Module Introduction

Purpose

- This training module provides an overview of the on-chip Flash memory that H8S series microcontrollers provide.

Objectives

- Learn about Renesas’ Flash memory technology.
- Understand the operating modes of the on-chip Flash memory and how that memory can be programmed.
- Discover ways to help ensure the success of embedded systems that use H8S series MCUs with on-chip Flash memory.

Content

- 22 pages
- 4 questions

Learning Time

- 45 minutes
H8 MCUs with On-chip Flash

1st Generation (0.8µm)
- H8/300
- H8/500
- 3642A 3643 3644
- 3334 3337 3434 3437
- 538 539
- 3048
- 7044 7045 7017

2nd Generation (0.6µm)
- H8/300
- 3854 3857
- 3337S 3437S
- 3067 3062 3039
- 2345 2357 2128 2132 2134 2138 2142 2144 2148
- 2134A 2138A 2144A 2148A 2214 2238B 2239 2268 2314 2315 2318 2319
- 2328 2329 2398 2612 2623 2626 2633 2636 2644 2646 2648 2676
- 7055S

3rd Generation (0.35µm)
- H8/300H (Tiny)
- 36014 3668 3684 3694 3672 3670 3664N 3664
- 3068 3064B 3062B 3052B 3048B 3022
- 2134A 2138A 2144A 2148A 2214 2238B 2239 2268 2314 2315 2318 2319
- 2328 2329 2398 2612 2623 2626 2633 2636 2644 2646 2648 2676

4th Generation (0.18µm)
- H8/300L
- 38024 3804
- 36014 3687 3684 3694 3672 3670 3664N 3664
- 3069

SH-C
- 7055
On-chip Flash

- Wide range of Flash sizes
  - (32KBytes to 1MByte)
- High-speed programming
  - Programming time: 80µs (typ.)/Byte
  - Erase time: 100ms (typ.)
- 10,000 Flash re-write cycles
  - Most devices: 100 re-write cycles (min.)
- User-friendly operating modes
- Can program 128Bytes at a time
- 10 years of data retention
- Wide temperature range
- Two protection modes
Flash Memory Control Circuits*

- The Flash Memory Control Register 1 (FLMCR1) puts the Flash memory into program mode, program-verify mode, erase mode, or erase-verify mode.
- The Flash Memory Control Register 2 (FLMCR2) displays the state of Flash memory programming and erasing operations. It is a read-only register that should not be written to.
- The Erase Block Register 1 (EBR1) specifies the Flash memory erase area block.
- The Erase Block Register 2 (EBR2) specifies the Flash memory erase area block.
- The RAM Emulation Register (RAMER) is available in selected H8S Flash MCUs. It specifies the area of Flash memory to be overlapped with part of RAM when emulating real-time Flash memory programming.

Legend

- FLMCR1: Flash memory control register 1
- FLMCR2: Flash memory control register 2
- EBR1: Erase block register 1
- EBR2: Erase block register 2
- RAMER: RAM emulation register

* Circuit shown is for H8S Flash MCUs built with 0.35μm process technology.
Block Configuration of Flash

Address H'00000

4 kbytes × 8

256 kbytes

Address H'00000

1 kbyte × 4

128 kbytes

Address H'1FFFF

28 kbytes

Address H'00000

32 kbytes

64 kbytes

Address H'3FFFF

64 kbytes

Address H'1FFFF

8 kbytes

Address H'5FFFF

8 kbytes

384 bytes

Address H'00000

32 kbytes

64 kbytes

Address H'3FFFF

64 kbytes

Address H'1FFFF

8 kbytes

Address H'5FFFF

32 kbytes

64 kbytes

Address H'00000

64 kbytes

Address H'3FFFF

64 kbytes

Address H'1FFFF

64 kbytes

Address H'5FFFF

64 kbytes

© 2008, Renesas Technology America, Inc., All Rights Reserved
Which of the following statements apply to the on-chip Flash memory available in many H8S MCU devices? Select all that apply and then click Done.

