H8S/2000 Series Vol. 1

H8S/2655 Series
H8S/2350 Series
H8S/2355 Series
H8S/2357 Series
H8S/2345 Series

Overview

HITACHI

ADE-802-221
Rev. 1.0
3/30/98
Hitachi, Ltd.
MC-Setsu

Hitachi
semiconductor

Notice

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.

2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi’s permission.

3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user’s unit according to this document.

4. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi’s semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.

5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.

6. MEDICAL APPLICATIONS: Hitachi’s products are not authorized for use in MEDICAL APPLICATIONS without the written consent of the appropriate officer of Hitachi’s sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi’s products are requested to notify the relevant Hitachi sales offices when planning to use the products in MEDICAL APPLICATIONS.

Preface

The H8S/2000 Series is a new series of Hitachi-original 16-bit microcomputers which offer higher performance and lower power consumption than the existing H8 Series—widely used for machine control, etc.—together with significantly greater ease of use. The new microcomputers include a CPU, ROM, RAM, a DMA controller (DMAC), data transfer controller (DTC), timers, a serial communication interface (SCI), A/D converter, D/A converter, and other supporting modules on a single chip, enabling them to be used in a wide range of applications covering all system sizes from small to large.

Another major feature of the H8S/2000 Series is CPU object-level upward-compatibility with the H8/300H Series, H8/300 Series, and H8/300L Series within the H8 Series, enabling existing software resources to be used.

Series	Features
H8S/2000	Upward-compatible with the H8/300H Series and H8/300 Series; twice the performance at the same frequency; multiply-and-accumulate instructions (for the H8S/2600 CPU only)
H8/300H	16-Mbyte linear address space; upward-compatible with the H8/300 Series; concise instruction set; powerful word-size and longword-size arithmetic instructions
H8/300	64-kbyte address space; general register system; concise instruction set; powerful bit manipulation instructions
H8/300L	Same CPU as the H8/300 Series; consumer application oriented peripheral functions; low voltage, low power consumption

This Overview outlines the features and functions of the H8S/2655 Series, H8S/2350 Series, H8S/2355 Series, H8S/2357, and H8S/2345 Series products within the H8S/2000 Series—products with powerful on-chip supporting functions suited to high-speed, high-precision real-time control—together with a brief description of the development environment.

Information on other H8S/2000 Series products is available in separate Overview publications.

Contents

Section 1 Features of H8S Series

1.1 H8S/2000 Series Functions

H8S/2000 Series microcomputers are designed for faster instruction execution, using a realtime control oriented CPU with an internal 32-bit architecture, and can run programs based on the C high-level language efficiently. As well as large-capacity ROM and RAM, these microcomputers include on-chip the peripheral functions needed for control systems. Easy and fast connection of external memory is possible, simplifying the implementation of sophisticated, high-performance systems.

The features of the H8S/2000 Series are shown below. The on-chip functions differ from product to product; see sections 1.2, Product Evolution Diagram and List of Functions, and 5, Guide to Products, for details.

High-performance H8S/2600 CPU, H8S/2000 CPU

- General-register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)

- High-speed operation suitable for realtime control
  - All frequently used instructions execute in 1 to 2 states
  - High-speed arithmetic operations (at 20 MHz operation)

8/16/32-bit register-register add/subtract:	50 ns
8 × 8-bit register-register multiply:	150 ns (H8S/2000 CPU: 600 ns)
16 ÷ 8-bit register-register divide:	600 ns
16 × 16-bit multiply-and-accumulate	200 ns (H8S/2000 CPU: 1000 ns)
32 ÷ 16-bit register-register divide:	1000 ns

- Instruction set suitable for high-speed operation
  - Sixty-nine basic instructions (sixty-five in H8S/2000 CPU)
  - 8/16/32-bit move/arithmetic and logic instructions
  - Unsigned/signed multiply and divide instructions
  - Multiply-and-accumulate instruction (H8S/2600 CPU only)
  - Powerful bit-manipulation instructions

- Two CPU operating modes
  - Normal mode*: H8/300 Series compatible, maximum 64-kbyte address space
  - Advanced mode: Maximum 16-Mbyte address space

Note: * Not supported in the H8S/2357 Series and H8S/2345 F-ZTAT™.

Supporting Modules

The supported functions vary from series to series; see section 5, Guide to Products, for details.

- Bus controller
  - Address space divided into 8 areas, with bus specifications settable independently for each area
  - Chip select output possible for each area
  - Selection of 8-bit or 16-bit access space for each area
  - 2-state or 3-state access space can be designated for each area
  - Number of program wait states can be set for each area
  - Burst ROM directly connectable
  - Maximum 8-Mbyte DRAM or PSRAM directly connectable (The pseudo-SRAM applies only to the H8S/2655 Series)
  - External bus release function

- DMA controller (DMAC)
  - Selection of short address mode or full address mode
  - Four channels in short address mode, two channels in full address mode
  - Transfer possible in repeat mode, block transfer mode, etc.
  - Single address mode transfer possible
  - Can be activated by internal interrupt

- Data transfer controller (DTC)
  - Activated by internal interrupt or software
  - Multiple transfers or multiple types of transfer possible for one activation source
  - Transfer possible in repeat mode, block transfer mode, etc.
  - Request can be sent to CPU for interrupt that activated DTC

- 16-bit timer-pulse unit (TPU)
  - Six-channel 16-bit timer on-chip
  - Pulse I/O processing capability for up to 16 pins
  - Automatic 2-phase encoder count capability

- Programmable pulse generator (PPG)
  - Maximum 16-bit pulse output possible with TPU as time base
  - Output trigger selectable in 4-bit groups
  - Non-overlap margin can be set
  - Direct output or inverse output setting possible

- 8-bit timer
  — 8-bit up-counter (external event count capability)
  — Two time constant registers
  — Two-channel connection possible

- Watchdog timer (WDT)
  — Watchdog timer or interval timer selectable

- On-chip serial communication interface (SCI)
  — Asynchronous mode or synchronous mode selectable
  — Multiprocessor communication function
  — Smart card interface function

- On-chip A/D converter (type 1)
  — Resolution: 10 bits
  — Input: 8 channels
  — High-speed conversion: 2.2 µs minimum conversion time (at 20 MHz operation)
  — Select or group mode, and single or scan mode selectable
  — Sample and hold circuit
  — A/D conversion can be activated by external trigger or timer trigger

- On-chip A/D converter (type 2)
  — Resolution: 10 bits
  — Input: 8 channels
  — Minimum conversion time: 6.7 µs (at 20 MHz operation)
  — Single or scan mode selectable
  — Sample and hold circuit
  — A/D conversion can be activated by external trigger or timer trigger

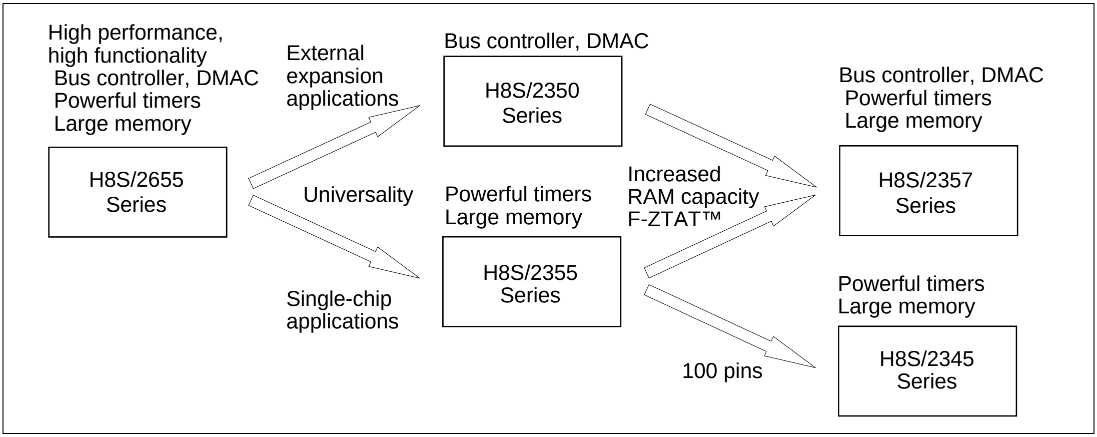
- On-chip D/A converter
  — Resolution: 8 bits
  — Output: 2 channels

- On-chip interrupt controller
  — Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
  — Selection of four interrupt control modes, with priority settable for each module

- Fine power consumption control
  - Medium-speed mode
  - Sleep mode
  - Module stop mode
  - Software standby mode
  - Hardware standby mode

1.2 Product Evolution Diagram and List of Functions

Evolution Diagram



List of Functions

Series		H8S/2655		H8S/2350		H8S/2355		H8S/2357		H8S/2345	
Type Name		**H8S/2655**	**H8S/2653**	**H8S/2350**	**H8S/2351**	**H8S/2355**	**H8S/2353**	**H8S/2357**	**H8S/2352**	**H8S/2345**	**H8S/2343**
CPU	H8S/2600 CPU	●	●								
	H8S/2000 CPU			●	●	●	●	●	●	●	●
MCU operating mode	Mode 1	●	●	●	●	●	●			●*1	●
	Mode 2	●	●		●	●	●			●*1	●
	Mode 3	●	●		●	●	●			●*1	●
	Mode 4	●	●	●	●	●	●	●	●	●	●
	Mode 5	●	●	●	●	●	●	●	●	●	●
	Mode 6	●	●		●	●	●	●	●	●	●
	Mode 7	●	●		●	●	●	●	●	●	●
	Mode 10							●	●	●*2	
	Mode 11							●	●	●*2	
	Mode 14							●	●	●*2	
	Mode 15							●	●	●*2	
ROM	128 kbytes	●				●		●		●	
	64 kbytes		●		●		●				●
	ROMless version			●					●		
	Mask ROM version	●	●		●	●	●	●		●	●
	ZTAT version*5	●				●		●		●	
	F-ZTAT version*6							●		●*4	
RAM	2 kbytes			●	●		●				●
	4 kbytes	●	●			●				●	
	8 kbytes							●	●		
Bus controller	DRAM interface	●	●	●	●			●	●		
	PSRAM interface	●	●								
	Burst ROM interface	●	●	●	●	●	●	●	●	●	●
	Write buffer function	●	●	●	●			●	●		
Interrupt controller	Interrupt control modes	4	4	2	2	2	2	2	2	2	2
	External interrupts	9	9	9	9	9	9	9	9	9	9
	Internal interrupts	52	52	42	42	47	47	52	52	43	43
Reset	Power-on reset	●	●	●	●	●	●	●	●	●	●
	Manual reset	●	●	●	●	●	●	●*7	—	●*8	●*8

List of Functions (cont)

Series		H8S/2655		H8S/2350		H8S/2355		H8S/2357		H8S/2345	
Type Name		**H8S/2655**	**H8S/2653**	**H8S/2350**	**H8S/2351**	**H8S/2355**	**H8S/2353**	**H8S/2357**	**H8S/2352**	**H8S/2345**	**H8S/2343**
DMAC		●	●	●	●			●	●		
DTC		●	●	●	●	●	●	●	●	●	●
Timers	TPU (16-bit)	X6	X6	X6	X6	X6	X6	X6	X6	X6	X6
	PPG	●	●	●	●			●	●		
	8-bit timers	X2	X2			X2	X2	X2	X2	X2	X2
	WDT	X1	X1	X1	X1	X1	X1	X1	X1	X1	X1
SCI	Channels	X3	X3	X2	X2	X3	X3	X3	X3	X2	X2
	Asynchronous mode	●	●	●	●	●	●	●	●	●	●
	Synchronous mode	●	●	●	●	●	●	●	●	●	●
	Multiprocessor function	●	●	●	●	●	●	●	●	●	●
	Smart card interface function	●	●	●	●	●	●	●	●	●	●
A/D converter	10-bit (type 1)	●	●								
	10-bit (type 2)			●	●	●	●	●	●	●	●
D/A converter	8-bit	●	●	●	●	●	●	●	●	●	●
I/O ports	Input pins	8	8	8	8	8	8	8	8	8	8
	I/O pins	87	87	87	87	87	87	87	87	71	71
Packages	TFP-120	●	●	●	●	●	●	●	●		
	FP-128	●	●	●	●	●	●	●	●		
	TFP-100B									●	●
	TFP-100G									●	●
	FP-100A									●	●
	FP-100B									●	●
Maximum operating frequency	2.7 to 5.5 V	10 MHz	10 MHz	10 MHz	10 MHz	10 MHz	10 MHz	10 MHz	10 MHz	10 MHz	10 MHz
	3.0 to 5.5 V	—	—	13 MHz	13 MHz	13 MHz^{*3}	13 MHz	13 MHz^{*4}	13 MHz	—	—
	5 V	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz

Notes:
1. Applies to the ZTAT and mask ROM versions only.
2. Applies to the F-ZTAT versions only.
3. Applies to the mask ROM versions only.
4. Under development
5. ZTAT is a trademark of Hitachi, Ltd.
6. F-ZTAT is a trademark of Hitachi, Ltd.
7. Manual reset is not supported in the H8S/2357 F-ZTAT version and mask ROM version.
8. Please contact your Hitachi sales representative for information on manual reset in the H8S/2345 Series.

●: Available

1.3 ZTAT™ and F-ZTAT™ Microcomputers

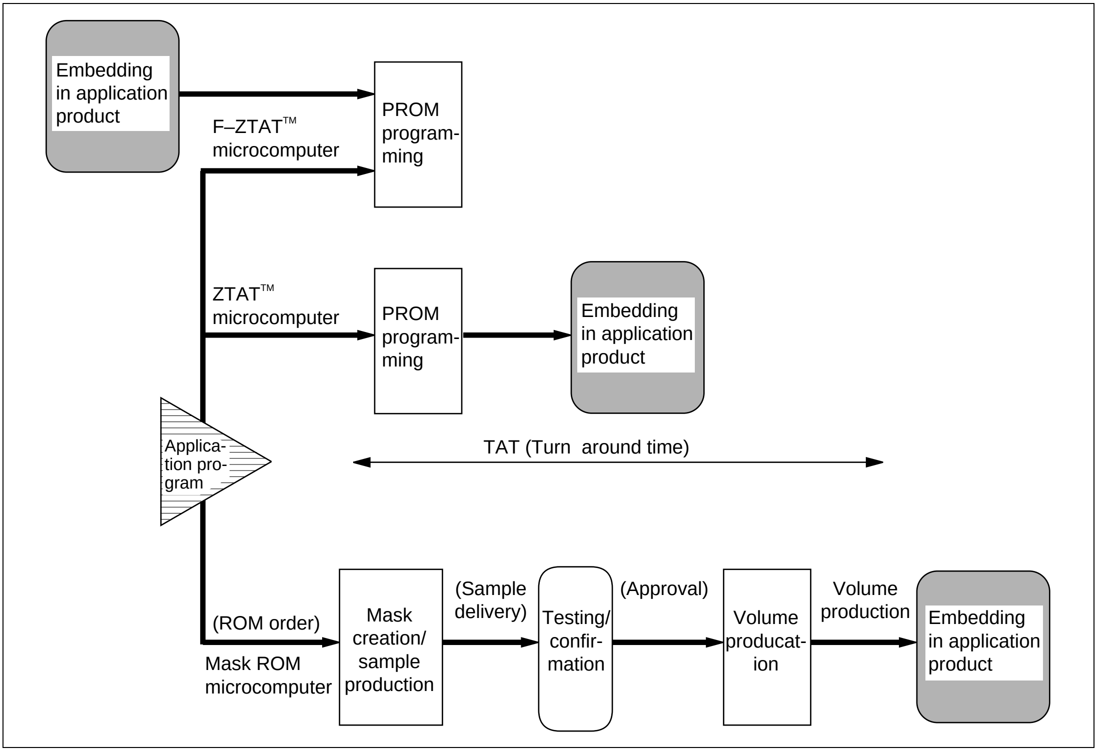
H8S/2000 Series products are available in ZTAT™ versions as well as mask ROM versions. ZTAT™ microcomputers have on-chip user-programmable ROM. After program development is complete, the user can write to the program ROM in situ, enabling faster product development. Versions with on-chip flash memory (H8S/2357 F-ZTAT™ and H8S/2345 F-ZTAT™) are also available.

ZTAT™ micrompters

- Shorter development TAT (Turn Around Time) and lower development costs
  - A completed program does not have to sent to the manufacturer, but can be written by the user himself
- Facilitates small-volume, multi-model production
  - More economical than mask ROM versions for small-volume production
  - Easily copes with fields requiring frequent specification changes
- General-purpose PROM writer can be used
  - Programming only requires connection of a special socket adapter to a general-purpose PROM writer

F-ZTAT™ microcomputer

- Shorter development TAT
  - Final adjustments can be made after product completion, allowing hardware-first development
- Lower development costs
  - Repeated programming possible in the development stage
  - Allows partial reprogramming even of small-volume products
  - Can be programmed and erased simply by connecting a special socket adapter to a general-purpose PROM writer
- Flexible system construction
  - On-board reprogramming allows unit-by-unit tuning
  - Partial program amendment possible in on-board state
  - Post-shipment version upgrading without changing board or chip


Embedding in application product
PROM program-ming
F–ZTAT™ microcomputer
ZTAT™ microcomputer
PROM program-ming
Embedding in application product
Applica-tion pro-gram
TAT (Turn around time)
(ROM order)
Mask ROM microcomputer
Mask creation/ sample production
(Sample delivery)
Testing/ confir-mation
(Approval)
Volume producat-ion
Volume production
Embedding in application product

Section 2 CPU

2.1 Overview

H8S/2000 Series products use either an H8S/2600 Series or H8S/2000 Series CPU core.

The H8S/2600 and H8S/2000 CPU are a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The CPUs have sixteen 16-bit general registers, can address a 16-Mbyte (architecturally 4-Gbyte) linear address space, and is ideal for realtime control.

Features

- Upward-compatible with H8/300 CPU
  — Can execute H8/300 and H8/300H object programs
- General-register architecture
  — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- Sixty-nine (H8S/2600 CPU) or sixty-five (H8S/2000 CPU) basic instructions
  — 8/16/32-bit arithmetic and logic instructions
  — Multiply and divide instructions
  — Powerful bit-manipulation instructions
  — Multiply-and accumulate instruction (H8S/2600 CPU only)
- Eight addressing modes
  — Register direct [Rn]
  — Register indirect [@ERn]
  — Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  — Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  — Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  — Immediate [#xx:8, #xx:16, or #xx:32]
  — Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  — Memory indirect [@@aa:8]
- 16-Mbyte address space
  — Program: 16 Mbytes
  — Data: 16 Mbytes (4 Gbytes architecturally)

- High-speed operation
  - All frequently-used instructions execute in one or two states
  - Maximum clock frequency: 20 MHz
  - 8/16/32-bit register-register add/subtract: 50 ns
  - 8 × 8-bit register-register multiply: 150 ns (H8S/2600 CPU) or 600 ns (H8S/2000 CPU)
  - 16 ÷ 8-bit register-register divide: 600 ns
  - 16 × 16-bit register-register multiply: 200 ns (H8S/2600 CPU) or 1000 ns (H8S/2000 CPU)
  - 32 ÷ 16-bit register-register divide: 1000 ns

- Two CPU operating modes
  - Normal mode
  - Advanced mode

- Power-down state
  - Transition to power-down state by SLEEP instruction
  - CPU clock speed selection

Differences between H8S/2600 CPU and H8S/2000 CPU: The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration
  - The MAC register is supported only by the H8S/2600 CPU.

- Basic instructions
  - The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.

- Number of execution states
  - The number of execution states of the MULXU and MULXS instructions.

The address space, EXR register functions, power-down states, etc., may vary depending on the product. Refer to the hardware manual for the relevant product for details.

Differences from H8/300 CPU: In comparison to the H8/300 CPU, the H8S/2600 CPU and the H8S/2000 CPU have the following enhancements.

- More general registers and control registers
  - Eight 16-bit expanded registers, and one 8-bit control register, have been added.

- Expanded address space
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16-Mbyte address space.

- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.

- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.

- Higher speed
  - Basic instructions execute twice as fast.

Differences from H8/300H CPU: In comparison to the H8/300H CPU, the H8S/2600 CPU and the H8S/2000 CPU have the following enhancements.

- Additional control register
  - One 8-bit control register has been added.

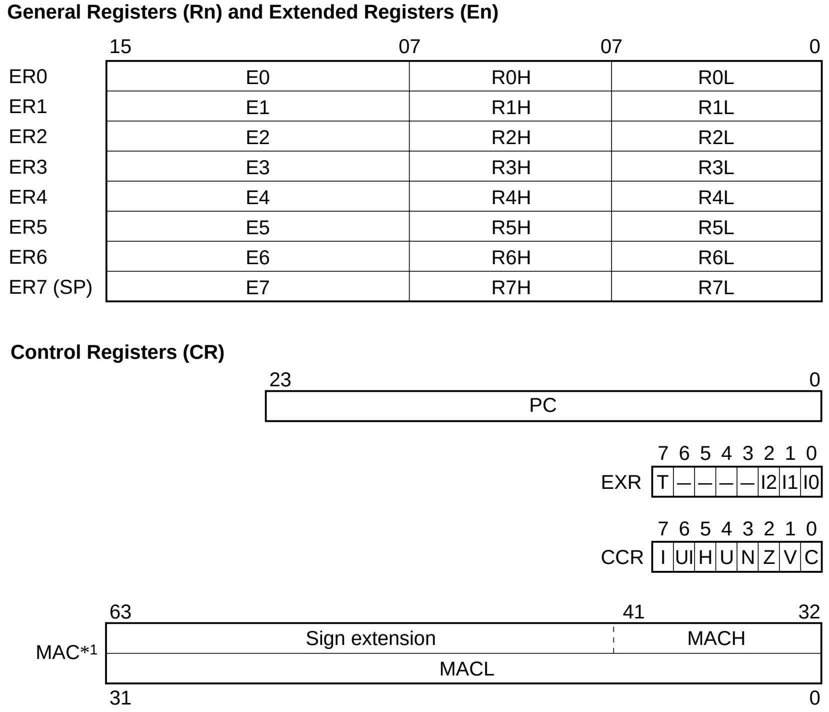
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.

- Higher speed
  - Basic instructions execute twice as fast.

2.2 Register Configuration

The H8S/2600 and H8S/2000 CPU have general registers and control registers. The eight 32-bit general registers all have identical functions and can be used as either address registers or data registers. The control registers are the 24-bit program counter (PC), 8-bit extended register (EXR), 8-bit condition code register (CCR), and 64-bit multiply-and-accumulate register (MAC).

CPU Registers



Legend:

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit	MAC:	Multiply-accumulate register*1
UI:	User bit or interrupt mask bit*2		

Notes: 1. MAC applies only to the H8S/2600 CPU.

2. Use as an interrupt mask bit is not possible with some series. Refer to the hardware manual for the relevant product for details.

