cypress NOR devices in Windows Embedded platforms.

### 1 Introduction

The Microsoft® Flash MDD (Model Device Driver) presents a block device interface to file systems in Windows CE. The Cypress NOR Flash PDD (Physical Device Driver) is the low-level driver that enables the flash MDD to operate on Cypress NOR devices. The Cypress NOR PDD can also be integrated into the OAL (OEM Adaptation Layer) in order to program raw data to the flash memory with no MDD metadata. This application note describes the features and parameters of the Cypress NOR PDD to unlock the maximum potential for Cypress NOR devices in Windows Embedded platforms.

Figure 1. Block Driver



### 2 Features

- Supports a single x16 device or two devices interleaved to form a 32-bit bus (2x16)
- Automatically configures based on device characteristics from CFI information
- Optionally reserves code space at the start and/or end of the flash array
- Supports various file system sector sizes
- Modular code allows for simple platform customizing

### 3 Customizing

The Cypress NOR PDD can be customized to meet the needs of different systems. Separate libraries are linked together to create the Cypress NOR PDD. These libraries include the core PDD logic, the platform interface, and flash parameters. Any one of these modules can be updated without having to rebuild the other modules. Initially, each library must be built before linking the DLL in the project build. Using Platform Builder, the component

libraries can be compiled from the Solution Explorer window by browsing to the applicable component under PUBLIC\SPN\_NOR.

```

+---Catalog
+---CESYSGEN
+---NORPARMS
|   +---oal
|   +---registry
+---NORPCI
+---oak
|   +---FILES
|   +---lib
|   +---target
+---PLATFORM
|   +---arm9
|   +---default
+---SPNNORD
  
```

The PLATFORM tree contains multiple options for code that is specific to the platform. These modules contain the low-level routines to initialize and interface with flash for a given platform. The default code (defaultplat.cpp) is a starting point for writing a new platform interface. The ARM9 example (arm9plat.cpp) includes processor-specific initialization.

The PLATFORM options also appear as radio button selections in the catalog. Note that changing this catalog selection does not force a rebuild of the flash driver DLL (spnnord.dll). Rebuild any component library and then build the Windows CE project. This will cause the correct libraries to be linked.

The NORPARMS tree contains two options for providing flash and system parameters to the core Cypress NOR PDD code. The registry example (registryparms.cpp) queries the system registry for most of the parameter values. This code is built by default. The OAL example (oalparms.cpp) shows how to specify these parameters statically. If the Cypress NOR PDD is integrated into the OAL, the registry is not available. Parameters for a given platform must be specified with either approach.

Table 1 shows the Cypress NOR PDD parameters and the associated registry entries.

Table 1. NOR PDD Parameters

Parameter	Registry Entry
SPN_PARAM_BASE_ADDR	"MemBase"
SPN_PARAM_SECTORSIZE	"FileSystemSectorSize"
SPN_PARAM_CODESIZE_START	"ReservedSizeAtStart"
SPN_PARAM_CODESIZE_END	"ReservedSizeAtEnd"
SPN_PARAM_BUS_WIDTH_BITS	"BusWidthBits"
SPN_PARAM_DIRECT_MAP_MODE	

## 4 Registry Entries

Several registry entries are used to configure the Cypress NOR PDD.

Table 2. Registry Entries

Registry Entry	Optional	Description
MemBase	No	Physical start of the flash array
MemLen	No	This value is not used directly by the PDD
ReservedSizeAtStart	Yes	Reserved bytes from start of flash
ReservesSizeAtEnd	Yes	Reserved bytes from end of flash
FileSystemSectorSize	Yes	Minimum sector size reported to flash MDD
BusWidthBits	No	10h for 16 bits, or 20h for 32 bits (2x16)

The "ReservedSizeAtStart" and "ReservedSizeAtEnd" registry entry values are rounded up to the next erase block size.

The "FileSystemSectorSize" must match the sector size for the file system used. This value must not be less than the write buffer size, but may be an even multiple of the write buffer size. For two interleaved devices (2x16), this value must be at least double the write buffer size of a single device. Two interleaved devices with 512B write buffers form a combined write buffer size of 1 kB. File system sectors sizes are normally 512B. Since the combined write buffer size is larger than the typical file system size, a larger file system sector size is needed. 1024 (dword:400) is a valid value.

Setting the "BusWidthBits" entry to "10" indicates a single flash device with a 16 bit data bus. The value "20" indicates two interleaved devices, each with a 16 bit data bus, resulting in a 32 bit data bus.

The "MemLen" registry key must exist for the PDD to successfully call MmMapIoSpace, which maps the physical flash address to a virtual address space the PDD can access. The flash size is read from CFI. The "MemLen" value is ignored

The remaining registry entries are documented at [www.msdn.com](http://www.msdn.com). Search for "block driver registry" and "MemBase MemLen".

