Nintendo DS Game Card Manual

Version 1.01

The contents in this document are highly confidential and should be handled accordingly.

NTR-06-0375-001-A1
Released: November 9, 2005

Confidential

Table of Contents

1 Introduction 5
1.1 Block Diagram 5
2 Structure 6
2.1 ROM 6
2.1.1 ROM Types 6
2.1.2 ROM Registration Data 7
2.1.3 Memory Map 10
2.2 Backup Memory 12
2.2.1 Lineup 12

Revision History

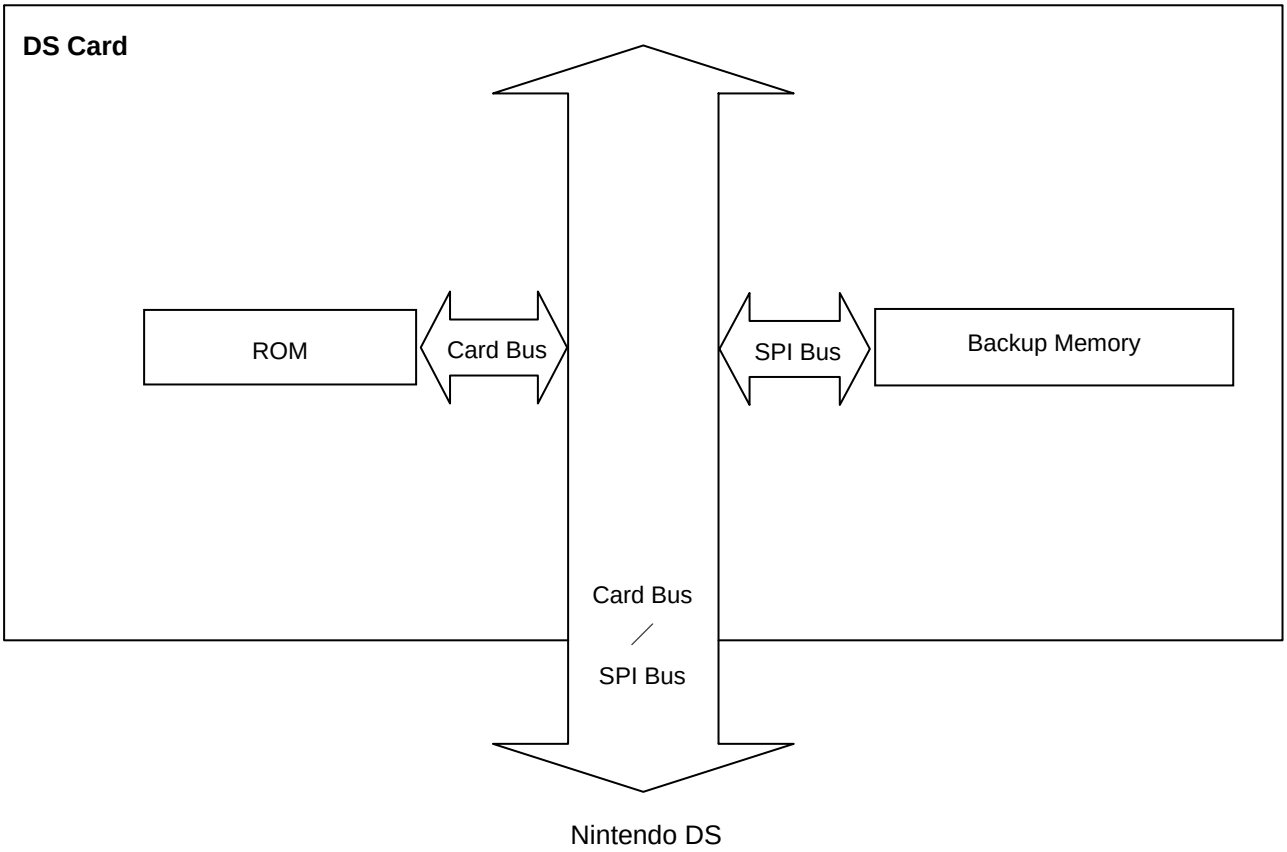
Version	Revision Date	Description
1.01	10/17/2005	• In the “Lineup” section on page 10, changed the EEPROM data storage period from 40 years to 10 years. • In the “Lineup” section on page 10, deleted mention of a scheduled release of 512-kilobit EEPROM in the second half of 2005. It is available now. • Added 4-megabit FLASH memory in the “Lineup” section on page 10. • In the “Lineup” section on page 10, deleted mention of a scheduled release of FRAM in the second half of 2005. It is available now (consultation required regarding delivery time). Changed the number of guaranteed rewrites.
1.00	07/01/2005	• Initial release