CPU Registers

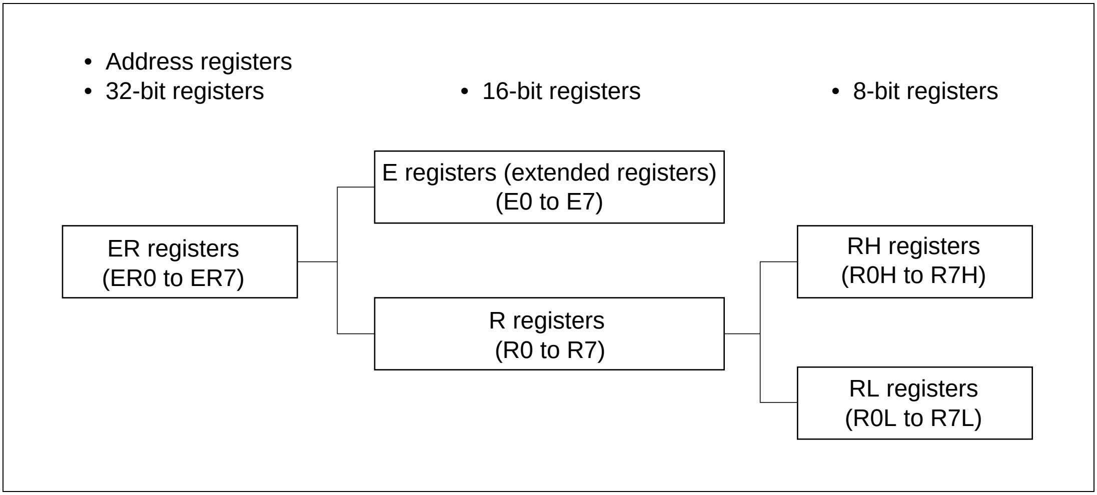
General Registers: The CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

The figure below illustrates the usage of the general registers. The usage of each register can be selected independently.

Usage of General Registers


• Address registers
• 32-bit registers
• 16-bit registers
• 8-bit registers
E registers (extended registers) (E0 to E7)
ER registers (ER0 to ER7)
R registers (R0 to R7)
RH registers (R0H to R7H)
RL registers (R0L to R7L)

Control Registers: The control registers are the 24-bit program counter (PC), 8-bit extended control register (EXR), and 8-bit condition-code register (CCR), and 64-bit multiply-and-accumulate register (MAC)*.

- **Program Counter (PC)**

  This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0.)

- **Extended Control Register (EXR)**

  An 8-bit register. Includes a trace bit (T) and interrupt mask bits (I2 to I0).

  - Bit 7—Trace Bit (T): Specifies whether or not trace mode is set. When this bit is cleared to 0, instructions are executed sequentially. When set to 1, trace exception handling is started each time an instruction is executed.
  - Bits 6 to 3—Reserved
  - Bits 2 to 0—Interrupt Mask Bits (I2 to I0): These bits specify the interrupt request mask level (0 to 7). See section 2.9, Interrupts, for details.

    EXR can be manipulated by the LCD, STC, ANDC, ORC, and XORC instructions. Except in the case of STC, interrupts (including NMI) are not accepted for 3 states after the instruction is executed.

- **Condition-Code Register (CCR)**

  This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

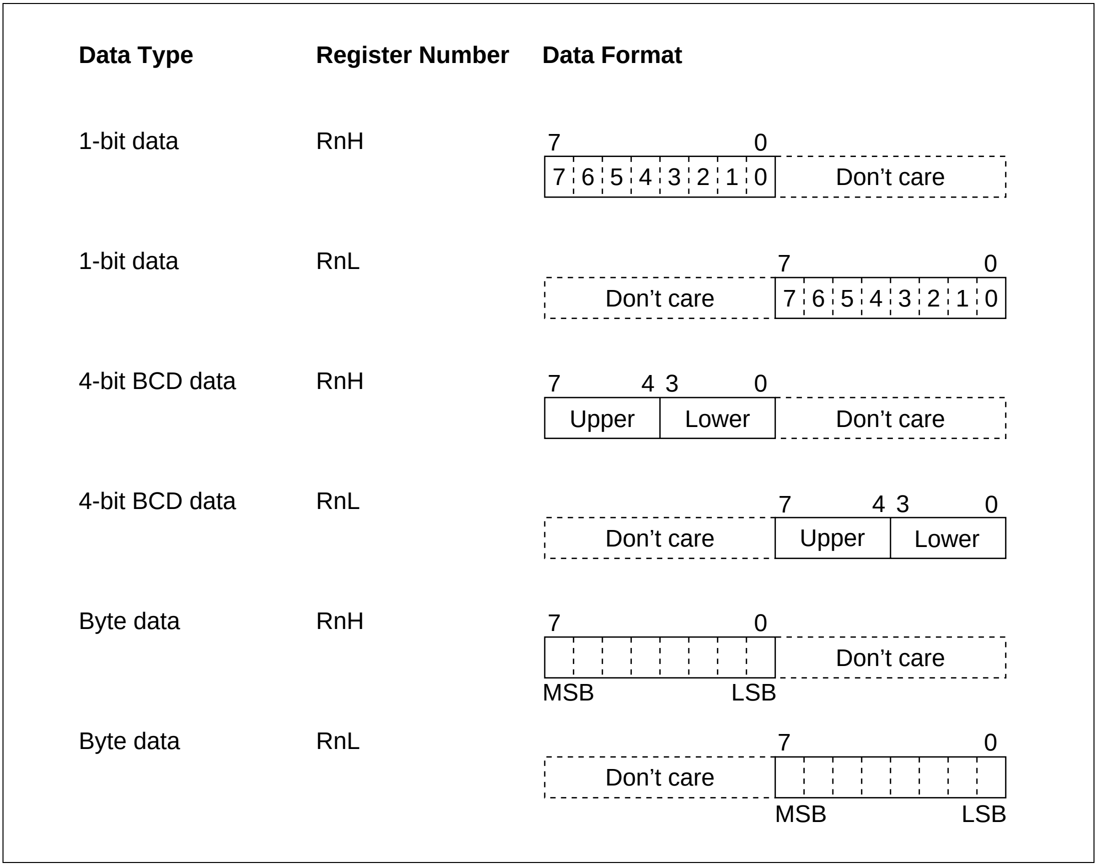
  - Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. (NMI is accepted regardless of the I bit setting.) The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 2.9, Interrupts.
  - Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details, refer to section 2.9, Interrupts.
  - Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.
  - Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.
  - Bit 3—Negative Flag (N): Stores the value of the most significant bit (sign bit) of data.
  - Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.
  - Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.
  - Bit 0—Carry Flag (C): Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:
    - Add instructions, to indicate a carry
    - Subtract instructions, to indicate a borrow
    - Shift and rotate instructions, to store the carry

  The carry flag is also used as a bit accumulator by bit manipulation instructions.

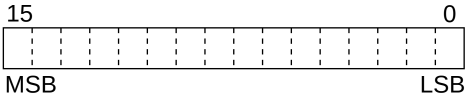
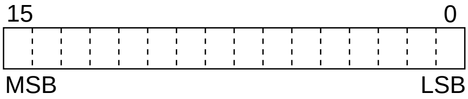
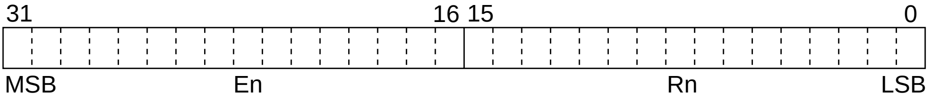
2.3 Data Formats

The CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

General Register Data Formats



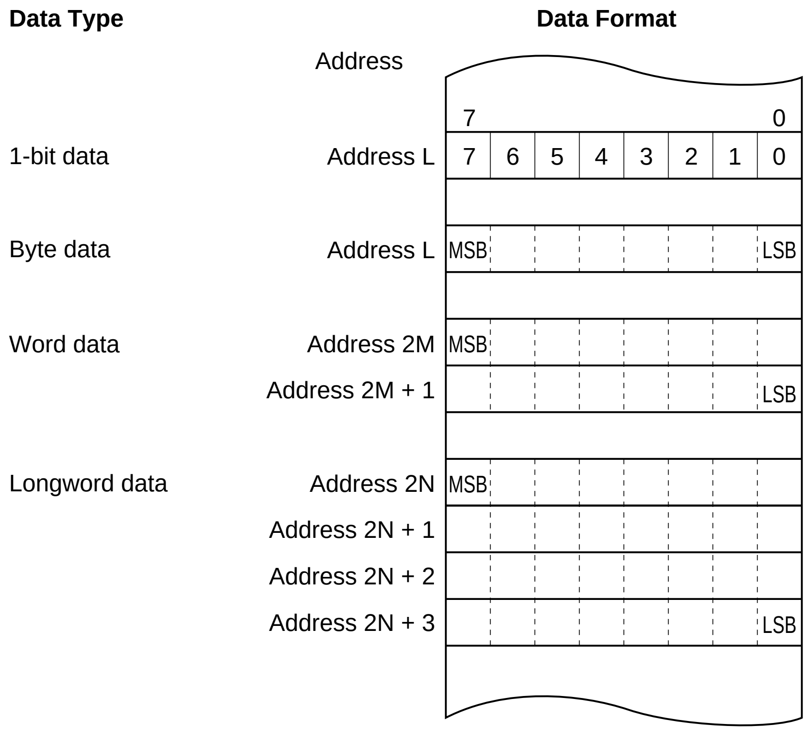
General Register Data Formats (cont)

Data Type	Register Number	Data Format
Word data	Rn	
Word data	En	
Longword data	ERn	

Legend:

ERn:	General register ER
En:	General register E
Rn:	General register R
RnH:	General register RH
RnL:	General register RL
MSB:	Most significant bit
LSB:	Least significant bit

Memory Data Formats



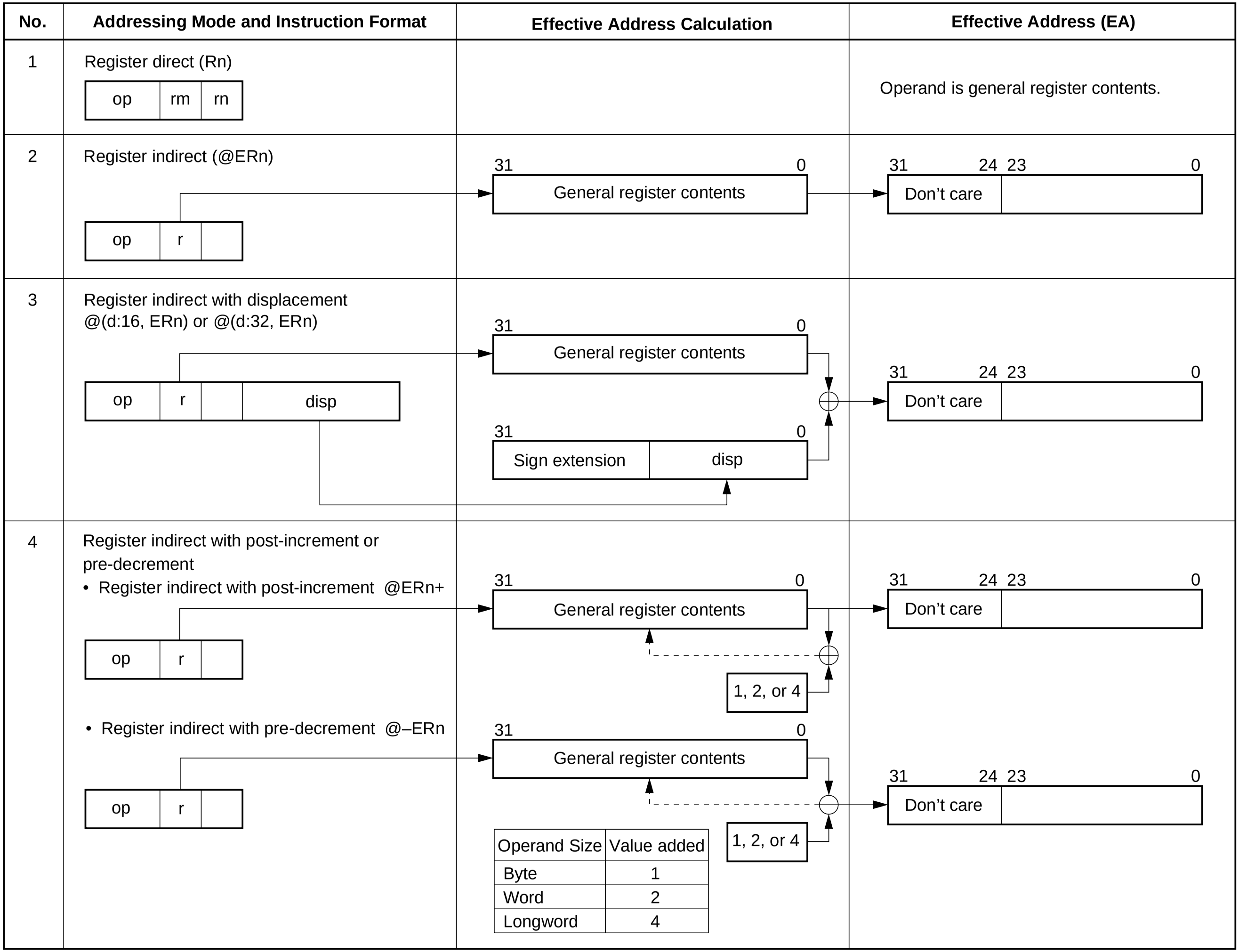
2.4 Addressing Modes

The H8S/2000 CPU supports eight addressing modes. The H8S/2600 CPU supports eight addressing modes for easy use of the contents of the 16-Mbyte (architecturally 4-Gbyte) address space.

Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

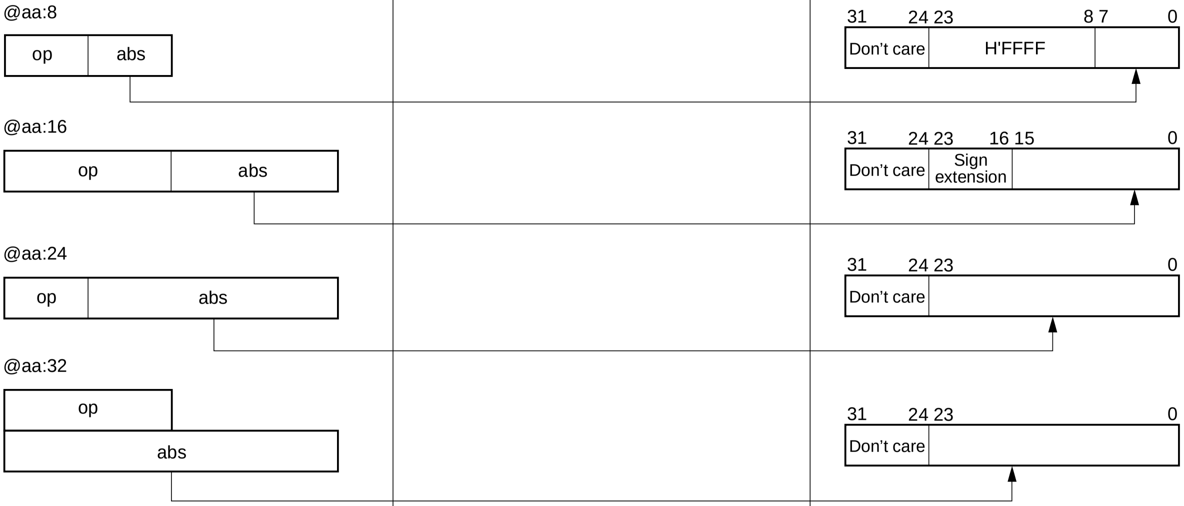
Effective Address Calculation: In normal mode, the upper 8 bits of the effective address are ignored, giving a 16-bit address.

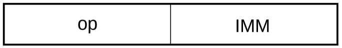


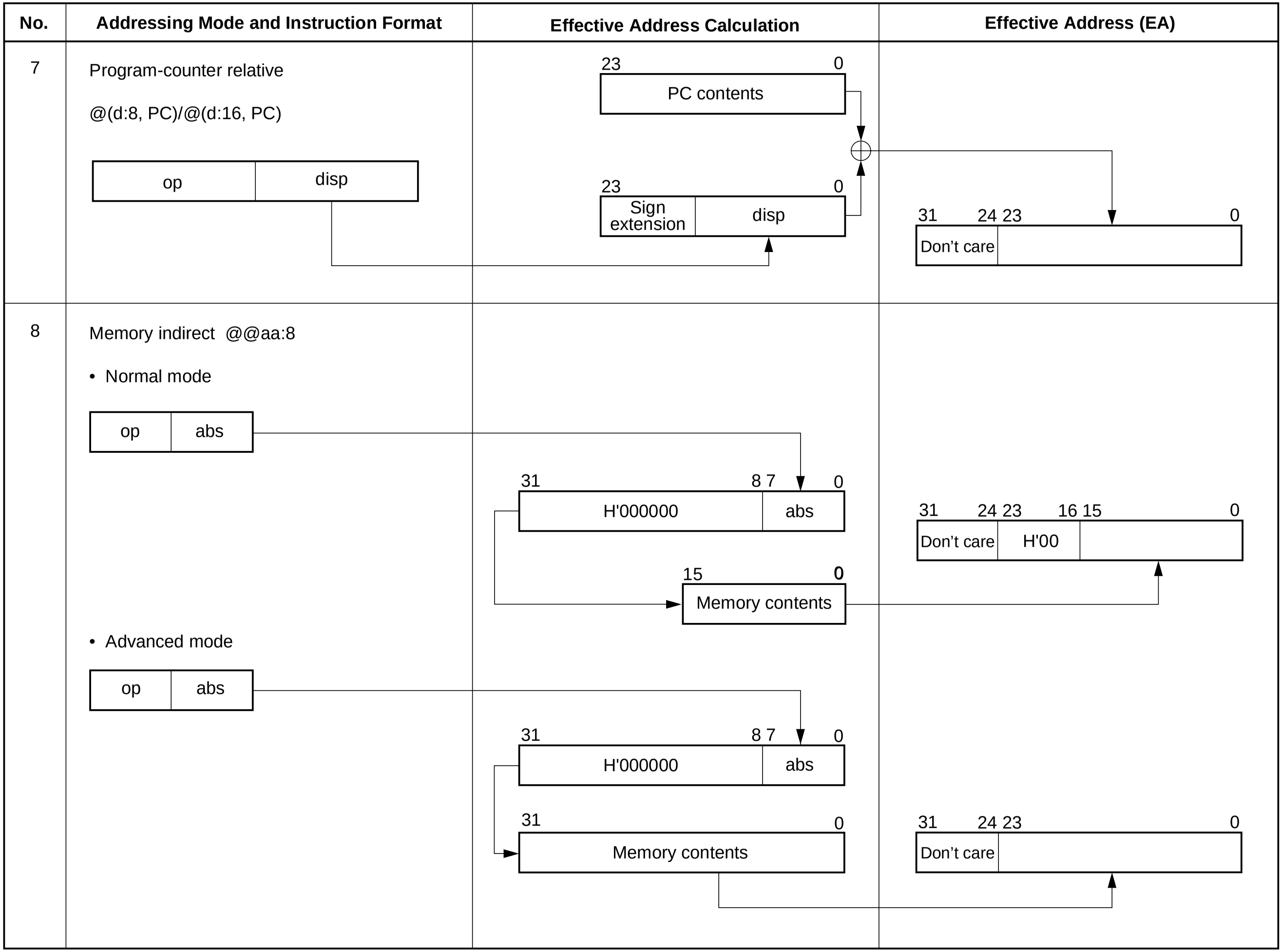
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn)		Operand is general register contents.
2	Register indirect (@ERn)		
3	Register indirect with displacement @(d:16, ERn) or @(d:32, ERn)		
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+ • Register indirect with pre-decrement @–ERn		

Operand Size	Value added
Byte	1
Word	2
Longword	4

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8 @aa:16 @aa:24 @aa:32		
6	Immediate #xx:8/#xx:16/#xx:32		Operand is immediate data.







No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
7	Program-counter relative @(d:8, PC)/@(d:16, PC)		
8	Memory indirect @@aa:8 • Normal mode • Advanced mode		

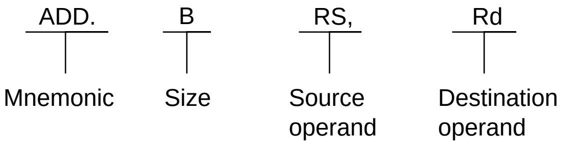
2.5 Instruction Set

The H8S/2600 CPU has 69 types of instructions, and the H8S/2000 CPU has 65 types of instructions.

Features

- Upward-compatible at object level with H8/300H and H8/300 CPUs
- General register architecture
- 8/16/32-bit transfer instructions and arithmetic and logic instructions
  - — Byte (B), word (W), and longword (L) formats for transfer instructions and basic arithmetic and logic instructions
- Unsigned and signed multiply and divide instructions
- Multiply-and-accumulate instruction (H8S/2600 CPU only)
- Powerful bit manipulation instructions
- Instructions for saving and restoring multiple registers

Assembler Format: The ADD instruction format is shown below as an example.