Below is an example of registry settings for a Cypress NOR device, without reserving space at the beginning, nor the end:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\SPNNORD]
"Dll"="flashmdd.dll"
"FlashPddDll"="spnnord.dll"
"Order"=dword:0
"Prefix"="DSK"
"Ioct1"=dword:4
"Profile"="SpansionNOR"
"IClass"="{A4E7EDDA-E575-4252-9D6B-4195D48BB865}"
"MemBase"=dword:18000000
"MemLen"=dword:00000000
"FileSystemSectorSize"=dword:200
"ReservedSizeAtStart"=dword:00000000
"ReservedSizeAtEnd"=dword:00000000
"BusWidthBits"=dword:10

; Override names in default profile
[HKEY_LOCAL_MACHINE\System\StorageManager\Profiles\SpansionNOR]
"Name"="MSFLASH for Spansion NOR"
"Folder"="Spansion NOR Flash"
"PartitionDriver"="flashpart.dll"
"DefaultFileSystem"="FATFS"

[HKEY_LOCAL_MACHINE\System\StorageManager\AutoLoad\SpansionNOR]
"DriverPath"="Drivers\BuiltIn\SPNNORD"
; LoadFlags 0x01 == load synchronously
"LoadFlags"=dword:1
"Order"=dword:0
```

The following optional registry entries can be used to control how the Storage Manager handles the flash disk at boot time.

Table 3. Optional Registry Entries

Registry Entry	Default	Description
AutoMount	1	Automatically attempt to mount, if set to 1
AutoPart	0	Automatically create largest possible partition, if set to 1
AutoFormat	0	Automatically format the store when unformatted, if set to 1

In order to override the Storage Manager default value for a given entry, add the entry to the profile key in spn\_nor.reg, for example:

```
[HKEY_LOCAL_MACHINE\System\StorageManager\Profiles\SpansionNOR]
"Name"="MSFLASH for Spansion NOR"
```

```

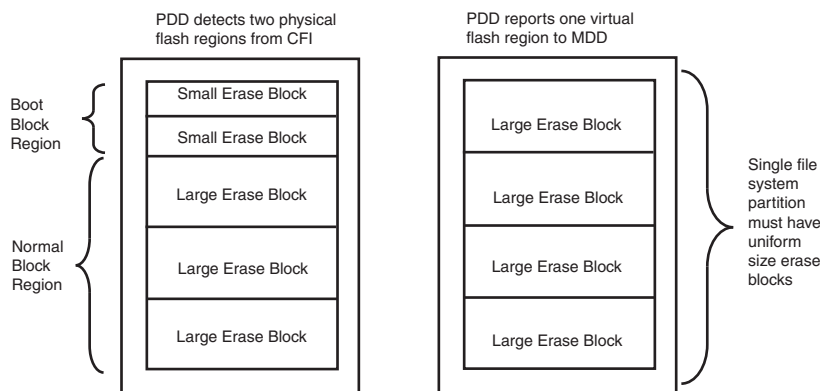
"Folder"="Spansion NOR Flash"
"PartitionDriver"="flashpart.dll"
"DefaultFileSystem"="FATFS"
"AutoPart"=dword:1
"AutoFormat"=dword:1
  
```

One way to manually format the drive and create a partition is using the Storage Manager Control Panel applet. This approach is useful during integration and development with the flash block driver.

## 5 Flash Regions

The PDD reports to the Flash MDD the number of flash regions, where each region has a uniform erase block size. The Flash MDD supports one or more partitions per flash region, but not a single partition spanning multiple flash regions. The Cypress NOR PDD presents a single flash region with uniform size erase blocks. Since a single flash region is reported, a partition can span the entire Cypress NOR device.

Figure 2. Erase Block Regions



By default, boot blocks are grouped into the normal block size for the device, and included in the single flash region reported. However, if the boot blocks are not intended for file system storage, they can be hidden from the Flash MDD using the `ReservedSizeAtStart` and/or `ReservedSizeAtEnd` registry entries.

## 6 Reserving Space

In order to hide part of the flash array from the Flash MDD, registry entries can be used to reserve space at the start and/or end of the flash. Care must be taken to ensure there is no conflict between software accessing the different areas of flash. The Flash MDD, along with the partition manager, will control access to the area of flash that is not reserved. File system storage will require program and erase operations on the flash device. During these operations, the flash device will display status. Information stored in the reserved area may become temporarily unavailable.

Since the device size is detected from CFI, a two chip enable device may have only half of the total flash array available at the given flash base address. The `"ReservedSizeAtEnd"` entry should be used to reserve half of the flash array, unless the memory space for both chip enables are available and contiguous.

## 7 Code Execution

When a single-bank flash is performing program or erase operations, status is displayed across the entire device. For this type of device, the reserved area can only be used for software that does not execute while the file system is running, like startup code. Likewise, the reserved area can store the Windows CE OS image (nk.bin) if the OS image is entirely shadowed to RAM. In order to demand page the OS image from flash, a BinFS partition can be created in the flash area managed by the Flash MDD.

When a multi-bank flash is performing program or erase operations, status is only displayed in one bank at a time. If the reserved area aligns with bank boundaries, then the reserved area can be used for read-only access, even while a file system is actively using the banks managed by the Flash MDD. This allows code to eXecute In Place

(XIP), or run directly out of flash during normal platform operation. Refer to your flash device data sheet to determine the number of banks.

## 8 Example Parameters

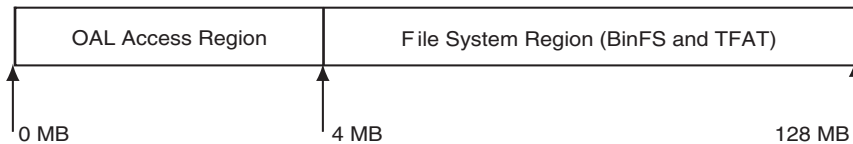
This example platform contains two S29GL512P devices in a 2x16 interleave. The combined write buffer size is 128B, so the typical sector size of 512B can be used. The first 4 MB hold a bootloader.

The remainder of the flash array is managed by the flash partition manager, with one partition for storing and demand paging the OS image using BinFS, and one partition for file storage with TFAT.

The following run-time registry entries (spn\_nor.reg) configure the Cypress NOR PDD to operate with the Flash MDD for both file system partitions. The first 4 MB are hidden from the Flash MDD.