- You can program 128Bytes at a time
- You can rewrite the Flash memory up to 10,000 times
- H8S MCUs have a Flash memory that will retain data for one year
- The Flash memory operates over a wide temperature range
- The size of Flash ranges from 32KBytes to 1MByte
Memory Maps (H8S/2633)

- **Modes 4 & 5**
  - H’000000: External Address Space
  - H’040000: External Address Space
  - H’FFB000: On-chip RAM
  - H’FFEFC0: External Area
  - H’FFFF80: Internal I/O Regs
  - H’FFFF40: External Area
  - H’FFFF60: Internal I/O Regs
  - H’FFFFC0: On-chip RAM
  - H’FFFFFF: On-chip RAM

- **Mode 6**
  - H’000000: On-chip RAM
  - H’040000: External Address Space
  - H’FFB000: On-chip RAM
  - H’FFEFC0: External Area
  - H’FFFF80: Internal I/O Regs
  - H’FFFF40: External Area
  - H’FFFF60: Internal I/O Regs
  - H’FFFFC0: On-chip RAM
  - H’FFFFFF: On-chip RAM

- **Mode 7**
  - H’000000: On-chip RAM
  - H’03FFFF: On-chip RAM
  - H’FFB000: External Address Space
  - H’FFEFC0: External Area
  - H’FFEFEF: Internal I/O Regs
  - H’FFFF3F: External Area
  - H’FFFF60: Internal I/O Regs
  - H’FFFFFC0: On-chip RAM
  - H’FFFFFF: On-chip RAM
Operating Modes of On-chip Flash

- Only make a transition between User mode and User Program mode when the CPU isn’t accessing the Flash memory.

Notes:
1. RAM emulation in selected MCUs
2. Switch to Programmer mode by using a dedicated PROM programmer.
Flash Memory Programming Methods

On-board Programming

- Boot mode
  - Host PC with FDT Software
  - Yokogawa (Push Button)

- User Program mode
  - (Code download via USB, I²C, Ethernet, SCI, or Parallel Port)

Off-board Programming

- Programmer (PROM writer) mode
  - Minato
  - Data I/O

- BP Micro
On-board Programming - Boot Mode

- **Host PC and FDT based programming**
  - Use FDT software to program on-chip Flash memory.
  - Program only one device at a time.
  - Erasing and writing into the Flash is controlled by the FDT software.
  - Application code from host PC is transferred via SCI port (or USB port).

- **Third-party system based programming**
  - A third-party system, such as the one from Yokogawa, is used to perform on-board, single-device programming.
  - There is push-button operation.
  - Multiple third-party systems may be set up in parallel to program several devices the same time.
  - **This type of programming** is suitable for low-volume production.
Four Stages of Boot Mode

1. Preparation
   - Host
     - Programming control program
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - Application program (old version)

2. Transfer of Programming Control Program
   - Host
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - Application program (old version)
     - Programming control program

3. Flash Initialization
   - Host
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - RAM
     - Flash memory prewrite-erase
     - Programming control program

4. Writing New Program
   - Host
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - RAM
     - Boot program area
     - Programming control program
Let's review the four stages of Boot mode. Complete each sentence by dragging the letters on the left to their corresponding descriptions on the right. Click Done when you are finished.

**Preparation stage**

In the ________, the Boot program in this LSI (originally built into the chip) is started.

**Programming control program transfer stage**

In the ________, the erase program in the Boot program area (in RAM) is executed, and the Flash memory is initialized (to H'FF).

**Flash memory initialization stage**

In the ________, the programming control program transferred from the host to RAM is executed.

**Writing new application code stage**

In the ________, the old program version or data remains written in the MCU's on-chip Flash memory.
User Program Mode

- The User Program mode for on-board programming is used to reprogram only selected portions of the on-chip Flash memory; it is not used to program all of the Flash memory cells.

- Erasing and writing the Flash is done by executing a Flash control program you have written.