- Data transfer instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
MOV	MOV.B #xx:8,Rd	B	2									#xx:8→Rd8	—	—	↕	↕	0	—	1	
	MOV.B Rs,Rd	B		2								Rs8→Rd8	—	—	↕	↕	0	—	1	
	MOV.B @ERs,Rd	B			2							@ERs→Rd8	—	—	↕	↕	0	—	2	
	MOV.B @(d:16,ERs),Rd	B				4						@(d:16,ERs)→Rd8	—	—	↕	↕	0	—	3	
	MOV.B @(d:32,ERs),Rd	B				8						@(d:32,ERs)→Rd8	—	—	↕	↕	0	—	5	
	MOV.B @ERs+,Rd	B					2					@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—	3	
	MOV.B @aa:8,Rd	B						2				@aa:8→Rd8	—	—	↕	↕	0	—	2	
	MOV.B @aa:16,Rd	B						4				@aa:16→Rd8	—	—	↕	↕	0	—	3	
	MOV.B @aa:32,Rd	B						6				@aa:32→Rd8	—	—	↕	↕	0	—	4	
	MOV.B Rs,@ERd	B			2							Rs8→@ERd	—	—	↕	↕	0	—	2	
	MOV.B Rs,@(d:16,ERd)	B				4						Rs8→@(d:16,ERd)	—	—	↕	↕	0	—	3	
	MOV.B Rs,@(d:32,ERd)	B				8						Rs8→@(d:32,ERd)	—	—	↕	↕	0	—	5	
	MOV.B Rs,@-ERd	B					2					ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—	3	
	MOV.B Rs,@aa:8	B						2				Rs8→@aa:8	—	—	↕	↕	0	—	2	
	MOV.B Rs,@aa:16	B						4				Rs8→@aa:16	—	—	↕	↕	0	—	3	
	MOV.B Rs,@aa:32	B						6				Rs8→@aa:32	—	—	↕	↕	0	—	4	
	MOV.W #xx:16,Rd	W	4									#xx:16→Rd16	—	—	↕	↕	0	—	2	
	MOV.W Rs,Rd	W		2								Rs16→Rd16	—	—	↕	↕	0	—	1	
	MOV.W @ERs,Rd	W			2							@ERs→Rd16	—	—	↕	↕	0	—	2	
	MOV.W @(d:16,ERs),Rd	W				4						@(d:16,ERs)→Rd16	—	—	↕	↕	0	—	3	
	MOV.W @(d:32,ERs),Rd	W				8						@(d:32,ERs)→Rd16	—	—	↕	↕	0	—	5	
	MOV.W @ERs+,Rd	W					2					@ERs→Rd16,ERs32+2→ERs32	—	—	↕	↕	0	—	3	
	MOV.W @aa:16,Rd	W						4				@aa:16→Rd16	—	—	↕	↕	0	—	3	
	MOV.W @aa:32,Rd	W						6				@aa:32→Rd16	—	—	↕	↕	0	—	4	
	MOV.W Rs,@ERd	W			2							Rs16→@ERd	—	—	↕	↕	0	—	2	
	MOV.W Rs,@(d:16,ERd)	W				4						Rs16→@(d:16,ERd)	—	—	↕	↕	0	—	3	
	MOV.W Rs,@(d:32,ERd)	W				8						Rs16→@(d:32,ERd)	—	—	↕	↕	0	—	5	
	MOV.W Rs,@-ERd	W					2					ERd32-2→ERd32,Rs16→@ERd	—	—	↕	↕	0	—	3	
	MOV.W Rs,@aa:16	W						4				Rs16→@aa:16	—	—	↕	↕	0	—	3	
	MOV.W Rs,@aa:32	W						6				Rs16→@aa:32	—	—	↕	↕	0	—	4	
	MOV.L #xx:32,ERd	L	6									#xx:32→ERd32	—	—	↕	↕	0	—	3	
	MOV.L ERs,ERd	L		2								ERs32→ERd32	—	—	↕	↕	0	—	1	
	MOV.L @ERs,ERd	L			4							@ERs→ERd32	—	—	↕	↕	0	—	4	
	MOV.L @(d:16,ERs),ERd	L				6						@(d:16,ERs)→ERd32	—	—	↕	↕	0	—	5	
	MOV.L @(d:32,ERs),ERd	L				10						@(d:32,ERs)→ERd32	—	—	↕	↕	0	—	7	
	MOV.L @ERs+,ERd	L					4					@ERs→ERd32,ERs32+4→ERs32	—	—	↕	↕	0	—	5	
	MOV.L @aa:16,ERd	L						6				@aa:16→ERd32	—	—	↕	↕	0	—	5	
	MOV.L @aa:32,ERd	L						8				@aa:32→ERd32	—	—	↕	↕	0	—	6	
	MOV.L ERs,@ERd	L			4							ERs32→@ERd	—	—	↕	↕	0	—	4	
	MOV.L ERs,@(d:16,ERd)	L				6						ERs32→@(d:16,ERd)	—	—	↕	↕	0	—	5	
	MOV.L ERs,@(d:32,ERd)	L				10						ERs32→@(d:32,ERd)	—	—	↕	↕	0	—	7	
	MOV.L ERs,@-ERd	L					4					ERd32-4→ERd32,ERs32→@ERd	—	—	↕	↕	0	—	5	
	MOV.L ERs,@aa:16	L						6				ERs32→@aa:16	—	—	↕	↕	0	—	5	
	MOV.L ERs,@aa:32	L						8				ERs32→@aa:32	—	—	↕	↕	0	—	6	
POP	POP.W Rn	W									2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—	3	
	POP.L ERn	L									4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—	5	
PUSH	PUSH.W Rn	W									2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—	3	
	PUSH.L ERn	L									4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—	5	
LDM	LDM @SP+,(ERm-ERn)	L									4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—	7/9/11 [1]	
STM	STM (ERm-ERn),@-SP	L									4	(SP-4→SP,ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—	7/9/11 [1]	
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used with these series.																	[2]	
MOVTPE	MOVTPE Rs,@aa:16																		[2]	

- Arithmetic instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
ADD	ADD.B #xx:8,Rd	B	2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕	1	
	ADD.B Rs,Rd	B		2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕	1	
	ADD.W #xx:16,Rd	W	4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2	
	ADD.W Rs,Rd	W		2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕	1	
	ADD.L #xx:32,ERd	L	6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3	
	ADD.L ERs,ERd	L		2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕	1	
ADDX	ADDX #xx:8,Rd	B	2									Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕	↕	1	
	ADDX Rs,Rd	B		2								Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕	↕	1	
ADDS	ADDS #1,ERd	L		2								ERd32+1→ERd32	—	—	—	—	—	—	1	
	ADDS #2,ERd	L		2								ERd32+2→ERd32	—	—	—	—	—	—	1	
	ADDS #4,ERd	L		2								ERd32+4→ERd32	—	—	—	—	—	—	1	
INC	INC.B Rd	B		2								Rd8+1→Rd8	—	—	↕	↕	↕	—	1	
	INC.W #1,Rd	W		2								Rd16+1→Rd16	—	—	↕	↕	↕	—	1	
	INC.W #2,Rd	W		2								Rd16+2→Rd16	—	—	↕	↕	↕	—	1	
	INC.L #1,ERd	L		2								ERd32+1→ERd32	—	—	↕	↕	↕	—	1	
	INC.L #2,ERd	L		2								ERd32+2→ERd32	—	—	↕	↕	↕	—	1	
DAA	DAA Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	↕	1	
SUB	SUB.B Rs,Rd	B		2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕	1	
	SUB.W #xx:16,Rd	W	4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2	
	SUB.W Rs,Rd	W		2								Rd16-Rs16→Rd16	—	[3]	↕	↕	↕	↕	1	
	SUB.L #xx:32,ERd	L	6									ERd32-#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3	
	SUB.L ERs,ERd	L		2								ERd32-ERs32→ERd32	—	[4]	↕	↕	↕	↕	1	
SUBX	SUBX #xx:8,Rd	B	2									Rd8-#xx:8-C→Rd8	—	↕	↕	[5]	↕	↕	1	
	SUBX Rs,Rd	B		2								Rd8-Rs8-C→Rd8	—	↕	↕	[5]	↕	↕	1	
SUBS	SUBS #1,ERd	L		2								ERd32-1→ERd32	—	—	—	—	—	—	1	
	SUBS #2,ERd	L		2								ERd32-2→ERd32	—	—	—	—	—	—	1	
	SUBS #4,ERd	L		2								ERd32-4→ERd32	—	—	—	—	—	—	1	
DEC	DEC.B Rd	B		2								Rd8-1→Rd8	—	—	↕	↕	↕	—	1	
	DEC.W #1,Rd	W		2								Rd16-1→Rd16	—	—	↕	↕	↕	—	1	
	DEC.W #2,Rd	W		2								Rd16-2→Rd16	—	—	↕	↕	↕	—	1	
	DEC.L #1,ERd	L		2								ERd32-1→ERd32	—	—	↕	↕	↕	—	1	
	DEC.L #2,ERd	L		2								ERd32-2→ERd32	—	—	↕	↕	↕	—	1	
DAS	DAS Rd	B		2								Rd8 decimal adjust → Rd8	—	*	↕	↕	*	—	1	
MULXU	MULXU.B Rs,Rd	B		2								Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—	—	—	—	3 (12)*3 to [6]	
	MULXU.W Rs,ERd	W		2								Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—	4 (20)*3 to [6]	
MULXS	MULXS.B Rs,Rd	B		4								Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—	4 (13)*3 to [7]	
	MULXS.W Rs,ERd	W		4								Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—	5 (21)*3 to [7]	
DIVXU	DIVXU.B Rs,Rd	B		2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[6]	[7]	—	—	12	
	DIVXU.W Rs,ERd	W		2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[6]	[7]	—	—	20	
DIVXS	DIVXS.B Rs,Rd	B		4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[8]	[7]	—	—	13	
	DIVXS.W Rs,ERd	W		4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[8]	[7]	—	—	21	
CMP	CMP.B #xx:8,Rd	B	2									Rd8-#xx:8	—	↕	↕	↕	↕	↕	1	
	CMP.B Rs,Rd	B		2								Rd8-Rs8	—	↕	↕	↕	↕	↕	1	
	CMP.W #xx:16,Rd	W	4									Rd16-#xx:16	—	[3]	↕	↕	↕	↕	2	
	CMP.W Rs,Rd	W		2								Rd16-Rs16	—	[3]	↕	↕	↕	↕	1	
	CMP.L #xx:32,ERd	L	6									ERd32-#xx:32	—	[4]	↕	↕	↕	↕	3	
	CMP.L ERs,ERd	L		2								ERd32-ERs32	—	[4]	↕	↕	↕	↕	1	
NEG	NEG.B Rd	B		2								0-Rd8→Rd8	—	↕	↕	↕	↕	↕	1	
	NEG.W Rd	W		2								0-Rd16→Rd16	—	↕	↕	↕	↕	↕	1	
	NEG.L ERd	L		2								0-ERd32→ERd32	—	↕	↕	↕	↕	↕	1	
EXTU	EXTU.W Rd	W		2								0 → (<bits 15 to 8> of Rd16)	—	—	0	↕	0	—	1	
	EXTU.L ERd	L		2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0	—	1	
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	—	1	
	EXTS.L ERd	L		2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	—	1	
TAS	TAS @ERd	B			4							@ERd-0 → CRR set, (1) → (<bit 7> of @ERd)	—	—	↕	↕	0	—	4	

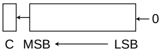
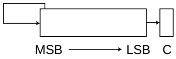
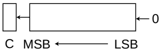
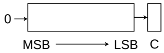
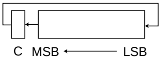
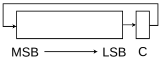
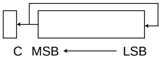
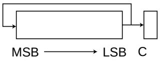
- Arithmetic instructions (cont)

Mnemonic		Operand Size	Addressing Model/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
MAC*4	MAC@ERn+,@ERm+	—					4					@ERn × @ERm+MAC → MAC (signed multiplication) ERn+2 → ERn,ERm+2 → ERm	—	—	— [11]	— [11]	— [11]	—	4	
CLRMAC*4	CLRMAC	—									2	0 → MACH,MACL	—	—	—	—	—	—	2 [12]	
LDMAC*4	LDMAC ERs,MACH	L		2								ERs → MACH	—	—	—	—	—	—	2 [12]	
	LDMAC ERs,MACL	L		2								ERs → MACL	—	—	—	—	—	—	2 [12]	
STMAC*4	STMAC MACH,ERd	L		2								MACH → ERd	—	—	↕	↕	↕	—	1 [12]	
	STMAC MACL,ERd	L		2								MACL → ERd	—	—	↕	↕	↕	—	1 [12]	

- Logical instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0	—	1	
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0	—	1	
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0	—	2	
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0	—	1	
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0	—	3	
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0	—	2	
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0	—	1	
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0	—	1	
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0	—	2	
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0	—	1	
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0	—	3	
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0	—	2	
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0	—	1	
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0	—	1	
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0	—	2	
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0	—	1	
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0	—	3	
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0	—	2	
NOT	NOT.B Rd	B		2								¬Rd8→Rd8	—	—	↕	↕	0	—	1	
	NOT.W Rd	W		2								¬Rd16→Rd16	—	—	↕	↕	0	—	1	
	NOT.L ERd	L		2								¬ERd32→ERd32	—	—	↕	↕	0	—	1	

- Shift instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	1	
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1	
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	1	
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1	
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	1	
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1	
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	SHAR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	SHAR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	SHAR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1	
	SHLL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1	
	SHLL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1	
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
SHLR	SHLR.B Rd	B		2									—	—	0	↕	0	↕	1	
	SHLR.B #2,Rd	B		2									—	—	0	↕	0	↕	1	
	SHLR.W Rd	W		2									—	—	0	↕	0	↕	1	
	SHLR.W #2,Rd	W		2									—	—	0	↕	0	↕	1	
	SHLR.L ERd	L		2									—	—	0	↕	0	↕	1	
	SHLR.L #2,ERd	L		2									—	—	0	↕	0	↕	1	
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTXL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTXR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	

- Bit manipulation instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
BSET	BSET #xx:3,Rd	B		2								(#xx:3 of Rd8)←1	—	—	—	—	—	—	1	
	BSET #xx:3,@ERd	B			4							(#xx:3 of @ERd)←1	—	—	—	—	—	—	4	
	BSET #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←1	—	—	—	—	—	—	4	
	BSET #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←1	—	—	—	—	—	—	5	
	BSET #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←1	—	—	—	—	—	—	6	
	BSET Rn,Rd	B		2								(Rn8 of Rd8)←1	—	—	—	—	—	—	1	
	BSET Rn,@ERd	B			4							(Rn8 of @ERd)←1	—	—	—	—	—	—	4	
	BSET Rn,@aa:8	B						4				(Rn8 of @aa:8)←1	—	—	—	—	—	—	4	
	BSET Rn,@aa:16	B						6				(Rn8 of @aa:16)←1	—	—	—	—	—	—	5	
	BSET Rn,@aa:32	B						8				(Rn8 of @aa:32)←1	—	—	—	—	—	—	6	
BCLR	BCLR #xx:3,Rd	B		2								(#xx:3 of Rd8)←0	—	—	—	—	—	—	1	
	BCLR #xx:3,@ERd	B			4							(#xx:3 of @ERd)←0	—	—	—	—	—	—	4	
	BCLR #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←0	—	—	—	—	—	—	4	
	BCLR #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←0	—	—	—	—	—	—	5	
	BCLR #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←0	—	—	—	—	—	—	6	
	BCLR Rn,Rd	B		2								(Rn8 of Rd8)←0	—	—	—	—	—	—	1	
	BCLR Rn,@ERd	B			4							(Rn8 of @ERd)←0	—	—	—	—	—	—	4	
	BCLR Rn,@aa:8	B						4				(Rn8 of @aa:8)←0	—	—	—	—	—	—	4	
	BCLR Rn,@aa:16	B						6				(Rn8 of @aa:16)←0	—	—	—	—	—	—	5	
	BCLR Rn,@aa:32	B						8				(Rn8 of @aa:32)←0	—	—	—	—	—	—	6	
BNOT	BNOT #xx:3,Rd	B		2								(#xx:3 of Rd8)← [¬(#xx:3 of Rd8)]	—	—	—	—	—	—	1	
	BNOT #xx:3,@ERd	B			4							(#xx:3 of @ERd)← [¬(#xx:3 of @ERd)]	—	—	—	—	—	—	4	
	BNOT #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)← [¬(#xx:3 of @aa:8)]	—	—	—	—	—	—	4	
	BNOT #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←[¬(#xx:3 of @aa:16)]	—	—	—	—	—	—	5	
	BNOT #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)← [¬(#xx:3 of @aa:32)]	—	—	—	—	—	—	6	
	BNOT Rn,Rd	B		2								(Rn8 of Rd8)← [¬(Rn8 of Rd8)]	—	—	—	—	—	—	1	
	BNOT Rn,@ERd	B			4							(Rn8 of @ERd)← [¬(Rn8 of @ERd)]	—	—	—	—	—	—	4	
	BNOT Rn,@aa:8	B						4				(Rn8 of @aa:8)← [¬(Rn8 of @aa:8)]	—	—	—	—	—	—	4	
	BNOT Rn,@aa:16	B						6				(Rn8 of @aa:16)← [¬(Rn8 of @aa:16)]	—	—	—	—	—	—	5	
	BNOT Rn,@aa:32	B						8				(Rn8 of @aa:32)← [¬(Rn8 of @aa:32)]	—	—	—	—	—	—	6	
BTST	BTST #xx:3,Rd	B		2								¬(#xx:3 of Rd8)→Z	—	—	—	↕	—	—	1	
	BTST #xx:3,@ERd	B			4							¬(#xx:3 of @ERd)→Z	—	—	—	↕	—	—	3	
	BTST #xx:3,@aa:8	B						4				¬(#xx:3 of @aa:8)→Z	—	—	—	↕	—	—	3	
	BTST #xx:3,@aa:16	B						6				¬(#xx:3 of @aa:16)→Z	—	—	—	↕	—	—	4	
	BTST #xx:3,@aa:32	B						8				¬(#xx:3 of @aa:32)→Z	—	—	—	↕	—	—	5	
	BTST Rn,Rd	B		2								¬(Rn8 of Rd8)→Z	—	—	—	↕	—	—	1	
	BTST Rn,@ERd	B			4							¬(Rn8 of @ERd)→Z	—	—	—	↕	—	—	3	
	BTST Rn,@aa:8	B						4				¬(Rn8 of @aa:8)→Z	—	—	—	↕	—	—	3	
	BTST Rn,@aa:16	B						6				¬(Rn8 of @aa:16)→Z	—	—	—	↕	—	—	4	
	BTST Rn,@aa:32	B						8				¬(Rn8 of @aa:32)→Z	—	—	—	↕	—	—	5	
BLD	BLD #xx:3,Rd	B		2								(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BLD #xx:3,@ERd	B			4							(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BLD #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BLD #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BLD #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BILD	BILD #xx:3,Rd	B		2								¬(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BILD #xx:3,@ERd	B			4							¬(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BILD #xx:3,@aa:8	B						4				¬(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BILD #xx:3,@aa:16	B						6				¬(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BILD #xx:3,@aa:32	B						8				¬(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BST	BST #xx:3,Rd	B		2								C→(#xx:3 of Rd8)	—	—	—	—	—	—	1	
	BST #xx:3,@ERd	B			4							C→(#xx:3 of @ERd)	—	—	—	—	—	—	4	
	BST #xx:3,@aa:8	B						4				C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4	
	BST #xx:3,@aa:16	B						6				C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5	
	BST #xx:3,@aa:32	B						8				C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6	
BIST	BIST #xx:3,Rd	B		2								¬C→(#xx:3 of Rd8)	—	—	—	—	—	—	1	
	BIST #xx:3,@ERd	B			4							¬C→(#xx:3 of @ERd)	—	—	—	—	—	—	4	
	BIST #xx:3,@aa:8	B						4				¬C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4	
	BIST #xx:3,@aa:16	B						6				¬C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5	
	BIST #xx:3,@aa:32	B						8				¬C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6	
BAND	BAND #xx:3,Rd	B		2								C∧(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BAND #xx:3,@ERd	B			4							C∧(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BAND #xx:3,@aa:8	B						4				C∧(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BAND #xx:3,@aa:16	B						6				C∧(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BAND #xx:3,@aa:32	B						8				C∧(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	

- Bit manipulation instructions (cont)

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States[1]	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal[2]	Advanced
BIAND	BIAND #xx:3,Rd	B		2								C∧[¬(#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIAND #xx:3,@ERd	B			4							C∧[¬(#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIAND #xx:3,@aa:8	B						4				C∧[¬(#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIAND #xx:3,@aa:16	B						6				C∧[¬(#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIAND #xx:3,@aa:32	B						8				C∧[¬(#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	
BOR	BOR #xx:3,Rd	B		2								C∨(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BOR #xx:3,@ERd	B			4							C∨(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BOR #xx:3,@aa:8	B						4				C∨(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BOR #xx:3,@aa:16	B						6				C∨(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BOR #xx:3,@aa:32	B						8				C∨(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BIOR	BIOR #xx:3,Rd	B		2								C∨[¬(#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIOR #xx:3,@ERd	B			4							C∨[¬(#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIOR #xx:3,@aa:8	B						4				C∨[¬(#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIOR #xx:3,@aa:16	B						6				C∨[¬(#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIOR #xx:3,@aa:32	B						8				C∨[¬(#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	
BXOR	BXOR #xx:3,Rd	B		2								C⊕(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BXOR #xx:3,@ERd	B			4							C⊕(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BXOR #xx:3,@aa:8	B						4				C⊕(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BXOR #xx:3,@aa:16	B						6				C⊕(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BXOR #xx:3,@aa:32	B						8				C⊕(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BIXOR	BIXOR #xx:3,Rd	B		2								C⊕[¬(#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIXOR #xx:3,@ERd	B			4							C⊕[¬(#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIXOR #xx:3,@aa:8	B						4				C⊕[¬(#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIXOR #xx:3,@aa:16	B						6				C⊕[¬(#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIXOR #xx:3,@aa:32	B						8				C⊕[¬(#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	

- Branch instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation		Condition Code						Number of States[1]	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		Branch Condition	I	H	N	Z	V	C	Normal[2]	Advanced
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then PC←PC+d else next;	Always	—	—	—	—	—	—	2	
	BRA d:16(BT d:16)	—							4					—	—	—	—	—	—	3	
	BRN d:8(BF d:8)	—							2				Never	—	—	—	—	—	—	2	
	BRN d:16(BF d:16)	—							4					—	—	—	—	—	—	3	
	BHI d:8	—							2				C∨Z=0	—	—	—	—	—	—	2	
	BHI d:16	—							4					—	—	—	—	—	—	3	
	BLS d:8	—							2				C∨Z=1	—	—	—	—	—	—	2	
	BLS d:16	—							4					—	—	—	—	—	—	3	
	BCC d:8(BHS d:8)	—							2				C=0	—	—	—	—	—	—	2	
	BCC d:16(BHS d:16)	—							4					—	—	—	—	—	—	3	
	BCS d:8(BLO d:8)	—							2				C=1	—	—	—	—	—	—	2	
	BCS d:16(BLO d:16)	—							4					—	—	—	—	—	—	3	
	BNE d:8	—							2				Z=0	—	—	—	—	—	—	2	
	BNE d:16	—							4					—	—	—	—	—	—	3	
	BEQ d:8	—							2				Z=1	—	—	—	—	—	—	2	
	BEQ d:16	—							4					—	—	—	—	—	—	3	
	BVC d:8	—							2				V=0	—	—	—	—	—	—	2	
	BVC d:16	—							4					—	—	—	—	—	—	3	
	BVS d:8	—							2				V=1	—	—	—	—	—	—	2	
	BVS d:16	—							4					—	—	—	—	—	—	3	
	BPL d:8	—							2				N=0	—	—	—	—	—	—	2	
	BPL d:16	—							4					—	—	—	—	—	—	3	
	BMI d:8	—							2				N=1	—	—	—	—	—	—	2	
	BMI d:16	—							4					—	—	—	—	—	—	3	
	BGE d:8	—							2				N⊕V=0	—	—	—	—	—	—	2	
	BGE d:16	—							4					—	—	—	—	—	—	3	
	BLT d:8	—							2				N⊕V=1	—	—	—	—	—	—	2	
	BLT d:16	—							4					—	—	—	—	—	—	3	
	BGT d:8	—							2				Z∨(N⊕V)=0	—	—	—	—	—	—	2	
	BGT d:16	—							4					—	—	—	—	—	—	3	
	BLE d:8	—							2				Z∨(N⊕V)=1	—	—	—	—	—	—	2	
	BLE d:16	—							4					—	—	—	—	—	—	3	
JMP	JMP @ERn	—			2							PC←ERn		—	—	—	—	—	—	2	
	JMP @aa:24	—						4				PC←aa:24		—	—	—	—	—	—	3	
	JMP @@aa:8	—								2		PC←@aa:8		—	—	—	—	—	—	4	5
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8		—	—	—	—	—	—	3	4
	BSR d:16	—							4			PC→@-SP,PC←PC+d:16		—	—	—	—	—	—	4	5
JSR	JSR @ERn	—			2							PC→@-SP,PC←ERn		—	—	—	—	—	—	3	4
	JSR @aa:24	—						4				PC→@-SP,PC←aa:24		—	—	—	—	—	—	4	5
	JSR @@aa:8	—								2		PC→@-SP,PC←@aa:8		—	—	—	—	—	—	4	6
RTS	RTS	—									2	PC←@SP+		—	—	—	—	—	—	4	5