1 Introduction

1.1 Block Diagram

The block diagram in Figure 1-1 represents the Nintendo DS proprietary Game Card (DS Card).

Figure 1-1 DS Card Block Diagram



2 Structure

The DS Card consists of ROM and backup memory.

2.1 ROM

2.1.1 ROM Types

ROM is divided into mask ROM and one-time PROM. The `makerom` settings file (RSF file) specifies which type of ROM image to create.

- `makerom` before NITRO-SDK Version 2.1 always creates mask ROM.
- Nintendo selects which ROM type to use in production.

The ROM type characteristics are shown in Table 2-1.

Table 2-1 ROM Type Characteristics

	Mask ROM	One-time PROM
Transfer Rate	5.99 MB/sec	1.52 MB/sec
Capacity	64 megabits, 128 megabits, 256 megabits, 512 megabits	128 megabits, 256 megabits, 512 megabits, 1 gigabit
Page Size	512 bytes (= 4 kilobits)	
Feature	Fast transfer rate	Short delivery times for repeat deliveries

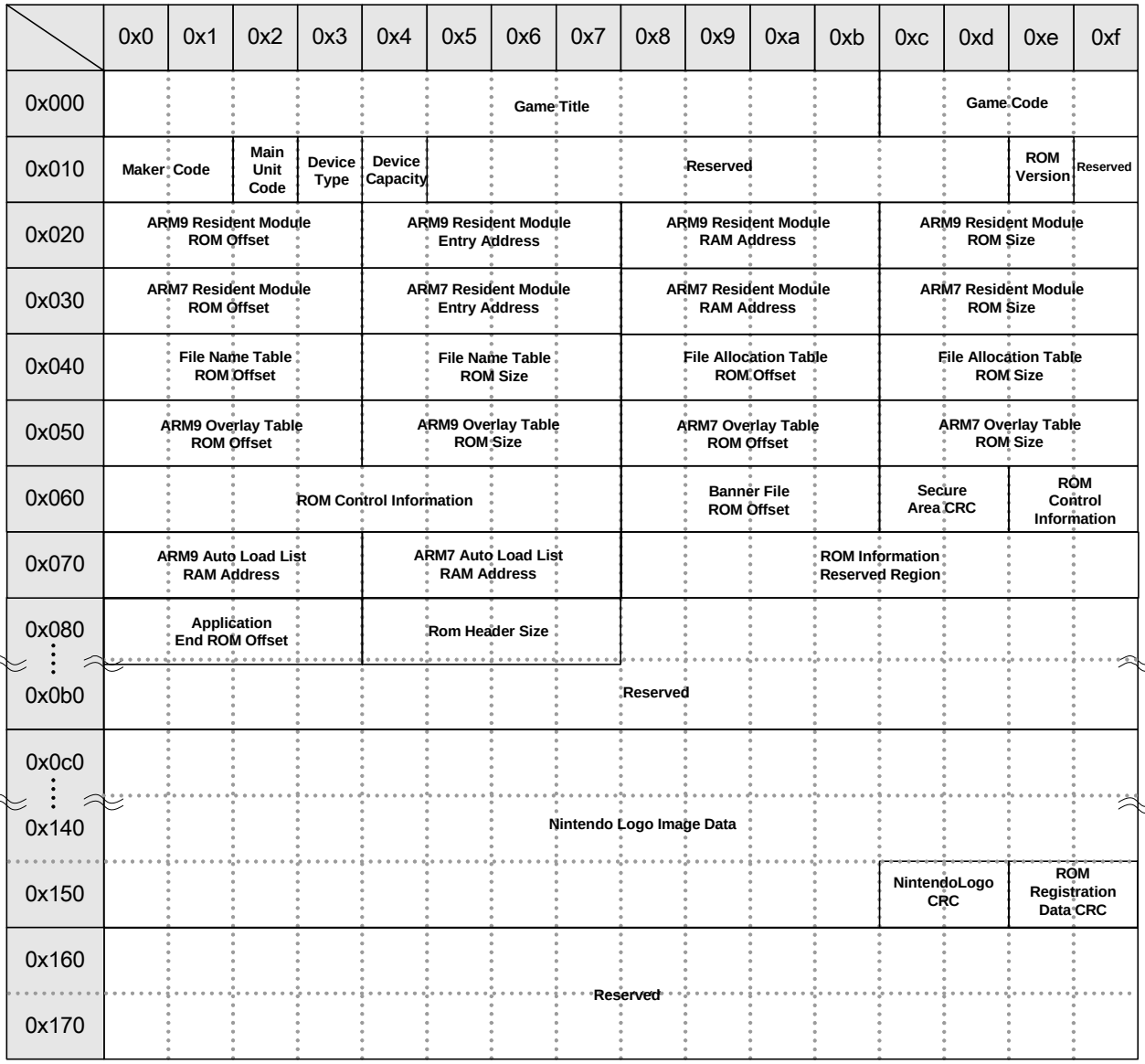
<Regarding the transfer rate>

- The values are theoretical values that exclude overhead.
  (The transfer rate depends on the application program, but in general, the transfer times in an actual program will be less than the transfer rates shown above.)
- If the RSF file specifies one-time PROM, production can use either one-time PROM or mask ROM; however, the one-time PROM transfer rate is used.

2.1.2 ROM Registration Data

ROM used on DS Game Cards must contain information about the game software written on the DS Card's ROM. This information about the game software is called *ROM registration data*. ROM registration data is generated with `makerom`. For details about specifications in the ROM specification file, see the `makerom` reference. Details of the ROM registration data are shown in Figure 2-1.