```
"MemBase"=dword:80000000
"MemLen"=dword:00000000
"FileSystemSectorSize"=dword:200
"ReservedSizeAtStart"=dword:00400000
"ReservedSizeAtEnd"=dword:00000000
"BusWidthBits"=dword:20
```

Figure 3. Memory Map



The following example parameter values (oalparms.cpp) configure the Cypress NOR PDD in the OAL for execution at boot time. The first 4 MB of the flash array are accessible for program and update of bootloader code. This area is programmed with direct map mode for raw reads and writes (no MDD meta data). The remaining 124 MB of the flash array are reserved from the perspective of the OAL. This example is for the same platform described above.

Table 4. Boot Time Parameters

Parameter	Value
SPN_PARM_BASE_ADDR	0x80000000
SPN_PARM_SECTORSIZE	0x200
SPN_PARM_CODESIZE_START	0
SPN_PARM_CODESIZE_END	0x7C00000
SPN_PARM_BUS_WIDTH_BITS	0x20
SPN_PARM_DIRECT_MAP_MODE	TRUE

## 9 Conclusion

Please consult your flash device data sheet to ensure your platform adheres to the device specifications. To obtain a copy of the NOR PDD, or for questions about using Cypress NOR devices, please contact Cypress via the "Support" link on our website: <http://www.cypress.com>.

## Document History Page

Document Title: AN98545 - Designing with the Cypress NOR PDD for Windows® Embedded CE 6.0R2 and Later Document Number: 001-98545				
Rev.	ECN No.	Orig. of Change	Submission Date	Description of Change
**	—	GJSW	06/29/2010	Initial version
*A	—	GJSW	03/05/2012	Customizing section: Added directory tree  Registry Entries section: Added table: Optional Registry Entries Updated text  Example Parameters section: Added Memory Map figure
*B	4981091	MSWI	10/27/2015	Updated in Cypress template
*C	5868884	AESATMP8	08/31/2017	Updated logo and Copyright.

## Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at [Cypress Locations](#).

### Products

ARM® Cortex® Microcontrollers	<a href="http://cypress.com/arm">cypress.com/arm</a>
Automotive	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
Clocks & Buffers	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
Interface	<a href="http://cypress.com/interface">cypress.com/interface</a>
Internet of Things	<a href="http://cypress.com/iot">cypress.com/iot</a>
Memory	<a href="http://cypress.com/memory">cypress.com/memory</a>
Microcontrollers	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
Power Management ICs	<a href="http://cypress.com/pmic">cypress.com/pmic</a>
Touch Sensing	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB Controllers	<a href="http://cypress.com/usb">cypress.com/usb</a>
Wireless Connectivity	<a href="http://cypress.com/wireless">cypress.com/wireless</a>

### PSoC® Solutions

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6](#)

### Cypress Developer Community

[Forums](#) | [WICED IOT Forums](#) | [Projects](#) | [Video](#) | [Blogs](#) | [Training](#) | [Components](#)

### Technical Support

[cypress.com/support](http://cypress.com/support)

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



Cypress Semiconductor  
 198 Champion Court  
 San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2010-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1s) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.