- System control instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—	—	—	7 [9]	8 [9]
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕	5 [9]	
SLEEP	SLEEP	—										Transition to the power-down state	—	—	—	—	—	—	2	
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—	—	—	2	
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕	↕	↕	1	
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—	—	—	1	
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕	↕	↕	3	
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—	—	—	3	
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕	4	
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—	—	—	4	
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕	6	
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—	—	—	6	
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕	4	
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—	—	—	4	
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕	↕	↕	4	
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—	—	—	4	
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕	↕	↕	5	
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—	—	—	5	
STC	STC CCR,Rd	B		2								CCR→Rd8	—	—	—	—	—	—	1	
	STC EXR,Rd	B		2								EXR→Rd8	—	—	—	—	—	—	1	
	STC CCR,@ERd	W			4							CCR→@ERd	—	—	—	—	—	—	3	
	STC EXR,@ERd	W			4							EXR→@ERd	—	—	—	—	—	—	3	
	STC CCR,@(d:16,ERd)	W				6						CCR→@(d:16,ERd)	—	—	—	—	—	—	4	
	STC EXR,@(d:16,ERd)	W				6						EXR→@(d:16,ERd)	—	—	—	—	—	—	4	
	STC CCR,@(d:32,ERd)	W				10						CCR→@(d:32,ERd)	—	—	—	—	—	—	6	
	STC EXR,@(d:32,ERd)	W				10						EXR→@(d:32,ERd)	—	—	—	—	—	—	6	
	STC CCR,@-ERd	W					4					ERd32-2→ERd32,CCR→@ERd	—	—	—	—	—	—	4	
	STC EXR,@-ERd	W					4					ERd32-2→ERd32,EXR→@ERd	—	—	—	—	—	—	4	
	STC CCR,@aa:16	W						6				CCR→@aa:16	—	—	—	—	—	—	4	
	STC EXR,@aa:16	W						6				EXR→@aa:16	—	—	—	—	—	—	4	
	STC CCR,@aa:32	W						8				CCR→@aa:32	—	—	—	—	—	—	5	
	STC EXR,@aa:32	W						8				EXR→@aa:32	—	—	—	—	—	—	5	
ANDC	ANDC #xx:8,CCR	B	2									CCR∧#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	ANDC #xx:8,EXR	B	4									EXR∧#xx:8→EXR	—	—	—	—	—	—	2	
ORC	ORC #xx:8,CCR	B	2									CCR∨#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	ORC #xx:8,EXR	B	4									EXR∨#xx:8→EXR	—	—	—	—	—	—	2	
XORC	XORC #xx:8,CCR	B	2									CCR⊕#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	XORC #xx:8,EXR	B	4									EXR⊕#xx:8→EXR	—	—	—	—	—	—	2	
NOP	NOP	—									2	PC←PC+2	—	—	—	—	—	—	1	

- Block transfer instructions

Mnemonic		Operand Size	Addressing Mode/Instruction Length (Bytes)									Operation	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal*2	Advanced
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n*5	
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n*5	

Notes: *1: The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.
*2: Some products do not support normal mode. Refer to the hardware manual for the relevant product.
*3: Figures in parentheses are for the H8S/2000 CPU.
*4: Applies only to the H8S/2600 CPU.
*5: n is the initial value of R4L or R4.
[1] 7 states when the number of restored/saved registers is 2, 9 states when 3, and 11 states when 4.
[2] Cannot be used with these series.
[3] Set to 1 when there is a carry from or borrow to bit 11; otherwise cleared to 0.
[4] Set to 1 when there is a carry from or borrow to bit 27; otherwise cleared to 0.
[5] If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.
[6] One additional state is required for execution immediately after a MULXU, MULXS, or STMAC instruction. Also, a maximum of three additional states are required for execution of a MULXU instruction within three states after execution of a MAC instruction. For example, if there is a one-state instruction (such as NOP) between a MAC instruction and a MULXU instruction, the MULXU instruction will be two states longer.
[7] A maximum of two additional states are required for execution of a MULXS instruction within two states after execution of a MAC instruction. For example, if there is a one-state instruction (such as NOP) between a MAC instruction and a MULXS instruction, the MULXS instruction will be one state longer.
[8] Set to 1 if the divisor is negative: otherwise cleared to 0.
[9] Set to 1 if the divisor is zero; otherwise cleared to 0.
[10] Set to 1 if the quotient is negative; otherwise cleared to 0.
[11] The result of a MAC instruction is reflected in the flags by executing an STMAC instruction.
[12] A maximum of three additional states are required for execution of this instruction within three states after execution of a MAC instruction. For example, if there is a one-state instruction (such as NOP) between the MAC instruction and this instruction, this instruction will be two states longer.
[13] When EXR is valid, the number of states is increased by 1.

Number of States Required for Execution: The number of states shown in the instruction set table is the number of states required for execution when the op code and operand data are located in a one-cycle area on which word access is possible, such as on-chip memory. When the op code or operand data is accessed from an on-chip supporting module or an external address, the number of states increases as shown in the table below.

- Number of States per Cycle

	Access Conditions						
				External Device			
		On-Chip Supporting Module		8-Bit Bus		16-Bit Bus	
Cycle	**On-Chip Memory**	**8-Bit Bus**	**16-Bit Bus**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State Access**
Instruction fetch	1	2n	n	4	6 + 2m	2	3 + m
Branch instruction read							
Stack operation							
Byte data access		n		2	3 + m		
Word data access		2n		4	6 + 2m		
Internal operation	1	1	1	1	1	1	1

Legend

m: Number of wait states inserted in external device access

n: Number of wait states inserted into on-chip supporting module access. Refer to the hardware manual for the relevant product for the actual number.

Condition Code Notation

Symbol	Meaning
↕	Changes according operation result
*	Indeterminate (value not guaranteed)
0	Always cleared to 0
1	Always set to 1
—	Not affected by operation result

Operation Notation

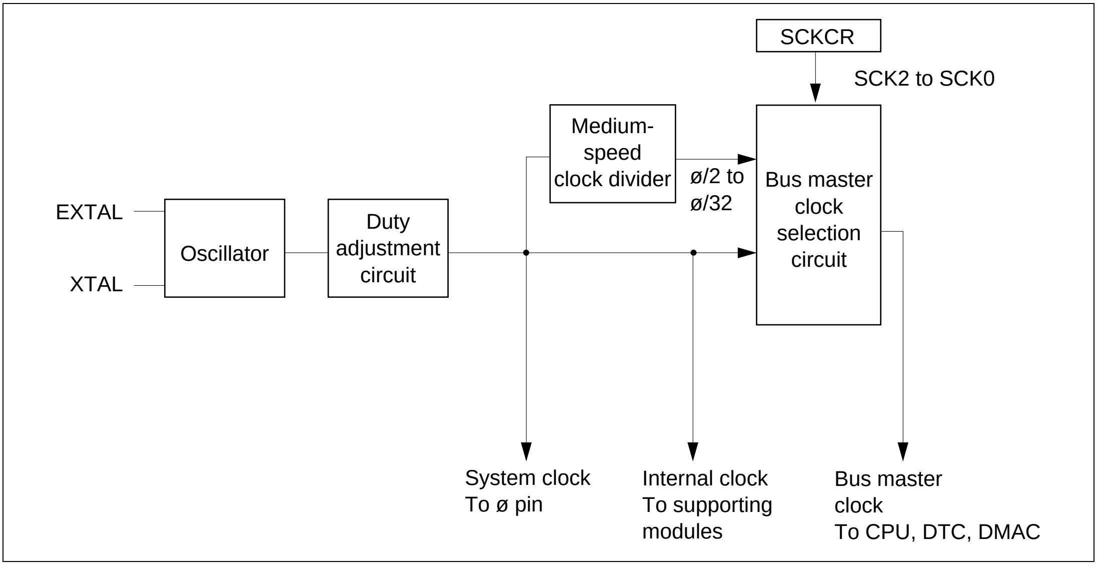
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
( ) < >	Operand contents
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

2.6 Basic Bus Timing

The CPU is driven by a system clock, denoted by the symbol ø. The period from one rising edge of ø to the next is referred to as a “state.” A bus cycle consists of one, two, or three states. Different methods are used to access on-chip memory, on-chip supporting modules, and the external address space.

Basic Clock Timing: An external clock is input to the EXTAL pin, or a crystal oscillator is connected to the EXTAL pin, to generate the system clock (ø). An external clock or crystal oscillator of the same frequency as the ø clock should be used.

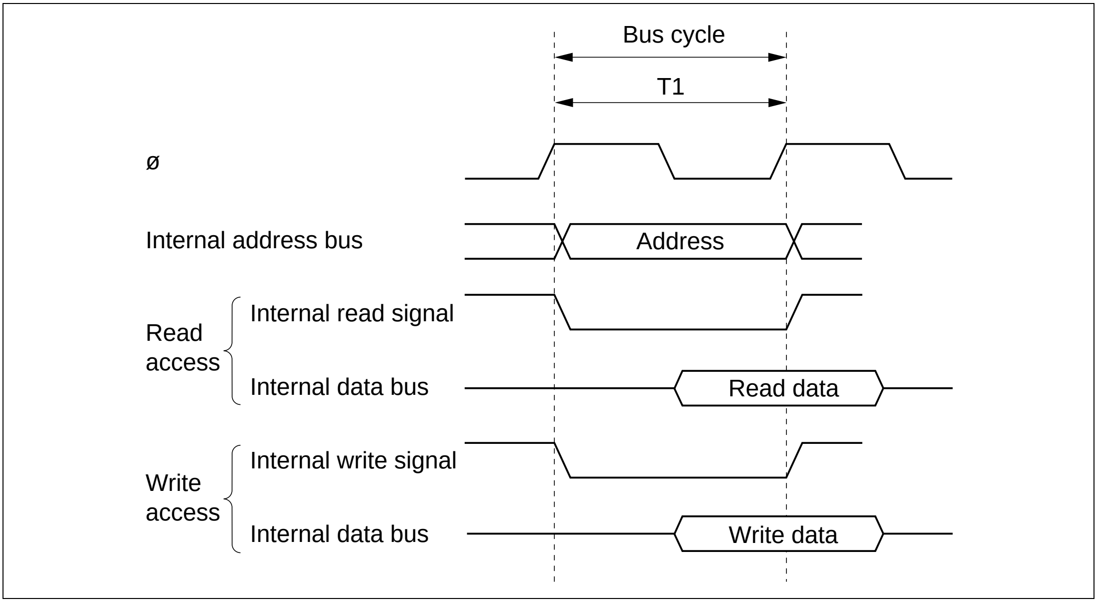


Basic Clock Timing

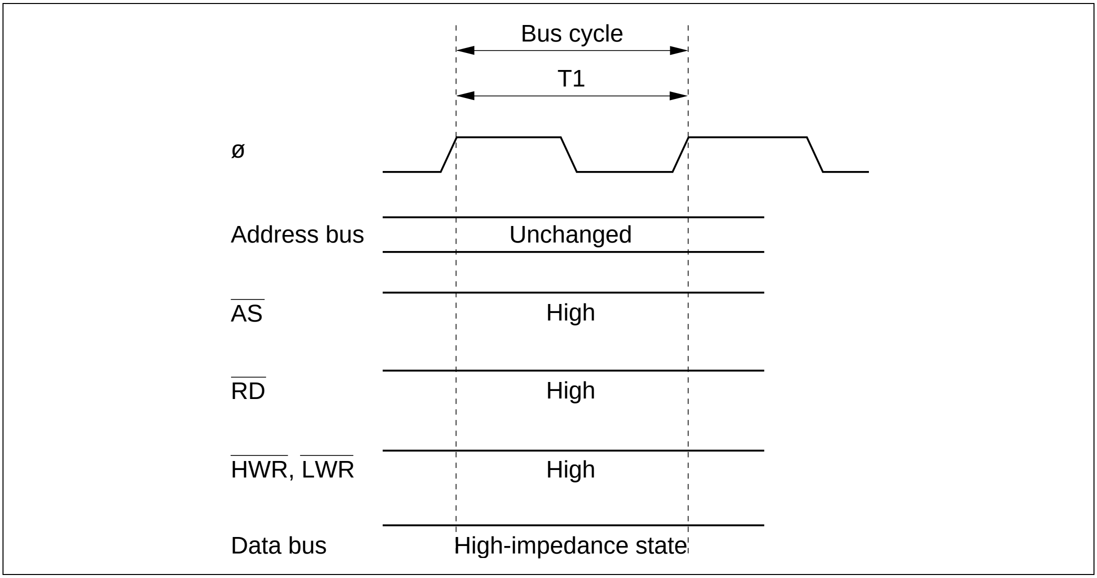
CPU Read/Write Cycles: The CPU operates on the basis of the ø clock. One ø clock cycle is called a state, and a bus cycle consists of one, two, or three states. Different access methods are used for on-chip memory, on-chip supporting modules, and external address space.

On-Chip Memory: On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word access.

- On-Chip Memory Access Cycle (One-state Access)

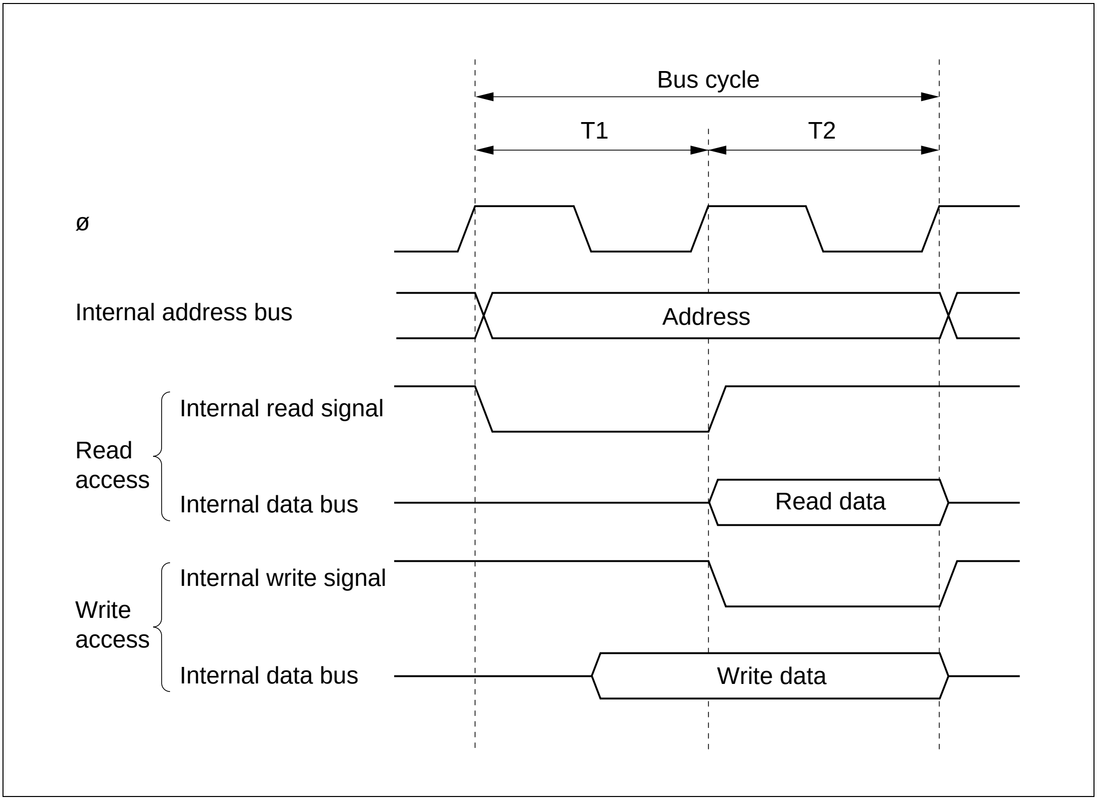


- Pin States during On-Chip Memory Access

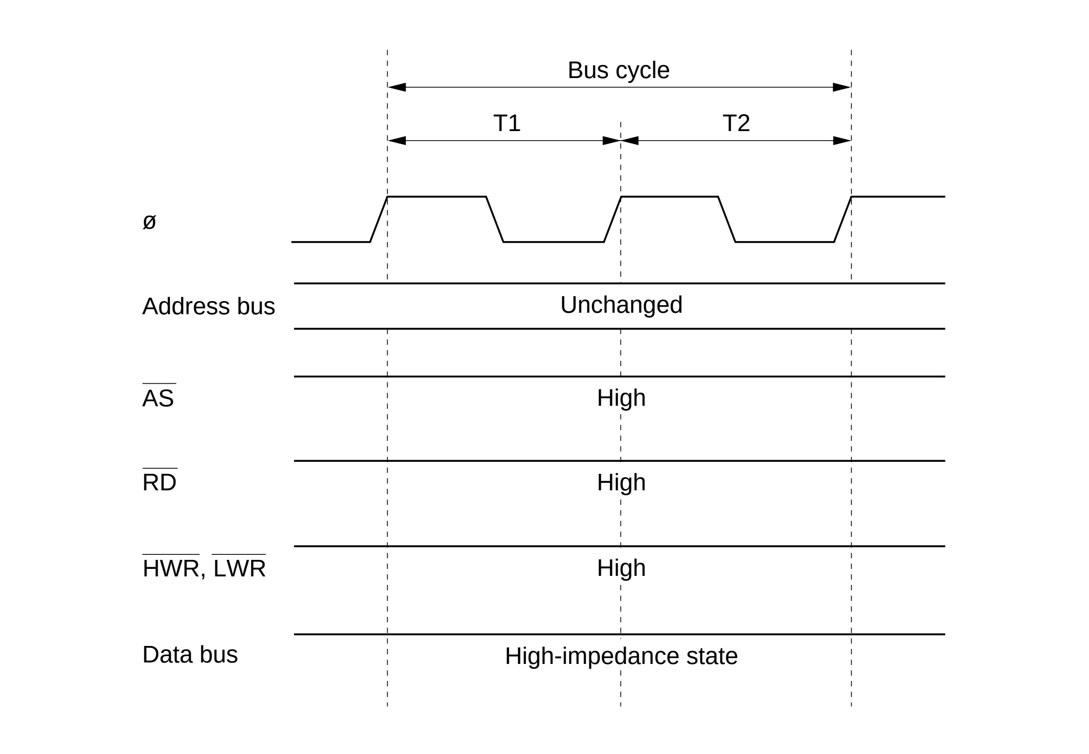


On-Chip Supporting Module Access Timing: The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed.

- On-Chip Supporting Module Access Cycle (Two-state Access)



- Pin States during On-Chip Supporting Module Access



External Address Space Access Timing: The external address space is accessed with an 8-bit or 16-bit data bus width in a two-state or three-state bus cycle. In three-state access, wait states can be inserted. For further details, refer to section 3.1, Bus Controller (BSC).

2.7 Processing States

The H8S/2000 CPU has five processing states: the reset state, program execution state, exception-handling state, bus-released state, and power-down state.

Reset State: State in which the CPU and all on-chip supporting modules are initialized and halted

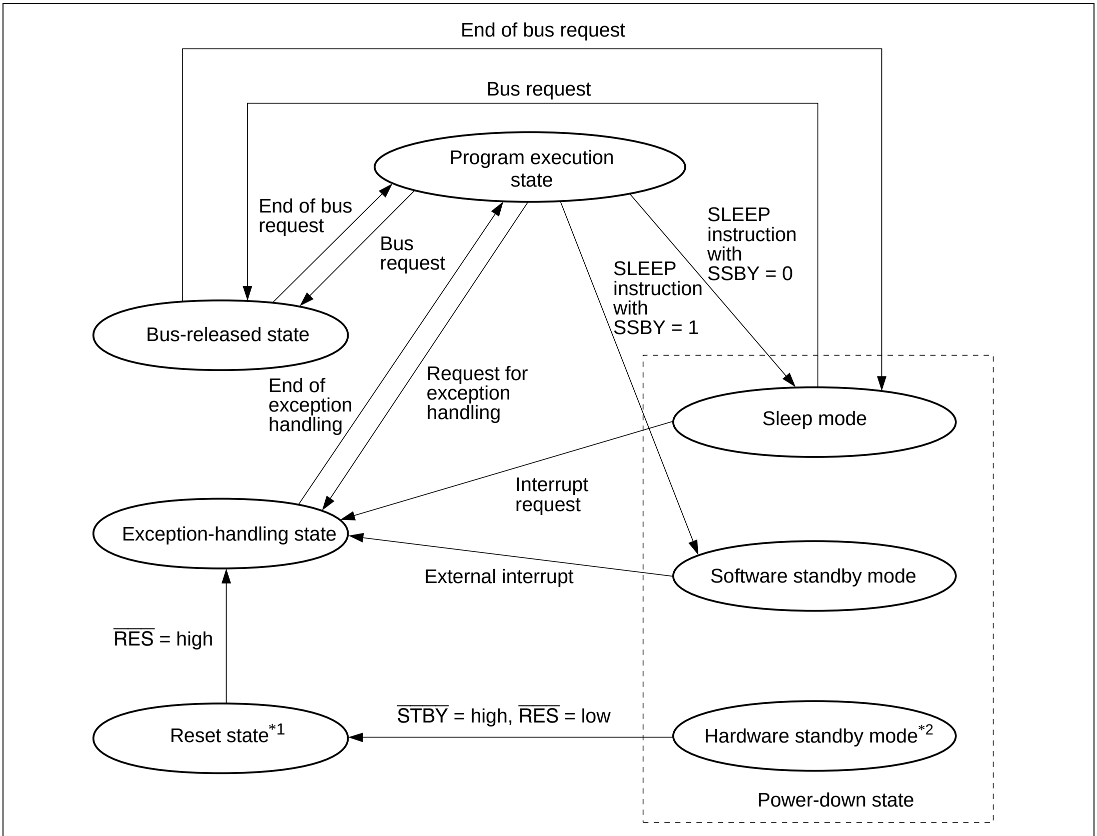
Program Execution State: State in which the CPU executes the program sequentially

Exception-Handling State: Transient state in which exception handling is executed as the result of a reset, interrupt, or trap instruction exception handling source

Bus-Released State: State in which the external bus is released in response to a bus request signal from a bus master other than the CPU

Power-Down State: State in which CPU operation is stopped, and power consumption is kept low (sleep mode, software standby mode, hardware standby mode). The power-down state also includes medium-speed mode and module stop mode.

State Transition Diagram



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{\text{RES}}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.

2. From any state, a transition to hardware standby mode occurs when $\overline{\text{STBY}}$ goes low.

2.8 Exception Handling

H8S/2000 CPU exception handling is initiated by a reset, a trap instruction, or an interrupt. A priority system is provided for exception handling, and simultaneously generated exceptions are handled in order of priority.

Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the $\overline{RES}$ pin, or when the watchdog timer overflows. The chip enters the power-on reset state when the NMI is high, or the manual reset state*4 when the NMI pin is low.
↑	Trace*1	Valid only in interrupt control modes 2 and 3. Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1.
↑	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued*2
Low	Trap instruction*3 (TRAPA)	Always accepted in the program execution state. Started by execution of a trap (TRAPA) instruction.