- You store the Flash control program in the MCU’s on-chip Flash, along with the application code you write, using the User Boot mode or the Program mode.

- You execute your control program and download new code or data via your choice of input port: USB, I²C, Ethernet, parallel, or SCI - whichever you prefer.
User Program Mode Operation

1. Preparation
   - Host
     - Programming/erase control program
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - Transfer program
     - Application program (old version)

2. Transfer of Programming Control Program
   - Host
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - Transfer program
     - Programming/erase control program
     - Application program (old version)

3. Flash Initialization
   - Host
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - Transfer program
     - Flash memory erase

4. Writing New Program
   - Host
     - New application program
   - Chip
     - Boot program
     - Flash memory
     - Transfer program
     - Programming/erase control program
     - New application program

Program execution
## Boot Mode vs. User Program Mode

<table>
<thead>
<tr>
<th></th>
<th>Boot Mode</th>
<th>User Program Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entire memory erase</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Block erase</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Programming control program*</td>
<td>Program/program-verify</td>
<td>Erase/erase-verify/program/program-verify/emulation</td>
</tr>
</tbody>
</table>
Flash Development Toolkit (FDT)

- Low-level code for Flash erase and write
- Supports Boot mode and User Program mode
- Supplied in high-level C language source code
- Contains the code primitives necessary to erase and write the Flash memory
- Helps you generate a set of routines tailored for the clock rate and compiler you are using

Features of the FDT:

- A Flash programming utility for Renesas MCUs with on-chip Flash memory
- Development environment for Flash support
- Repository of Flash algorithm source code for writing and erasing the on-chip Flash
- Simplifies creation of a customized Flash control program for User Program Mode
Off-board Programming

- Programming is done in Programmer mode on an automatic device writer or PROM programmer.

- One or more devices can be programmed in parallel using a socket.

- Adapter board makes a Flash MCU appear to the programming equipment to be a standard Flash memory chip.

- This mode allows high-volume production programming of Flash MCUs.

- Programming speed is a key requirement.

- Programming is performed on commercially available products, such as PROM programmers from Minato, Data I/O, BP Microsystems, and others.

- Application code development is done off-line on a PC.
Is the following statement true or false?

"The Flash Development Kit (FDT) encompasses a Flash programming utility, a development environment for Flash support, an algorithm for Flash erase and rewrite, and a utility that helps you build the customized kernel needed for the User Program mode."

- True
- False
Safeguarding Flash Content

Hardware Protection
- To enter program/erase-protected state, initialize registers FLMCR1, FLMCR2, EBR1, and EBR2.

Software Protection
- Set the SWE bit in Flash memory control register 1 (FLMCR1) to erase block registers 1 and 2 (EBR1, EBR2).

Error Protection
- Error protection is automatic. If an error occurs, then the FLER bit is set to 1.
- Program mode or erase mode cannot be re-entered by re-setting the P1 or E1 bit.
- Error is flagged and program/erase operation is aborted when:
  - MCU runaway occurs during Flash memory programming/erasing, or
  - Program/erase operation is not performed in accordance with the appropriate algorithm.
Programming/Erasing Precautions

- Use specified voltages and timing.
- Do not apply a high level to the FWE pin until Vcc has stabilized.
- Set FWE to 0 before turning off Vcc.
- Set FWE to 1 when MCU is stable.
- Do not connect FWE to Vcc.
- Do not clear SWE during program execution.
Which of the following statements correctly describe the programming aspects of the on-chip Flash on H8S MCU devices? Select all that apply and then click Done.

- Boot mode is used for low-volume programming of Flash MCUs
- Boot mode does not allow selective erasing of the Flash memory cells
- Programmer mode is the fastest way to erase and write the Flash memory
- Boot mode and User Program mode are off-board programming modes
- User Program mode can erase portions of the on-chip Flash selectively
Module Summary

- Flash memory design and features
- Block configuration and memory maps
- Operating modes
- Programming modes
  - On-board: Boot mode, User Program mode
  - Off-board: Programmer mode
- Flash protection
- Application tips