Figure 2-1 ROM Registration Data



Registration data types are shown in Table 2-2.

Table 2-2 ROM Registration Data Setting Methods

Type	Registration Data
FIX	Uses values set in the ROM header template as they are.
RSF	The descriptions in the ROM specification file are reflected in `makerom`. If there is no specification, this works the same as FIX.
GEN	The values generated by `makerom` are written.

Registration Data Details:

- Game Title <12 bytes> (RSF)
  This is the name of the software title. It is registered using ASCII codes from 0x20 to 0x5f.
  Use 0x20 for spaces and 0x00 for unused portions.
- Game Code <4 bytes> (FIX)
  This registers the 4-digit code that is set for each game title.
- Maker Code <2 bytes> (RSF)
  This registers the 2-digit licensee code that is assigned by Nintendo.
- Main Unit Code <1 byte> (FIX)
  This registers the identifying code of the main unit that the software is made for.
  Currently, this is fixed to 0x00.
- Device Type <1 byte> (FIX)
  This registers the type of device that is mounted in the Game Card.
- Device Capacity <1 byte> (RSF)
  This registers the ROM capacity.

Table 2-3 shows the correspondence between the set value and ROM capacity.

Table 2-3 Device Capacity Set Value and Corresponding ROM Capacity

Set Value	ROM Capacity	Set Value	ROM Capacity
0x00	1Mbit	**0x08**	256Mbit
0x01	2Mbit	**0x09**	512Mbit
0x02	4Mbit	**0x0a**	1Gbit
0x03	8Mbit	**0x0b**	2Gbit
0x04	16Mbit	**0x0c**	4Gbit
0x05	32Mbit	**0x0d**	8Gbit
0x06	64Mbit	**0x0e**	16Gbit
0x07	128Mbit	**0x0f**	32Gbit

- ROM Version <1 byte> (RSF)
  This registers the ROM version number.