Notes: 1. Trace exception handling is not executed after execution of an RTE instruction.
2. Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.
3. Trap instruction exception handling is always accepted, in the program execution state.
4. Manual reset is not supported in the H8S/2357 (F-ZTAT and mask ROM versions) or in the H8S/2352.
Please contact your Hitachi sales representative for information on manual reset in the H8S/2345 Series.

Exception Handling Operation: Exceptions originate from various sources. Trap instructions and interrupts are handled as follows:

1. The program counter (PC), condition code register (CCR), and extend register (EXR) are pushed onto the stack.
2. The interrupt mask bits are updated. The T bit is cleared to 0.
3. A vector address corresponding to the exception source is generated, and program execution starts from that address.

For a reset exception, steps 2 and 3 above are carried out.

Exception Vector Table

Exception Source		Vector Number	Vector Address*[1]	
			Normal Mode*[3]	Advanced Mode
Reset		0	H'0000 to H'0001	H'0000 to H'0003
Reserved for system use		1	H'0002 to H'0003	H'0004 to H'0007
		2	H'0004 to H'0006	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0009	H'0010 to H'0013
		5	H'000A to H'000B	H'0014 to H'0017
Direct transition		6	H'000C to H'000D	H'0018 to H'001B
External interrupt	NMI	7	H'000E to H'000F	H'001C to H'001F
Trap instruction (4 sources)		8	H'0010 to H'0011	H'0020 to H'0023
		9	H'0012 to H'0013	H'0024 to H'0027
		10	H'0014 to H'0015	H'0028 to H'002B
		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
	IRQ6	22	H'002C to H'002D	H'0058 to H'005B
	IRQ7	23	H'002E to H'002F	H'005C to H'005F
Internal interrupt*[2]		24 to 103	H'0030 to H'0031 to H'00B6to H'00B7	H'0060 to H'0063 to H'019C to H'019F

Notes: 1. Lower 16 bits of the address.

2. Refer to the hardware manual for the relevant product for details of the internal interrupt vector table.

3. Some products do not support normal mode. Refer to the hardware manual for the relevant product.

2.9 Interrupts

Interrupts are controlled by the interrupt controller. Interrupt sources comprise external interrupts from the external pins and internal interrupts from on-chip supporting modules. The interrupt control modes and the number of interrupt sources depend on the product. A separate vector number is assigned to each interrupt.

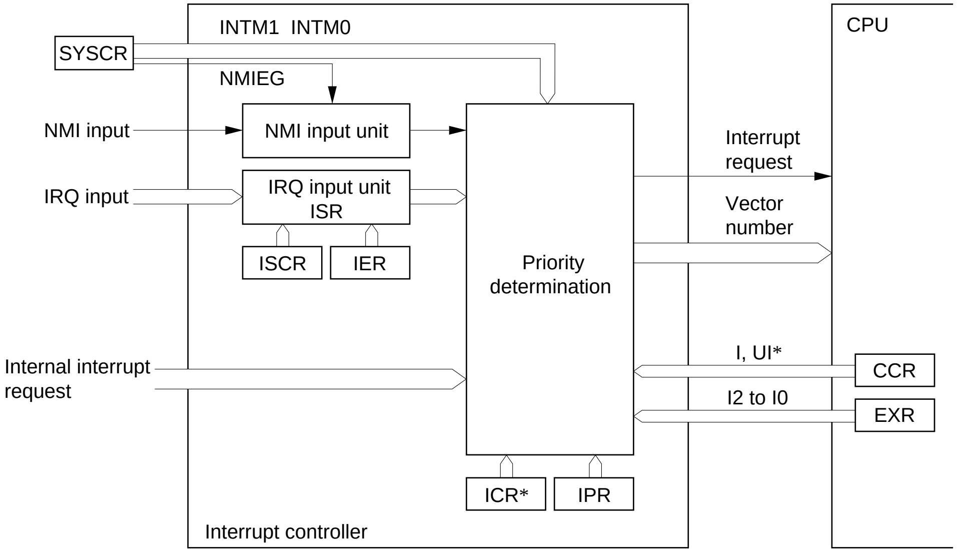
Item	H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Internal interrupts	52	42	47	52	43
External interrupts	9	9	9	9	9
Interrupt control modes	0, 1, 2, 3	0, 2	0, 2	0, 2	0, 2

Interrupt Control: Any of four interrupt control modes (depending on the product series) can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).

The interrupt controller controls interrupts on the basis of the control mode set by the INTM1 and INTM0 bits, the interrupt priorities set by interrupt control register (ICR) and interrupt priority register (IPR), and the masking conditions set by the I and UI bits in CCR and bits I2 to I0 in EXR.

NMI is the highest-priority interrupt, and is always accepted.

Block Diagram of Interrupt Controller



Legend:

ISCR	: IRQ sense control register
IER	: IRQ enable register
ISR	: IRQ status register
IPR	: Interrupt priority register
ICR	: Interrupt control register
SYSCR	: System control register

Note: * Only applies to the H8S/2655 Series.

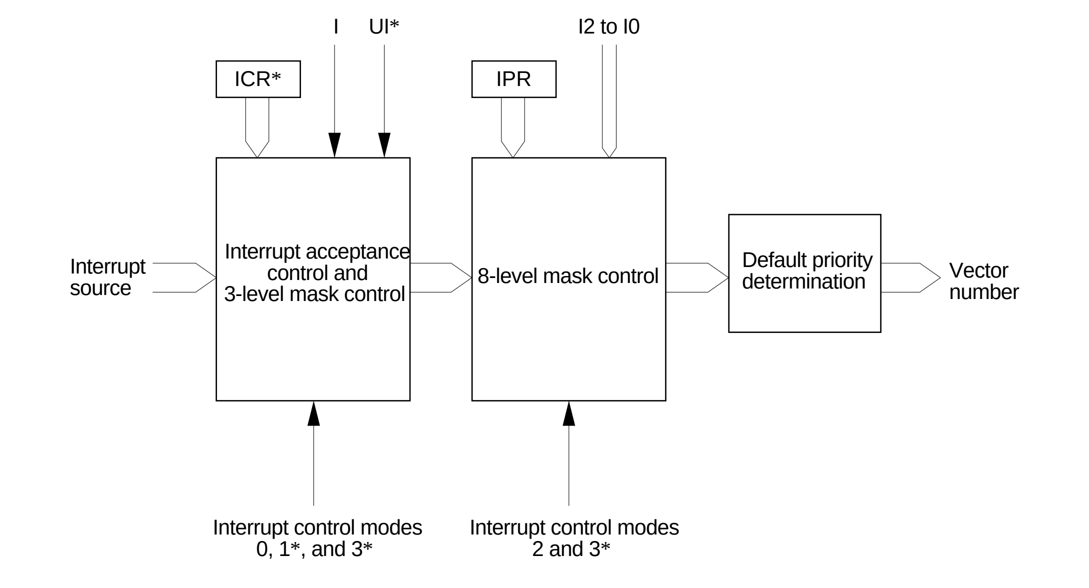
Interrupt Control Modes

Interrupt	SYSCR		Priority Setting	Interrupt	
Control Mode	**INTM1**	**INTM0**	**Registers**	**Mask Bits**	**Description**
0	0	0	ICR*1	I	Interrupt mask control is performed by the I bit. Priority can be set with ICR.*1
1*2		1	ICR	I, UI	3-level interrupt mask control is performed by the I and UI bits. Priority can be set with ICR.*1
2	1	0	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels can be set with IPR.
3*2		1	ICR*1, IPR	I, UI, I2 to I0	Control is performed by a combination of interrupt masking set by the I and UI bits and priority setting by ICR, based on 8-level interrupt mask control performed by bits I2 to I0 and 8-level priority setting by IPR.

Notes: 1. ICR only applies to the H8S/2655 Series.

2. Setting prohibited in the H8S/2350 Series, H8S/2355 Series, H8S/2357 Series, and H8S/2345 Series.

- Block Diagram of Interrupt Control Operation


I
UI*
I2 to I0
ICR*
IPR
Interrupt source
Interrupt acceptance control and 3-level mask control
8-level mask control
Default priority determination
Vector number
Interrupt control modes 0, 1*, and 3*
Interrupt control modes 2 and 3*

Note: * Only applies to the H8S/2655 Series.

Interrupt Control Mode 0: Enabling and disabling of IRQ interrupts and on-chip supporting module interrupts can be set by means of the I bit in CCR. Control level setting can be performed with ICR*. Interrupts are enabled when the I bit is cleared to 0, and disabled when set to 1.

Control level 1 interrupt sources have higher priority*.

Interrupt Control Mode 1*: Three-level masking can be implemented for IRQ interrupts and on-chip supporting module interrupts by means of the I and UI bits in CCR, and ICR.

- Control level 0 interrupt requests are enabled when the I bit is cleared to 0, and disabled when set to 1.
- Control level 1 interrupt requests are enabled when the I bit or UI bit is cleared to 0, and disabled when both the I bit and the UI bit are set to 1.

Interrupt Control Mode 2: Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Interrupt Control Mode 3*: Control of IRQ interrupts and on-chip supporting module interrupts is performed by a combination of interrupt masking set by the I and UI bits and control level setting by ICR, based on 8-level interrupt mask control performed by comparing the interrupt mask level in the CPU’s EXR (bits I2 to I0) and the priority set in IPR.

- Control level 0 interrupt requests are enabled when the I bit is cleared to 0, and disabled when set to 1.
- Control level 1 interrupt requests are enabled when the I bit or UI bit is cleared to 0, and disabled when both the I bit and the UI bit are set to 1.
- Eight-level priority control is performed when the I bit is cleared to 0.

Note: * Only applies to the H8S/2655 Series.

2.10 Operating Modes

The operating mode allows the selection of initial settings for the CPU operating mode, enabling/disabling of on-chip ROM, and bus width, by setting the mode pins (MD_2 to MD_0).

The supported operating modes depend on the product. Refer to the hardware manual for the relevant product.

2.10.1 Normal Modes (Modes 1 to 3)

Mode 1 (on-chip ROM disabled expanded mode): The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is disabled, and 8-bit bus mode is set, immediately after a reset.

Mode 2 (on-chip ROM enabled expanded mode): The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is enabled, and 8-bit bus mode is set, immediately after a reset.

Mode 3 (single-chip mode): The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is enabled, but external addresses cannot be accessed.
All I/O ports are available for use as input-output ports.

2.10.2 Advanced Modes (Modes 4 to 7)

Mode 4 (on-chip ROM disabled expanded mode*): The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.
The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, note that if 8-bit access is designated by the bus controller for all areas, the bus mode switches to 8 bits.

Mode 5 (on-chip ROM disabled expanded mode*): The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.
The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if at least one area is designated for 16-bit access by the bus controller, the bus mode switches to 16 bits and port E becomes a data bus.

Mode 6 (on-chip ROM enabled expanded mode): The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled.
The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if at least one area is designated for 16-bit access by the bus controller, the bus mode switches to 16 bits and port E becomes a data bus.

Note: * In the H8S/2345 Series, the upper address signals (A_{23} to A_{20}) are not output immediately after a reset in mode 4 or 5. To output the upper address signals (A_{23} to A_{20}), the corresponding P1DDR (port 1 data direction register) bits must be set to 1 beforehand.

Mode 7 (single-chip mode): The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed.
All I/O ports are available for use as input-output ports.

Modes 8 and 9 (F-ZTAT™ versions only): Modes 8 and 9 are not supported and must not be set.

2.10.3 Boot Modes (Modes 10 and 11)

Mode 10 (F-ZTAT™ versions only: advanced on-chip ROM enabled expanded mode): This is a flash memory boot mode.
The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas.
Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced on-chip ROM enabled expanded mode.

Mode 11 (F-ZTAT™ versions only: advanced single-chip mode): This is a flash memory boot mode.
The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed.
All I/O ports are available for use as input-output ports.
Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced single-chip mode.

Modes 12 and 13 (F-ZTAT™ versions only): Modes 12 and 13 are not supported and must not be set.

2.10.4 User Program Modes (Modes 14 and 15)

Mode 14 (F-ZTAT™ versions only: advanced on-chip ROM enabled expanded mode): This is a flash memory user program mode.
The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The initial bus mode after a reset is 8 bits, with 8-bit access to all areas.
Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced on-chip ROM enabled expanded mode.

Mode 15 (F-ZTAT™ versions only: advanced single-chip mode): This is a flash memory user program mode.
The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed.
All I/O ports are available for use as input-output ports.
Except for the fact that flash memory programming and erasing can be performed, operation in this mode is the same as in advanced single-chip mode.

MCU Operating Modes in Each Product

MCU Operating Mode	H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
0	—	—	—	—	—
1	○	○	○	—	○*2
2	○	○*1	○	—	○*2
3	○	○*1	○	—	○*2
4	○	○	○	○	○
5	○	○	○	○	○
6	○	○*1	○	○	○
7	○	○*1	○	○	○
8	—	—	—	—	—
9	—	—	—	—	—
10	—	—	—	○*3	○*3
11	—	—	—	○*3	○*3
12	—	—	—	—	—
13	—	—	—	—	—
14	—	—	—	○*3	○*3
15	—	—	—	○*3	○*3

Notes: 1. Can only be set in the H8S/2351.

2. Can only be set in the H8S/2345 ZTAT and mask ROM versions and the H8S/2343.

3. Can only be set in the F-ZTAT version.

○: Available

Section 3 Supporting Modules

3.1 Bus Controller (BSC)

These Series has a built-in bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

The bus controller also has a bus arbitration function, and controls the operation of the internal bus masters: the CPU, DMA controller (DMAC), and data transfer controller (DTC). (Depends on the product; refer to the hardware manual for the relevant product for details.)

Comparison of Product Series Functions

Item	H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Area partitioning (bytes)	128 k/2M	2M	2M	2M	2M
DRAM interface	Available (3 kinds)	Available (1 kind)	—	Available (1 kind)	—
Pseudo-SRAM interface	Available	—	—	—	—
Burst ROM interface	Available	Available	Available	Available	Available
Write buffer function	Available	Available	—	Available	—
Chip select signals	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
Bus requests output	Available	Available	—	Available	—

Features

- Manages external address space in area units
  - — In advanced mode, manages the external space as 8 areas of 128-kbytes/2-Mbytes
  - — In normal mode, manages the external space as a single area
  - — Bus specifications can be set independently for each area
  - — DRAM/PSRAM/burst ROM interfaces can be set

- Basic bus interface
  - — Chip select ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$) can be output for areas 0 to 7 (only $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ in the H8S/2345 Series)
  - — 8-bit access or 16-bit access can be selected for each area
  - — 2-state access or 3-state access can be selected for each area
  - — Program wait states can be inserted for each area

- DRAM interface
  - DRAM interface can be set for areas 2 to 5 (in advanced mode)

- Pseudo-SRAM (PSRAM) direct interface*
  - PSRAM interface can be set for areas 2 to 5 (in advanced mode)

- Burst ROM interface
  - Burst ROM interface can be set for area 0

- Idle cycle insertion
  - An idle cycle can be inserted in an external read cycle between different areas
  - An idle cycle can be inserted in an external write cycle immediately after an external read cycle

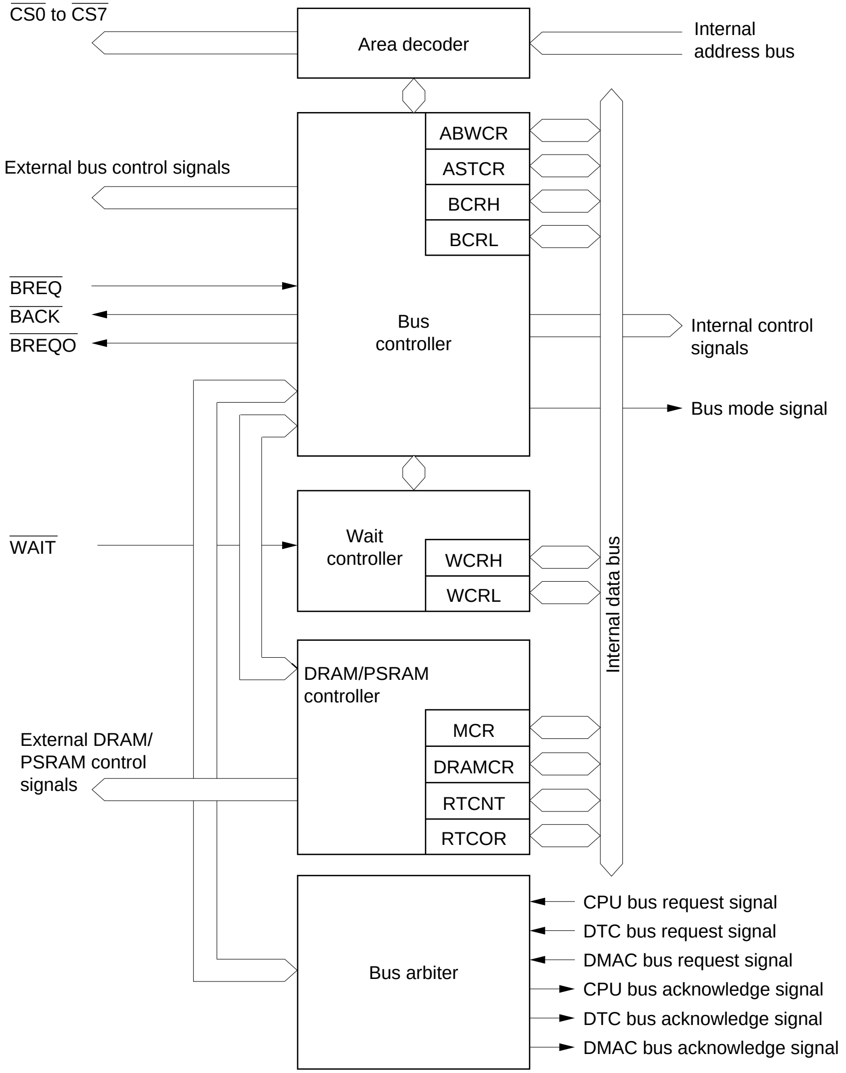
- Write buffer functions
  - External write cycle and internal access can be executed in parallel
  - DMAC single-address mode and internal access can be executed in parallel

- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC, and DTC

- Other features
  - Refresh counter (refresh timer) can be used as an interval timer
  - External bus release function

Note: * Only applies to the H8S/2655 Series.

Block Diagram (Example of H8S/2655 Series)



Legend:

ABWCR	: Bus width control register	WCRL	: Wait control register L
ASTCR	: Access state control register	MCR	: Memory control register
BCRH	: Bus control register H	DRAMCR	: DRAM control register
BCRL	: Bus control register L	RTCNT	: Refresh timer counter
WCRH	: Wait control register H	RTCOR	: Refresh time constant register

Bus Specifications: The external space bus specifications consist of three elements: bus width, number of access states, and number of program wait states.
The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

The following bus specifications can be set by means of the bus controller control register.

Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

Memory Interfaces: Memory interfaces comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on; a DRAM interface that allows direct connection of DRAM; a PSRAM interface that allows direct connection of PSRAM; and a burst ROM interface that allows direct connection of burst ROM. The interface can be selected independently for each area.

Notes: 1. The DRAM interface applies only to the H8S/2655 Series, H8S/2350 Series, and H8S/2357 Series.
2. The pseudo-SRAM interface applies only to the H8S/2655 Series.

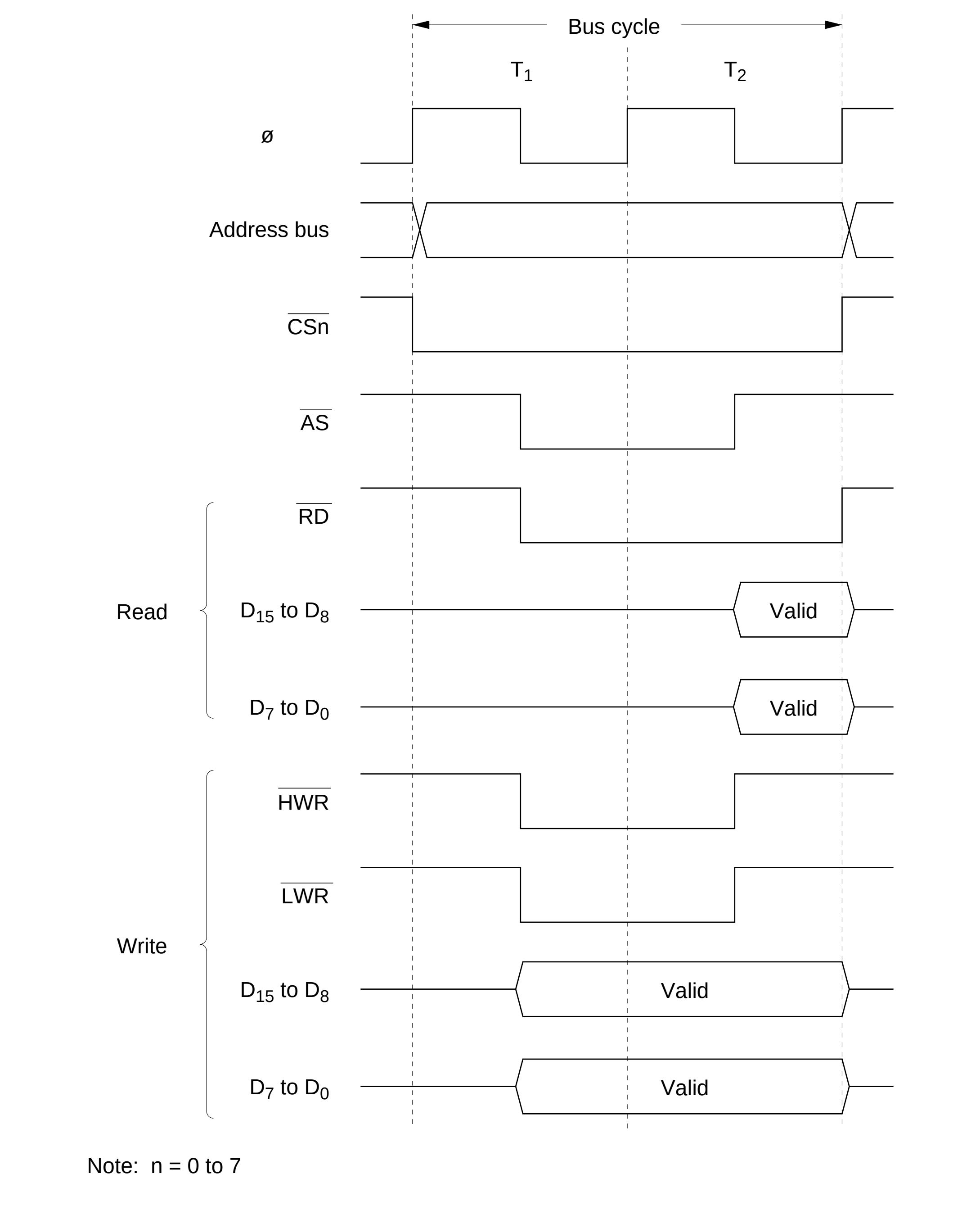
3.1.1 Basic Bus Interface

This interface can be designated for areas 0 to 7. When external address space is accessed, the chip select signal ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$) for each area can be output (only $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ in the H8S/2345 Series).

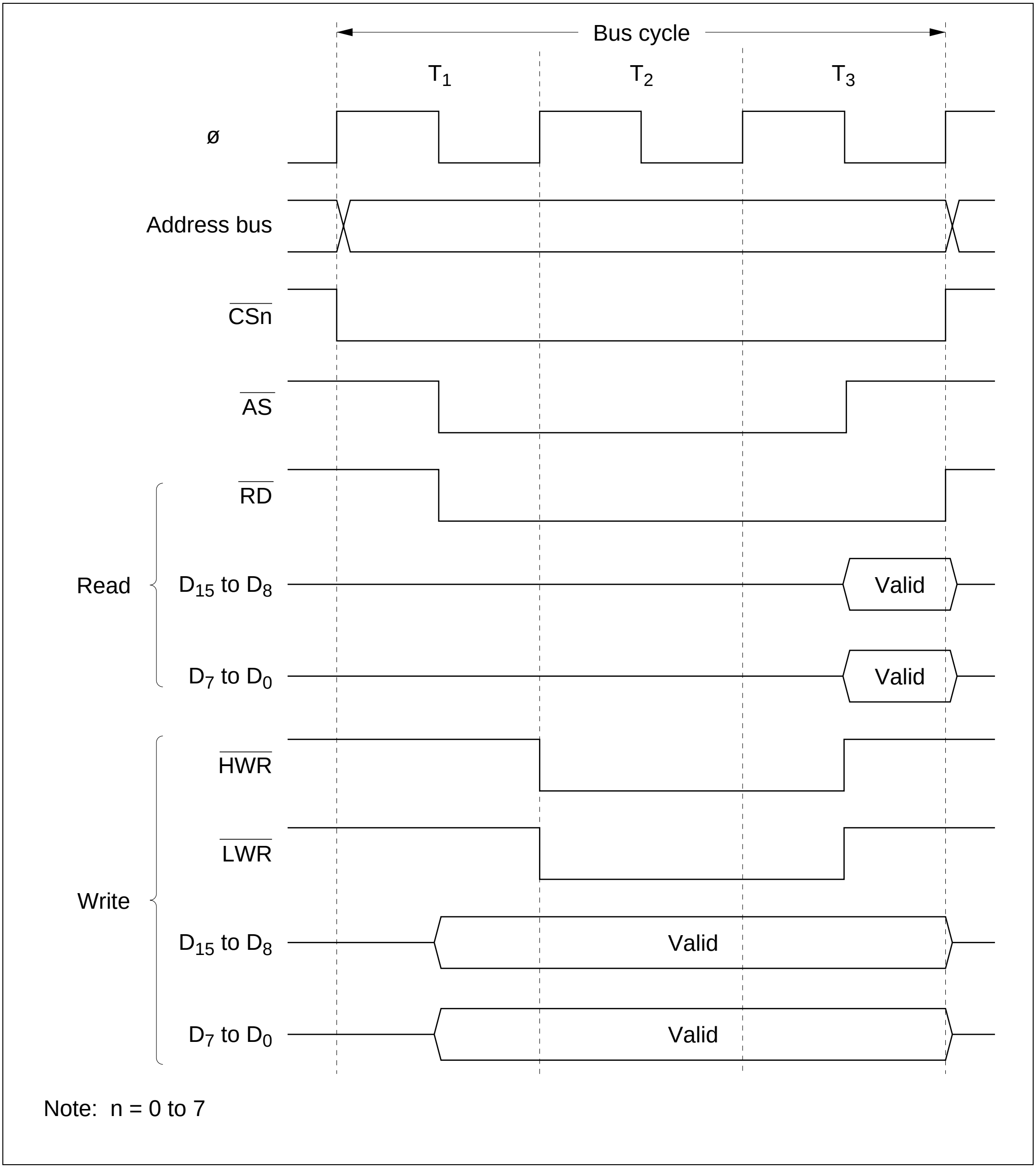
In 3-state access space, 0 to 3 program wait states or a pin wait by means of the $\overline{\text{WAIT}}$ pin can be inserted.

After a reset, all areas are designated as basic bus interface, 3-state access space (the bus width is determined by the MCU operating mode).

Basic Bus Timing



Basic Bus Timing (Word Access to 16-Bit 2-State Access Space)



Basic Bus Timing (Word Access to 16-Bit 3-State Access Space)

3.1.2 DRAM Interface

In advanced mode, external space areas 2 to 5 can be designated as DRAM space, and DRAM interfacing performed. With the DRAM interface, DRAM can be directly connected to the H8S/2000 Series. Selectable DRAM space settings are: one area (area 2); two areas (areas 2 and 3); and four areas (areas 2 to 5). In an area designated as DRAM space, the $\overline{\text{CS}}$ pin functions as the $\overline{\text{RAS}}$ pin.

Features

- 2/4/8-Mbyte or 128*/256*/512*-kbyte DRAM space can be set

- Address multiplexing
  - — Row address and column address are multiplexed.
  - — Selection of 8, 9, or 10 bits as the row address shift size

- Basic timing
  - — 4-state basic timing
  - — Wait state insertion possible

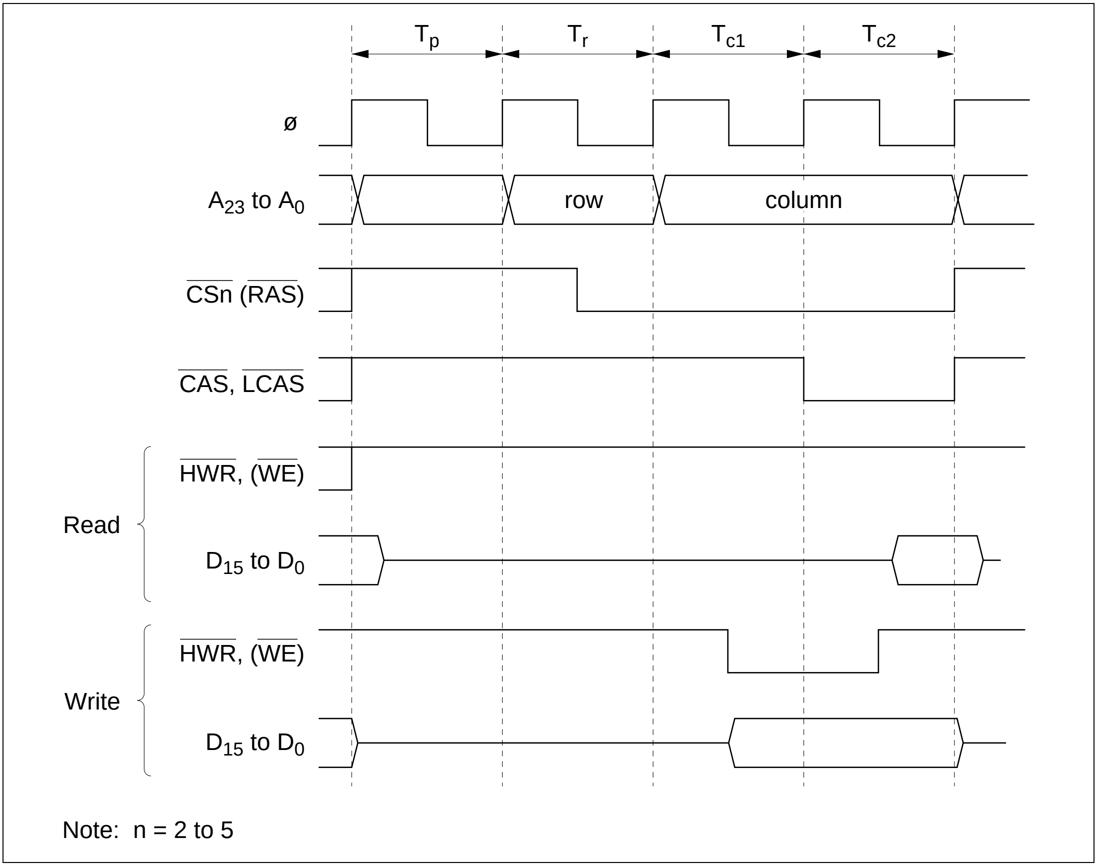
- DRAM interface
  - — Selection of 2-CAS system (LCASS = 1)*, 2-CAS system (LCASS = 0), or 2-WE system* for control signals required for byte access, according to the kind of DRAM

- Burst operation
  - — Fast page mode

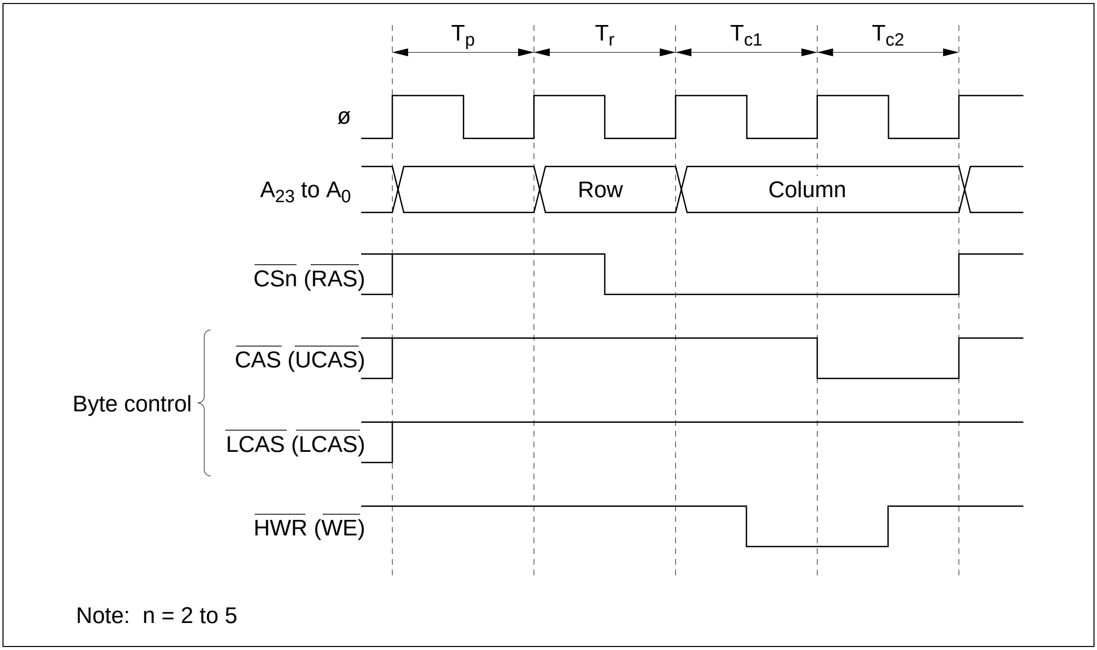
- Refresh control
  - — Selection of CAS-before-RAS refreshing or self-refreshing
  - — Can be used as interval timer

Note: * Can only be set in the H8S/2655 Series.

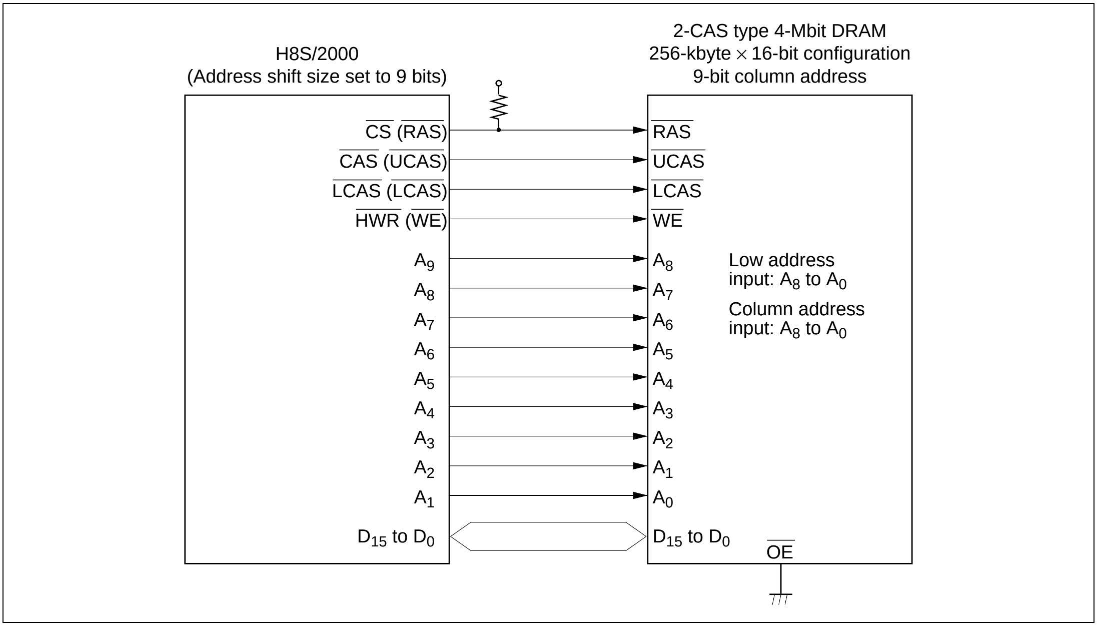
DRAM Basic Timing


T_p
T_r
T_{c1}
T_{c2}
ø
A_{23} to A_0
row
column
$\overline{CSn}$ ($\overline{RAS}$)
$\overline{CAS}$, $\overline{LCAS}$
Read
$\overline{HWR}$, ($\overline{WE}$)
D_{15} to D_0
Write
$\overline{HWR}$, ($\overline{WE}$)
D_{15} to D_0
Note: n = 2 to 5

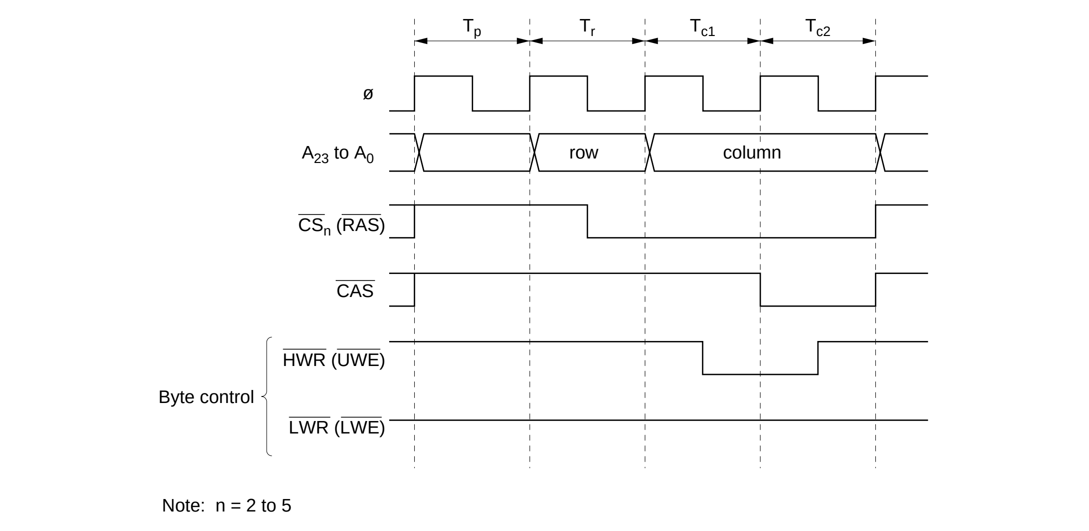
Basic Access Timing (2-CAS System (LCASS = 0))


T_p
T_r
T_{c1}
T_{c2}
ø
A_{23} to A_0
Row
Column
$\overline{CSn}$ ($\overline{RAS}$)
Byte control
$\overline{CAS}$ ($\overline{UCAS}$)
$\overline{LCAS}$ ($\overline{LCAS}$)
$\overline{HWR}$ ($\overline{WE}$)
Note: n = 2 to 5

2-CAS System Control Timing (LCASS = 0) (Upper Byte Write Access)

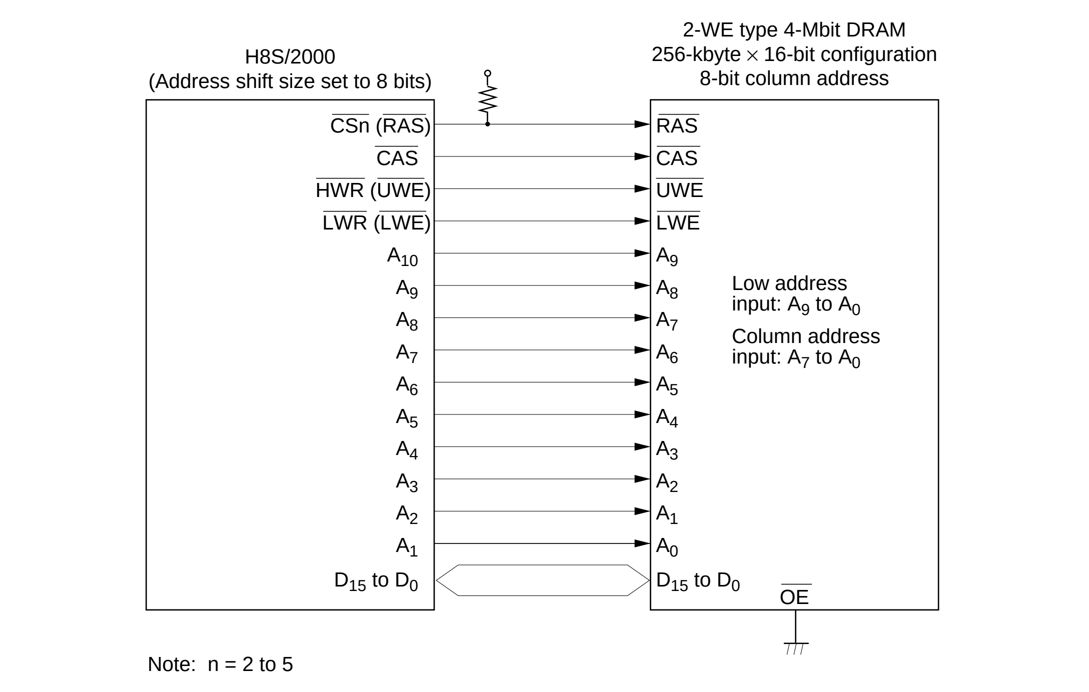

H8S/2000
(Address shift size set to 9 bits)
2-CAS type 4-Mbit DRAM
256-kbyte × 16-bit configuration
9-bit column address
$\overline{CS}$ ($\overline{RAS}$)
$\overline{CAS}$ ($\overline{UCAS}$)
$\overline{LCAS}$ ($\overline{LCAS}$)
$\overline{HWR}$ ($\overline{WE}$)
A_9
A_8
A_7
A_6
A_5
A_4
A_3
A_2
A_1
D_{15} to D_0
$\overline{RAS}$
$\overline{UCAS}$
$\overline{LCAS}$
$\overline{WE}$
A_8
A_7
A_6
A_5
A_4
A_3
A_2
A_1
A_0
D_{15} to D_0
$\overline{OE}$
Low address input: A_8 to A_0
Column address input: A_8 to A_0

Example of 2-CAS Type (LCASS = 0) DRAM Connection



T_p
T_r
T_{c1}
T_{c2}
ø
A_{23} to A_0
row
column
$\overline{CS_n}$ ($\overline{RAS}$)
$\overline{CAS}$
$\overline{HWR}$ ($\overline{UWE}$)
Byte control
$\overline{LWR}$ ($\overline{LWE}$)
Note: n = 2 to 5

2-WE System Control Timing (Upper Byte Write Access)



Example of 2-WE Type DRAM Connection

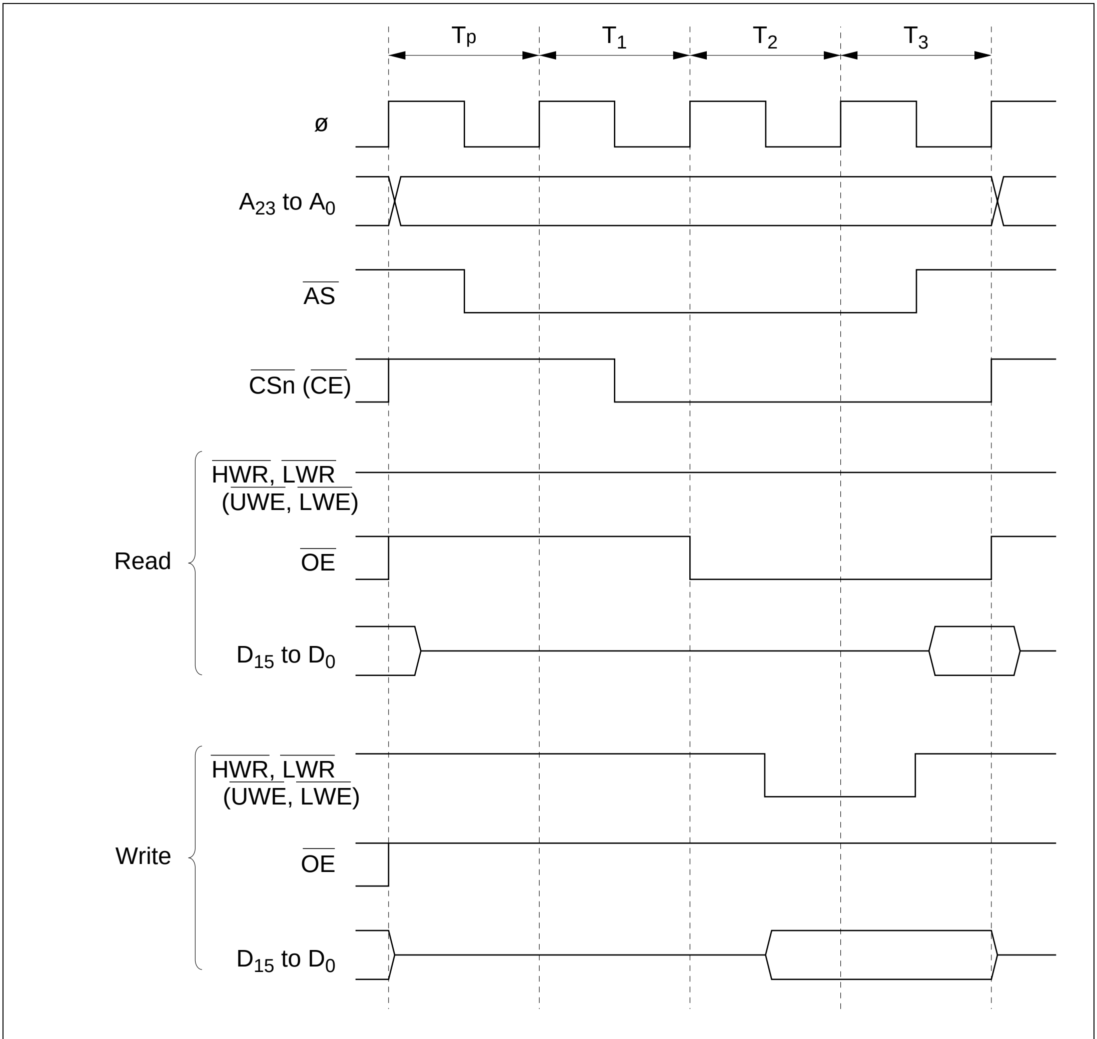
3.1.3 Pseudo-SRAM Interface (H8S/2655 Series only)

In advanced mode, external space areas 2 to 5 can be designated as pseudo-SRAM (PSRAM) space, and PSRAM interfacing performed. Selectable PSRAM space setting are: one area (area 2); two areas (areas 2 and 3); and four areas (areas 2 to 5).