- ARM9/ARM7 Static Module ROM Offset <4 bytes> (GEN)
  This stores the ROM transfer source address of each processor's static module.
- ARM9/ARM7 Static Module Entry Address <4 bytes> (GEN)
  This stores the execution start address of each processor's static module.
- ARM9/ARM7 Static Module RAM Address <4 bytes> (GEN)
  This stores the RAM transfer destination address of each processor's static module.
- ARM9/ARM7 Static Module ROM Size <4 bytes> (GEN)
  This stores the size of each processor's static module.
- File Name Table/File Allocation Table ROM Offset <4 bytes> (GEN)
  This stores the ROM address of each file-related table.
- File Name Table/File Allocation Table ROM Size <4 bytes> (GEN)
  This stores the size of each file-related table.
- ARM9/ARM7 Overlay Table ROM Offset <4 bytes> (GEN)
  This stores the ROM addresses of the overlay tables used for each processor.
- ARM9/ARM7 Overlay Table ROM Size <4 bytes> (GEN)
  This stores the sizes of the overlay tables used for each processor.
- ROM Control Information <10 bytes> (RSF)
  This stores parameters that control the ROM.
  When using the Debugger, some parameters are overwritten after they are loaded.
- Banner File ROM Offset <4 bytes> (RSF)
  This stores the ROM offset to the banner file that summarizes the identifying image information for the application and that is shown on the menu when NITRO starts up.
  When there is no banner file, this is zero.
  Banner files are created with `makebanner` and integrated into the application ROM image with `makerom`.
- Secure Region CRC <2 bytes> (GEN)
  This stores CRC-16 for the ROM secure region.
  Calculations with this algorithm use an initial value of `0xffff`.
- ARM9/ARM7 Auto Load List RAM Address <4 bytes> (GEN)
  This stores, at startup, the address of the list used to transfer a portion of the static module to an execution address that is mapped in another region.
- ROM Information Reserved Region <8 bytes> (FIX)
  This region is reserved to store ROM-related information.
  Must be set to `0x00`.
- Application End ROM Offset <4 bytes> (GEN)
  This stores the end ROM offset of the region used by the application inside the ROM image.

- ROM Header Size <4 bytes> (GEN)
  This normally stores the size of the ROM header template. In Single-Card Play, this stores the size of the ROM header template included in the executable image on the child device.
- Nintendo Logo Image Data <156 bytes> (FIX)
  This stores the image data used to display the logo when the main unit starts.
  The contents are verified before the application starts.
- Nintendo Logo CRC <2 bytes> (FIX)
  This is the CRC-16 for the Nintendo Logo Image Data.
  This is a fixed value obtained by using the algorithm with an initial value of `0xffff`.
- ROM Registration Data CRC <2 bytes> (GEN)
  This stores the CRC-16 for the area up to the Nintendo Logo CRC.
  Calculations with this algorithm use an initial value of `0xffff`.
- Reserved Region (FIX)
  Must be set to `0x00`.