Features

- 2/4/8-Mbyte or 128/256/512-kbyte PSRAM space can be set
- Signal multiplexing
  - — Multiplexing of the refresh signal ($\overline{\text{RFSH}}$) and the output enable signal ($\overline{\text{OE}}$)
- Basic timing
  - — 4-state basic timing
  - — Wait state insertion possible
- Burst operation
  - — Static column mode
- Refresh control
  - — Selection of auto-refreshing or self-refreshing

PSRAM Basic Timing

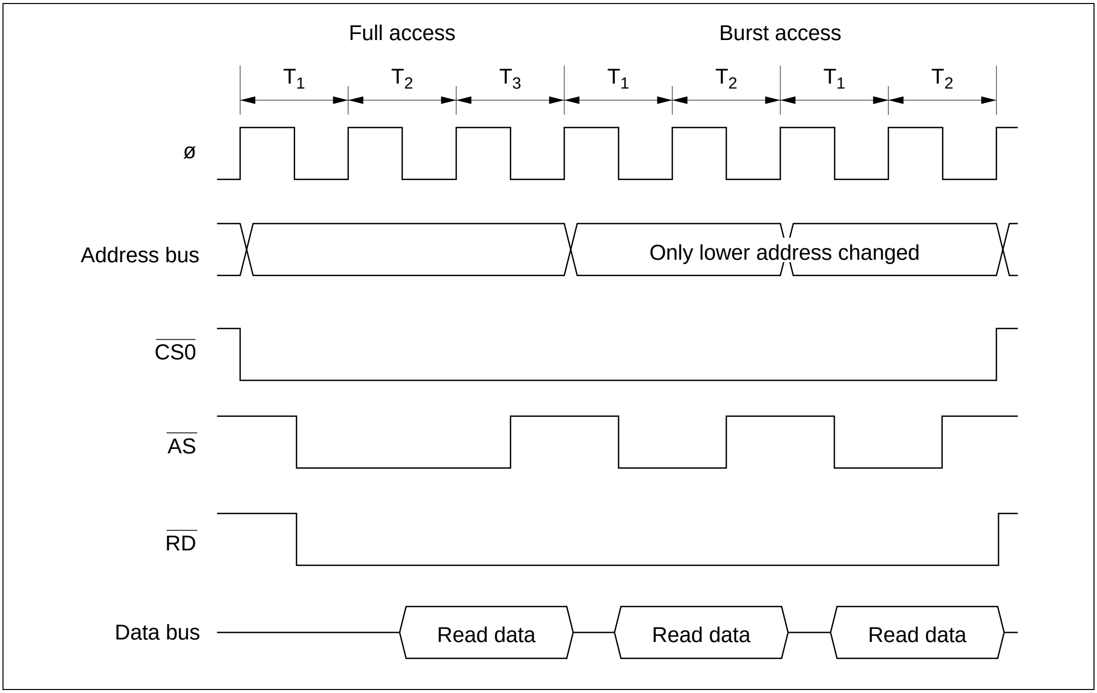


Note: n = 2 to 5

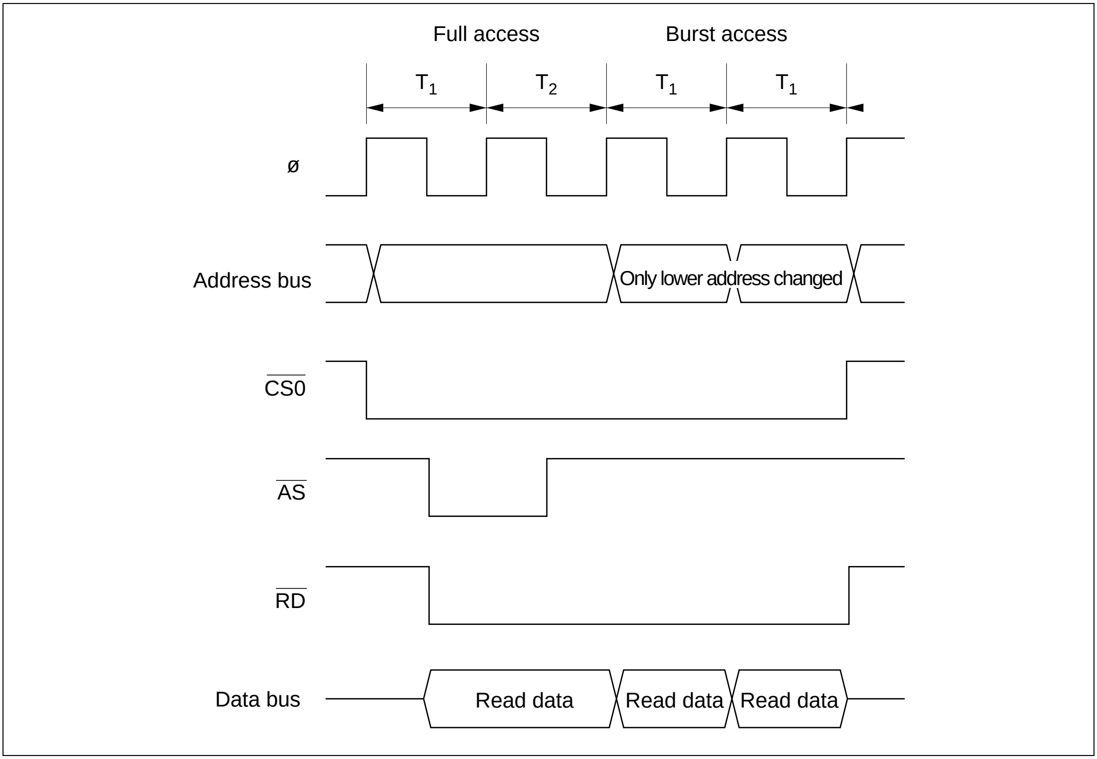
3.1.4 Burst ROM Interface

External space area 0 can be designated as burst ROM space, and burst ROM space interfacing can be performed. The burst ROM space interface enables 16-bit configuration ROM with burst access capability to be accessed at high speed.

Consecutive burst accesses of a maximum of 4 words or 8 words can be performed for CPU instruction fetches only. One or two states can be selected for burst access.



Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 1)



Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 0)

3.2 DMA Controller (DMAC)

The DMA controller (DMAC) can carry out data transfer on up to 4 channels (channels 0A, 0B, 1A, and 1B). Short address transfer can be performed on each channel independently, and full address transfer is possible by using pairs of channels.

Product Series with On-Chip DMAC

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	Available	—	Available	—

Features

- Choice of short address mode or full address mode

  Short address mode
  - Maximum of 4 channels can be used
  - Choice of dual address mode or single address mode
  - In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as16 bits
  - In single address mode, transfer source or transfer destination address only is specified as 24 bits
  - In single address mode, transfer can be performed in one bus cycle
  - Choice of sequential mode, idle mode, or repeat mode for dual address mode and single address mode

  Full address mode
  - Maximum of 2 channels can be used
  - Transfer source and transfer destination address specified as 24 bits
  - Choice of normal mode or block transfer mode

- 16-Mbyte address space can be specified directly

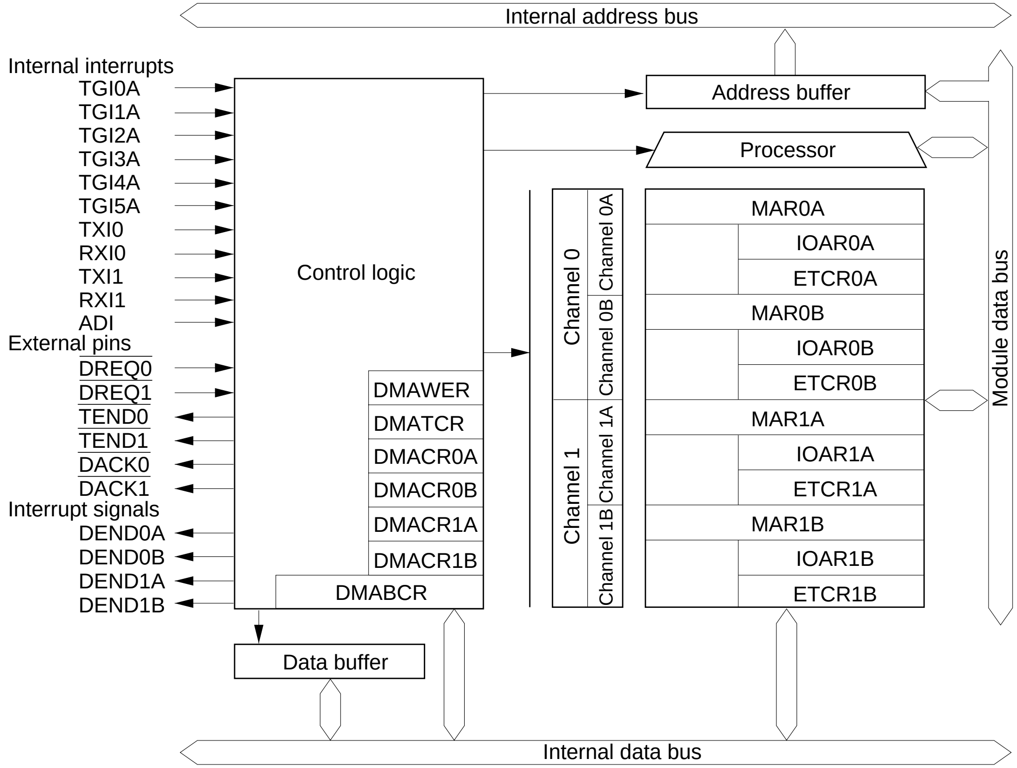
- Byte or word can be set as the transfer unit

- Activation sources: internal interrupt, external request, auto-request (depending on transfer mode)
  - Six 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  - Serial communication interface (SCI0, SCI1) transmission complete interrupt, reception complete interrupt
  - A/D converter conversion end interrupt
  - External request

— Auto-request

- Module stop mode can be set
  - — The initial setting enables DMAC registers to be accessed. DMAC operation is halted by setting module stop mode

Block Diagram



Legend:

DMAWER : DMA write enable register
DMATCR : DMA terminal control register
DMABCR : DMA band control register (for all channels)
DMACR : DMA control register
MAR : Memory address register
IOAR : I/O address register
ETCR : Executive transfer counter register

Block Diagram of DMAC

3.3 Data Transfer Controller (DTC)

The data transfer controller (DTC) can be activated by an interrupt or software to perform data transfer. Multichannel data transfer can be carried out, and a variety of transfer modes are available.

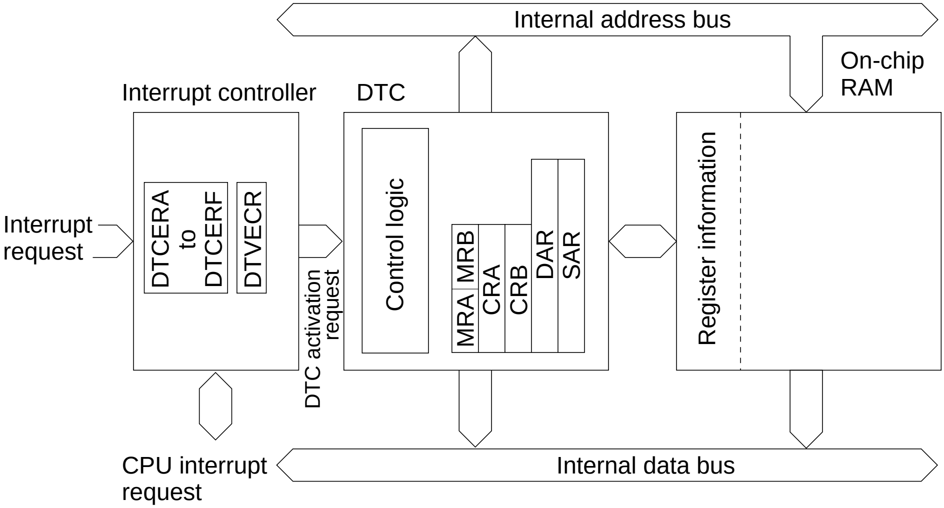
Product Series with On-Chip DTC

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	Available	Available	Available	Available

Features

- Transfer possible over any number of channels
  - — Transfer information is stored in memory
  - — One activation source can trigger a number of data transfers (chain transfer)
- Wide range of transfer modes
  - — Normal, repeat, and block transfer modes available
  - — Incrementing, decrementing, and fixing of transfer source and destination addresses can be selected
- Direct specification of 16-Mbyte address space possible
  - — Transfer source and transfer destination address specified as 24 bits
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - — An interrupt request can be issued to the CPU after one data transfer ends
  - — An interrupt request can be issued to the CPU after the specified data transfers have completely ended
- Activation by software is possible
- Module stop mode can be set
  - — The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode.

Block Diagram



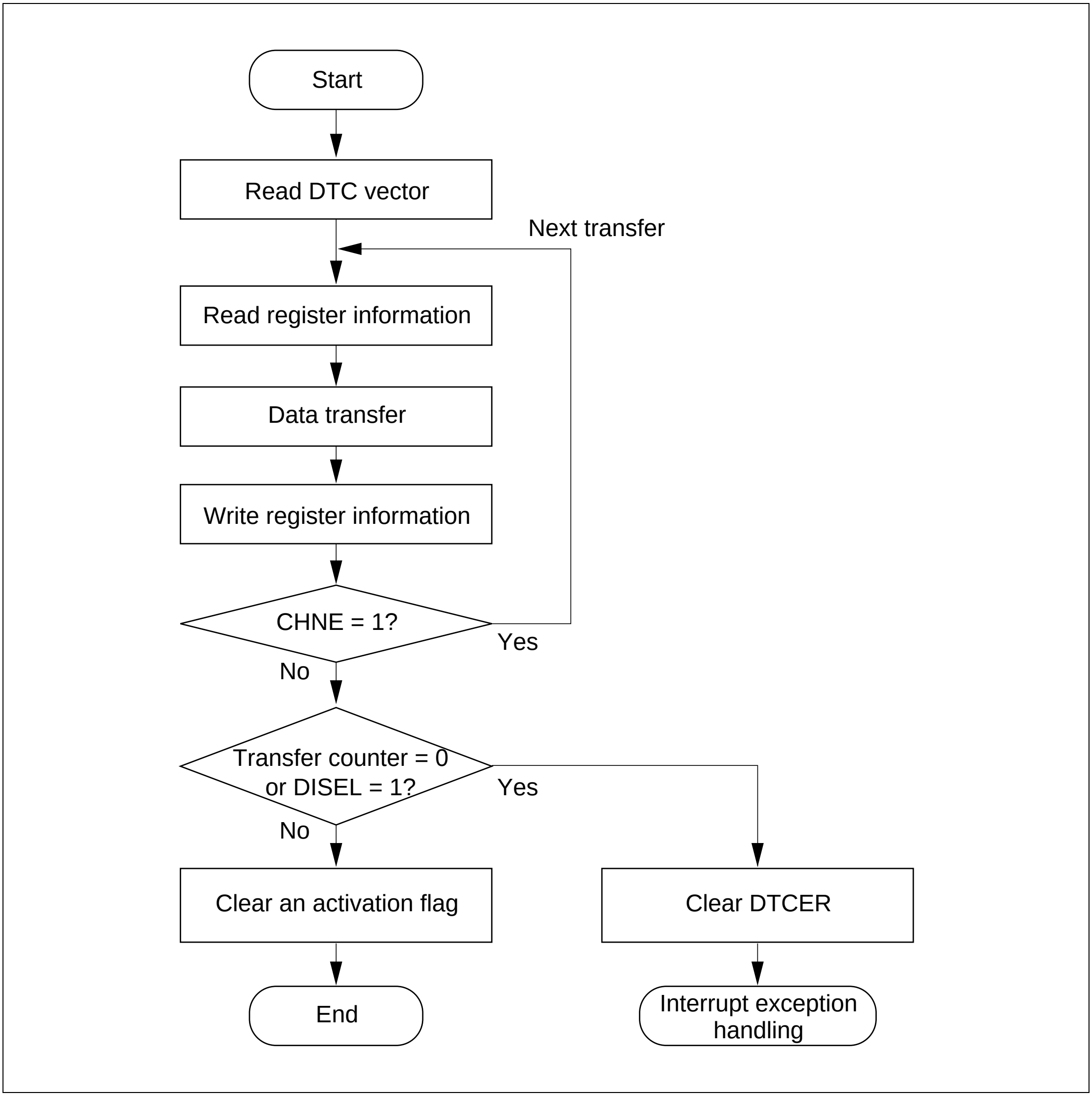
Legend:

MRA, MRB: DTC mode registers A and B
CRA, CRB: DTC transfer count registers A and B
SAR: DTC source address register
DAR: DTC destination address register
DTCERA to DTCERF: DTC enable registers A to F
DTVECR: DTC vector register

Block Diagram of DTC

Data Transfer Operation: When activated, the DTC reads register information that is already stored in memory and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. The DTC can also execute a number of transfers in response to a single activation source (chain transfer).

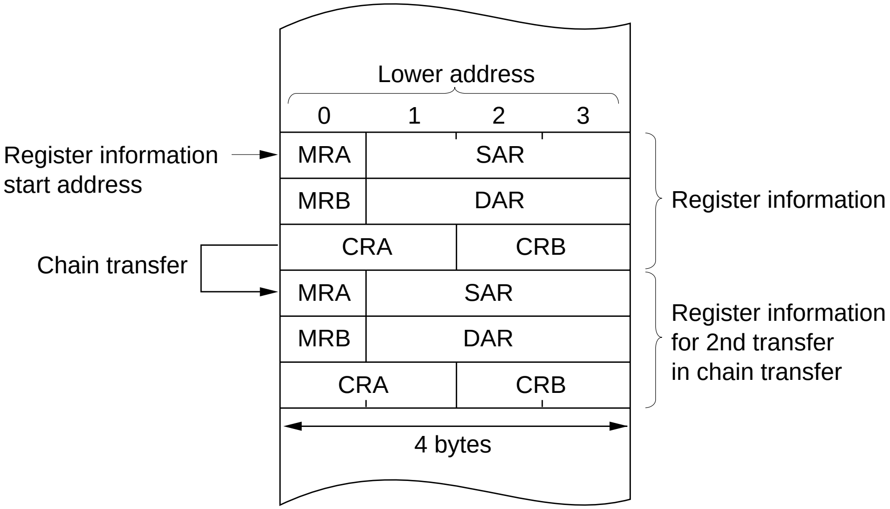
- Flowchart of DTC Operation



DTC Activation Sources: The DTC is activated by an interrupt or by a vector number write to the DTC vector register (DTVECR) by software. An interrupt request can be designated as a CPU interrupt source or a DTC activation source.
When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priority specifications have no effect. If there is more than one activation source at the same time, the DTC is activated in accordance with the default priorities.

Interrupt Sources and DTC Vector Address: The DTC vector address indicates the start address of the register information in memory. The MRA, SAR, MRB, DAR, CRA, and CRB registers are located in that order from the start address of the register information. Locate the register information in the on-chip RAM (addresses H'FFF800 to H'FFFBFF).

- Location of DTC Register Information in Address Space



Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1	DTCE*2	Priority
Write to DTVECR	Software	DTVECR	H'0400+ (DTVECR [6:0] <<1)	—	High
IRQ0	External pin	16	H'0420	DTCEA7	
IRQ1		17	H'0422	DTCEA6	
IRQ2		18	H'0424	DTCEA5	
IRQ3		19	H'0426	DTCEA4	
IRQ4		20	H'0428	DTCEA3	
IRQ5		21	H'042A	DTCEA2	
IRQ6		22	H'042C	DTCEA1	
IRQ7		23	H'042E	DTCEA0	
ADI (A/D conversion end)	A/D	28	H'0438	DTCEB6	
TGI0A (GR0A compare match/ input capture)	TPU channel 0	32	H'0440	DTCEB5	
TGI0B (GR0B compare match/ input capture)		33	H'0442	DTCEB4	
TGI0C (GR0C compare match/ input capture)		34	H'0444	DTCEB3	
TGI0D (GR0D compare match/ input capture)		35	H'0446	DTCEB2	
TGI1A (GR1A compare match/ input capture)	TPU channel 1	40	H'0450	DTCEB1	
TGI1B (GR1B compare match/ input capture)		41	H'0452	DTCEB0	
TGI2A (GR2A compare match/ input capture)	TPU channel 2	44	H'0458	DTCEC7	
TGI2B (GR2B compare match/ input capture)		45	H'045A	DTCEC6	Low

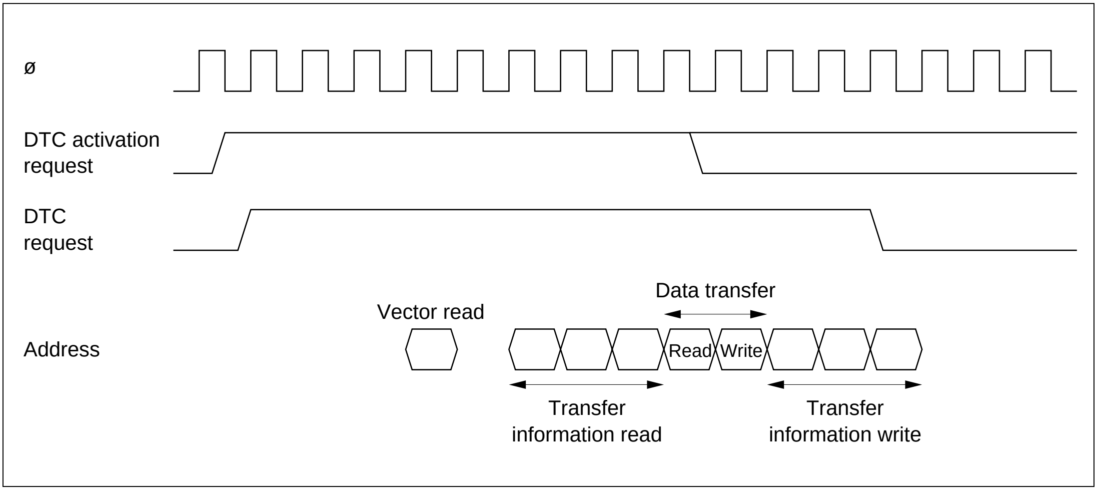
Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs (cont)

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1	DTCE*2	Priority
TGI3A (GR3A compare match/ input capture)	TPU channel 3	48	H'0460	DTCEC5	High
TGI3B (GR3B compare match/ input capture)		49	H'0462	DTCEC4	
TGI3C (GR3C compare match/ input capture)		50	H'0464	DTCEC3	
TGI3D (GR3D compare match/ input capture)		51	H'0466	DTCEC2	
TGI4A (GR4A compare match/ input capture)	TPU channel 4	56	H'0470	DTCEC1	
TGI4B (GR4B compare match/ input capture)		57	H'0472	DTCEC0	
TGI5A (GR5A compare match/ input capture)	TPU channel 5	60	H'0478	DTCED5	
TGI5B (GR5B compare match/ input capture)		61	H'047A	DTCED4	
CMIA0	8-bit timer channel 0	64	H'0480	DTCED3	
CMIB0		65	H'0482	DTCED2	
CMIA1	8-bit timer channel 1	68	H'0488	DTCED1	
CMIB1		69	H'048A	DTCED0	
DMTEND0A (DMAC transfer end 0)	DMAC	72	H'0490	DTCEE7	
DMTEND0B (DMAC transfer end 1)		73	H'0492	DTCEE6	
DMTEND1A (DMAC transfer end 2)		74	H'0494	DTCEE5	
DMTEND1B (DMAC transfer end 3)		75	H'0496	DTCEE4	
RXI0 (reception complete 0)	SCI channel 0	81	H'04A2	DTCEE3	
TXI0 (transmit data empty 0)		82	H'04A4	DTCEE2	
RXI1 (reception complete 1)	SCI channel 1	85	H'04AA	DTCEE1	
TXI1 (transmit data empty 1)		86	H'04AC	DTCEE0	
RXI2 (reception complete 2)	SCI channel 2	89	H'04B2	DTCEF7	
TXI2 (transmit data empty 2)		90	H'04B4	DTCEF6	Low

Notes: 1. Lower 16 bits of the start address.