2.1.3 Memory Map

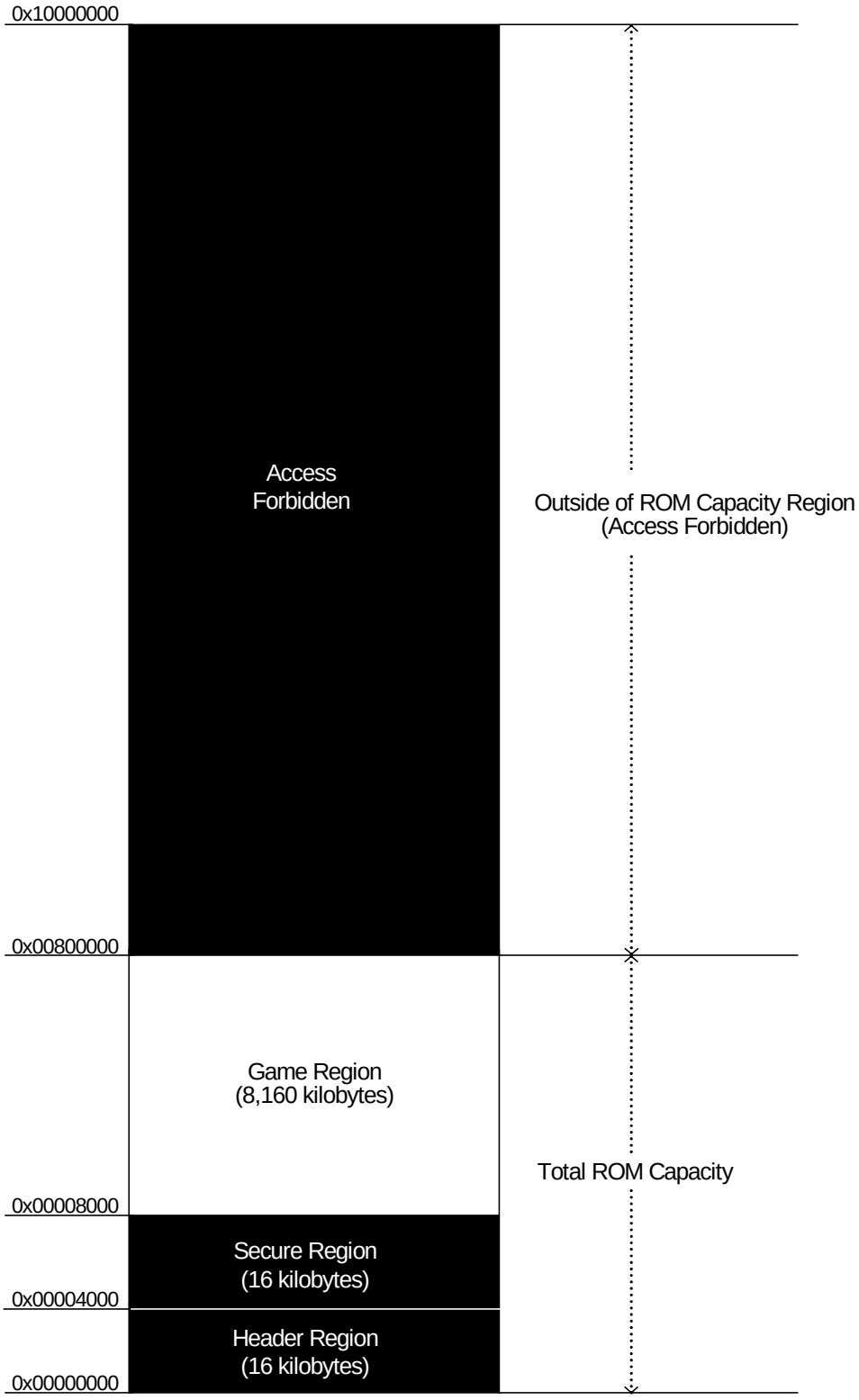
The memory map is the same for both mask ROM and one-time PROM devices.

- Figure 2-2 shows an example of a memory map for a 64-megabit DS Card.
- For cards other than the 64-megabit card, the address value of the upper limit of the game region is different.

The game region capacity is as follows.

Game Region Capacity = *ROM total capacity* − 32 kilobytes

Figure 2-2 DS Card ROM Region Memory Map (64-megabit Card)



2.2 Backup Memory

2.2.1 Lineup

Table 2-4 shows the lineup of backup memory devices.

Table 2-4 Backup Memory Device Lineup

Memory Type	Capacity	Page Size	No. of Guaranteed Rewrites	Required Time for Rewrite (1 byte - 1 page)	Data Storage Period
EEPROM	4 kilobits	16 bytes	1 million	5 ms	10 years
	64 kilobits	32 bytes			
	512 kilobits	128 bytes			
FLASH Memory	2 megabits	256 bytes	100,000 (10,000)[1]	25 ms[2] 300 ms[3]	20 years
	4 megabits[4]		100,000	25ms	
FRAM (planned)[4]	256 kilobits	-	10 billion[5] (rewrite + read)	0 ms	10 years

[1] Ten thousand (10,000) is the guaranteed number of rewrites for rewrites that require 25 ms.

[2] This time is guaranteed for devices with less than 10,000 total rewrites.

[3] This time is guaranteed for devices with more than 10,000 and less than 100,000 total rewrites.

[4] Please contact support@noa.com early in your development, if you wish to use this component.

[5] In the case of FRAM, the guaranteed value refers to the total number of reads and rewrites, not just rewrites.

<About rewrite units>

- EEPROM internally maintains a buffer of one page, and rewrites are executed in chunks from one byte to one page in size.
- FLASH memory internally maintains a buffer of one page, and rewrites are executed in units of one page.