2. The on-chip modules differ from product to product. DTCE bits with no corresponding interrupt are reserved, and must be cleared to 0.

DTC Operation Timing (Example in Normal Mode or Repeat Mode)



DTC Execution Statuses

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

N: Block size (initial setting of CRAH and CRAL)

- Number of States Required for Each Execution Status

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8		16	
Access states			1	1	2	2	2	3	2	3
Execution status	Vector read	S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write	S_J	1	—	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	2	3+m
	Word data read	S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write	S_L	1	1	2	2	2	3+m	2	3+m
	Word data write	S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation	S_M	1							

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to one, plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma\,(J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

3.4 16-Bit Timer Pulse Unit (TPU)

The 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels. The TPU can provide up to 16 kinds of pulse input/output.

The TPU can perform PWM output, pulse width measurement, and two-phase encoder processing, and can activate the data transfer controller (DTC). It can also generate an A/D converter start trigger.

Product Series with On-Chip TPU

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	Available	Available	Available	Available

Features

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5), each of which can be set independently as an output compare/input capture register
  - TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge detection
  - Counter clear operation: Counter clearing possible by compare match or input capture
  - Synchronous operation: Multiple timer counters (TCNT) can be written to simultaneously

    Simultaneous clearing by compare match and input capture possible

    Register simultaneous input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set

    Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible

- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible

- Cascaded operation
  - Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel 1 (channel 4) overflow/underflow

- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit interface

- 26 interrupt sources
  - For channels 0 and 3, four compare match/input capture dual-function interrupts and one overflow interrupt can be requested independently
  - For channels 1, 2, 4, and 5, two compare match/input capture dual-function interrupts, one overflow interrupt, and one underflow interrupt can be requested independently

- Automatic transfer of register data
  - Block transfer, 1-word data transfer, and 1-byte data transfer possible by data transfer controller (DTC) or DMA controller (DMAC)* activation

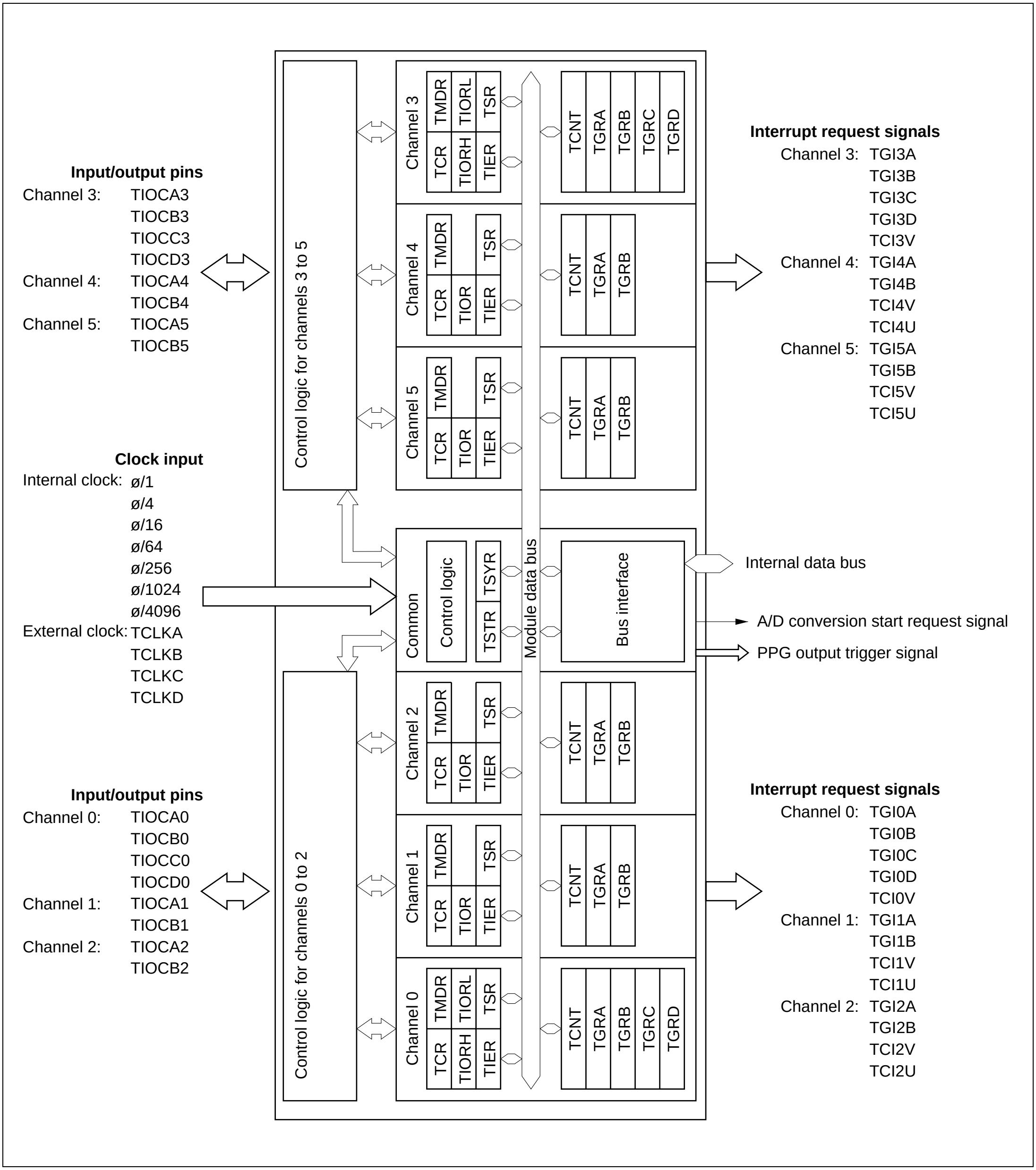
- Programmable pulse generator (PPG) output trigger can be generated*
  - Channel 0 to 3 compare match/input capture signals can be used as PPG output trigger

- A/D converter conversion start trigger can be generated
  - Channel 0 to 5 compare match A/input capture A signals can be used as A/D converter conversion start trigger

- Module stop mode can be set
  - The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode.

Note: * The H8S/2355 Series and H8S/2345 Series do not have a DMAC or PPG.

Block Diagram



3.5 Programmable Pulse Generator (PPG)

A built-in programmable pulse generator (PPG) provides pulse outputs by using the 16-bit timer-pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (group 3 to group 0) that can operate both simultaneously and independently.

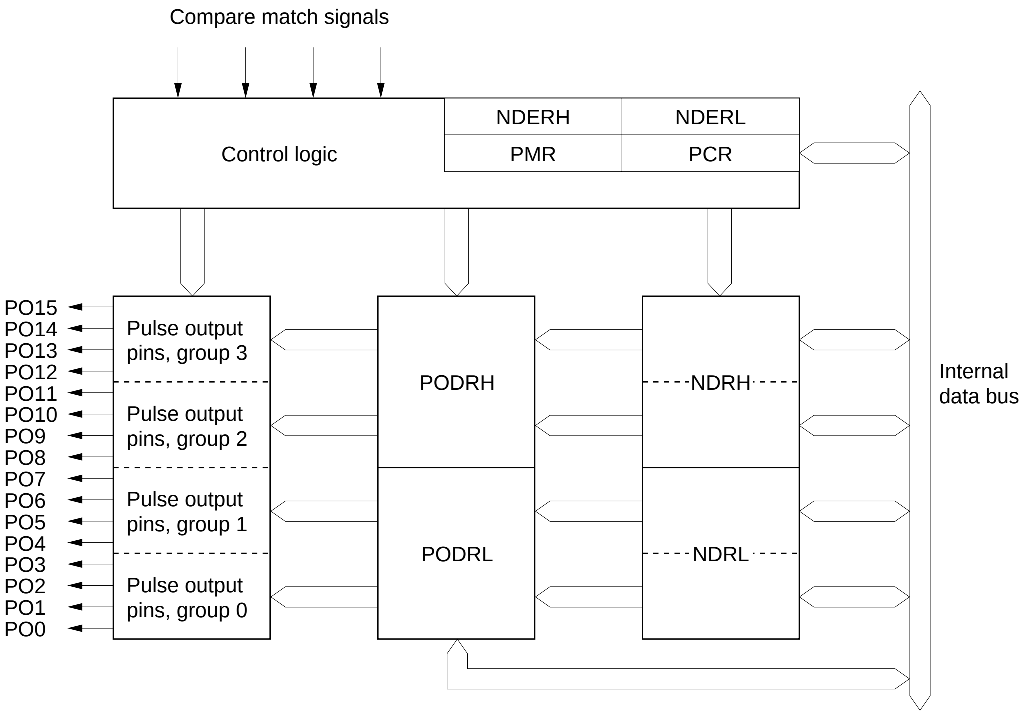
Product Series with On-Chip PPG

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	Available	—	Available	—

Features

- 16-bit output data
  - — Maximum 16-bit data can be output, and output can be enabled on a bit-by-bit basis.
- Four output groups
  - — Output trigger signals can be selected in 4-bit groups to provide up to four different 4-bit outputs.
- Selectable output trigger signals
  - — Output trigger signals can be selected for each group from the compare match signals of four TPU channels.
- Non-overlap mode
  - — A non-overlap margin can be provided between pulse outputs.
- Can operate together with the data transfer controller (DTC) and DMA controller (DMAC).
  - — The compare match signals selected as output trigger signals can activate the DTC or DMAC for sequential output of data without CPU intervention.
- Settable inverted output
  - — Inverted data can be output for each group.
- Module stop mode can be set
  - — The initial setting is for PPG operation to be halted. Register access is enabled by clearing module stop mode.

Block Diagram



Legend:

PMR	: PPG output mode register
PCR	: PPG output control register
NDERH	: Next data enable register H
NDERL	: Next data enable register L
NDRH	: Next data register H
NDRL	: Next data register L
PODRH	: Output data register H
PODRL	: Output data register L

3.6 8-Bit Timers

The 8-bit timer module has two channels based on an 8-bit counter. Each channel has an 8-bit counter (TCNT) and two time constant registers (TCORA and TCORB) that are constantly compared with the TCNT value to detect compare match events. The 8-bit timer module can thus be used for a variety of functions, including pulse output with an arbitrary duty cycle.

Product Series with On-Chip

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	—	Available	Available	Available

Features

- Selection of four clock sources
  — The counters can be driven by one of three internal clock signals (ø/8, ø/64, or ø/8192) or an external clock input (enabling use as an external event counter).

- Selection of three ways to clear the counters
  — The counters can be cleared on compare match A or B, or by an external reset signal.

- Timer output control by a combination of two compare match signals
  — The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output.

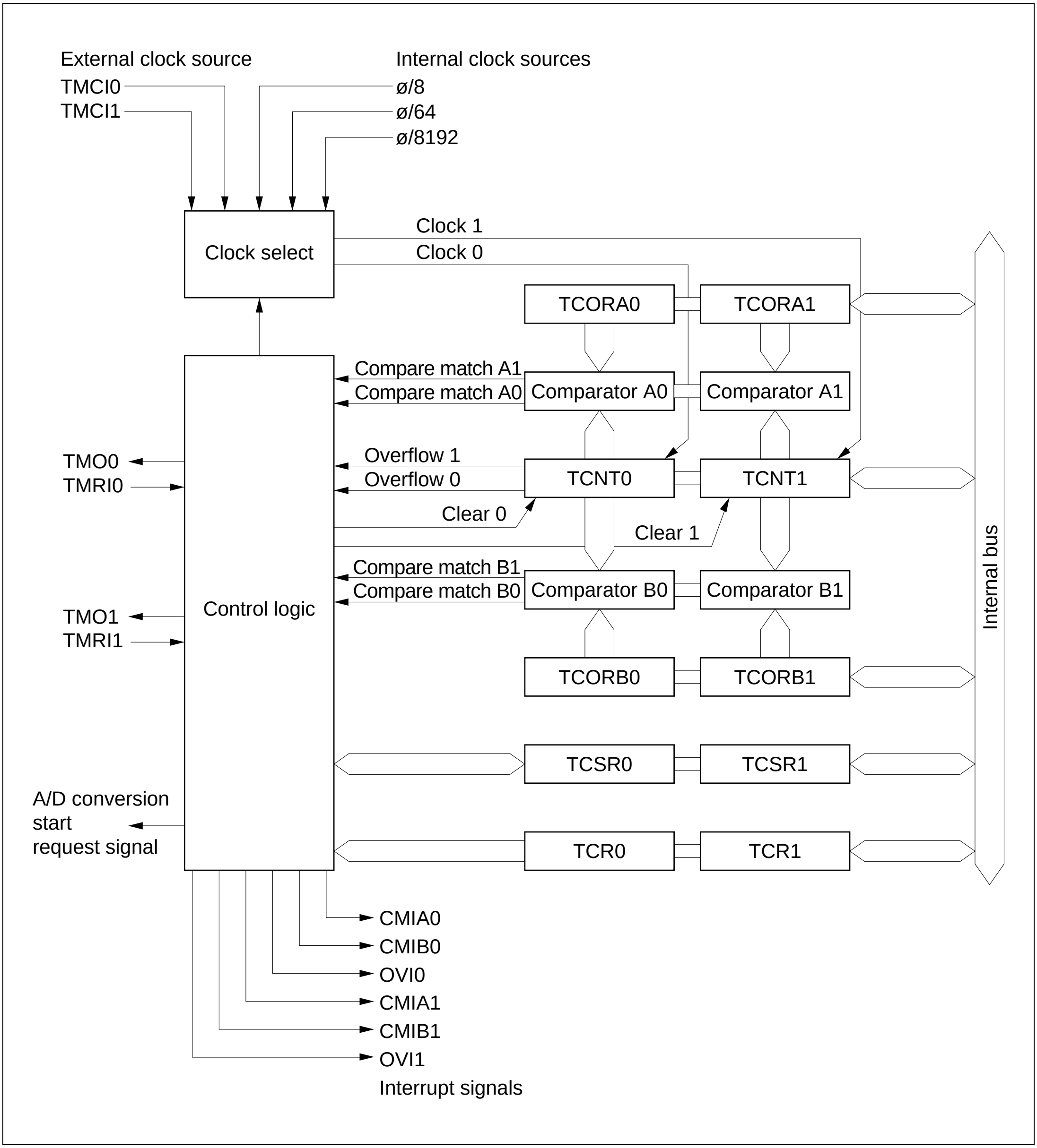
- Provision for cascading of two channels
  — Operation as a 16-bit timer is possible, using channel 0 for the upper 8 bits and channel 1 for the lower 8 bits (16-bit count mode).
  — Channel 1 can be used to count channel 0 compare matches (compare match count mode).

- Three independent interrupts
  — Compare match A and B and overflow interrupts can be requested independently.

- A/D converter conversion start trigger can be generated
  — The channel 0 compare match A signal can be used as an A/D converter start trigger

- Module stop mode can be set
  — The initial setting is for 8-bit timer operation to be halted. Register access is enabled by clearing module stop mode.

Block Diagram



3.7 Watchdog Timer (WDT)

These series have a single-channel on-chip watchdog timer (WDT) for monitoring system operation.
When this watchdog function is not needed, the WDT can be used as an interval timer.

Product Series with On-Chip WDT

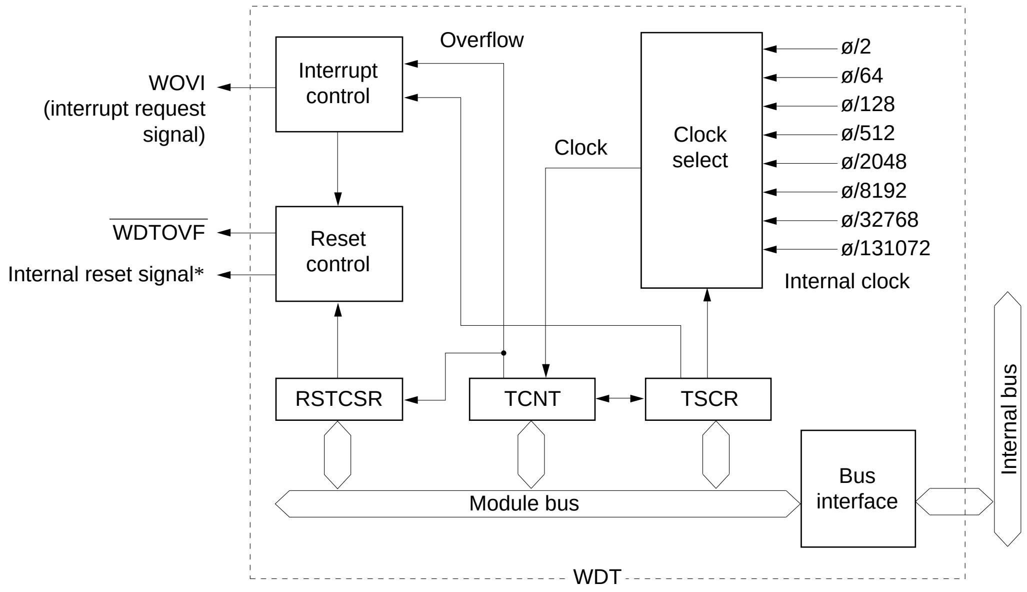
H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	Available	Available	Available	Available

Features

- Switchable between watchdog timer mode and interval timer mode
- $\overline{\text{WDTOVF}}$ output when in watchdog timer mode*
  — If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$. It is possible to select whether or not the entire H8S/2000 Series is reset at the same time. This internal reset can be a power-on reset or a manual reset.

  Note: * The $\overline{\text{WDTOVF}}$ pin function is not available in the H8S/2357 and H8S/2345 F-ZTAT versions.
- Interrupt generation when in interval timer mode
  — If the counter overflows, the WDT generates an interval timer interrupt.
- Choice of eight counter clock sources.

Block Diagram



Legend:

TCSR : Timer control/status register
TCNT : Timer counter
RSTCSR : Reset control/status register

Note: * The type of internal reset signal depends on a register setting. Either power-on reset or manual reset can be selected.

3.8 Serial Communication Interface (SCI)

The serial communication interface (SCI) can handle both asynchronous and clocked synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function). The number of on-chip channels depends on the product series.

Product Series with On-Chip SCI

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
3 channels	2 channels	3 channels	3 channels	2 channels

Features

- Choice of asynchronous or clocked synchronous serial communication mode

  Asynchronous mode

  — Serial data communication executed using asynchronous system in which synchronization is achieved character by character

  — Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)

  — A multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data communication formats

Data length:	7 or 8 bits
Stop bit length:	1 or 2 bits
Parity:	Even/odd/none
Multiprocessor bit:	1 or 0

  — Receive error detection: Parity, overrun, and framing errors

  — Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

  Synchronous mode

  — Serial data communication synchronized with a clock

  — Serial data communication can be carried out with other chips that have a synchronous communication function

  — One serial data communication format

  — Data length: 8 bits

  — Receive error detection: Overrun errors detected

- Full-duplex communication capability
  - The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously
  - Data register double-buffering enables continuous transmission/reception

- Selection of LSB-first or MSB-first transfer
  - This choice can be made regardless of the communication mode (with the exception of 7-bit data transfer in asynchronous mode)

- On-chip dedicated baud rate generator allows any bit rate to be selected

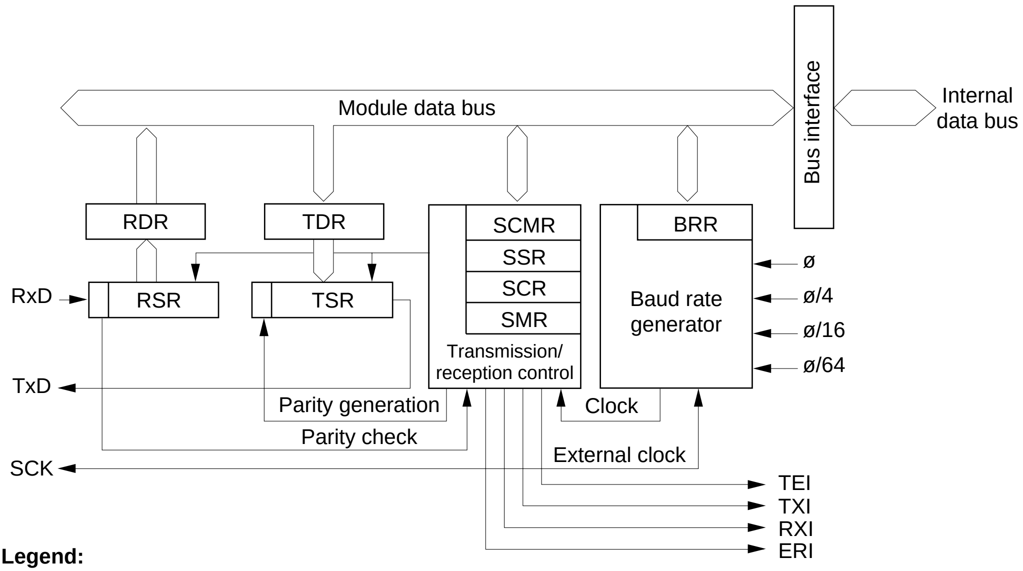
- Selection of transmit/receive clock source
  - An internal clock from the baud rate generator or an external clock from the SCK pin can be selected

- Four interrupt sources
  - Four interrupt sources—transmit data empty, transmission end, receive data full, and receive error—that can issue requests independently
  - The transmit data empty interrupt and receive data full interrupt can activate the DMA controller (DMAC)* or data transfer controller (DTC) to execute data transfer

    Note: * The H8S/2355 Series and H8S/2345 Series do not have a DMAC.

- Module stop mode setting
  - The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode.

Block Diagram



Legend:

SCMR:	Smart card mode register
RSR:	Receive shift register
RDR:	Receive data register
TSR:	Transmit shift register
TDR:	Transmit data register
SMR:	Serial mode register
SCR:	Serial control register
SSR:	Serial status register
BRR:	Bit rate register

3.9 Smart Card Interface

SCI supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function.

Product Series with On-Chip

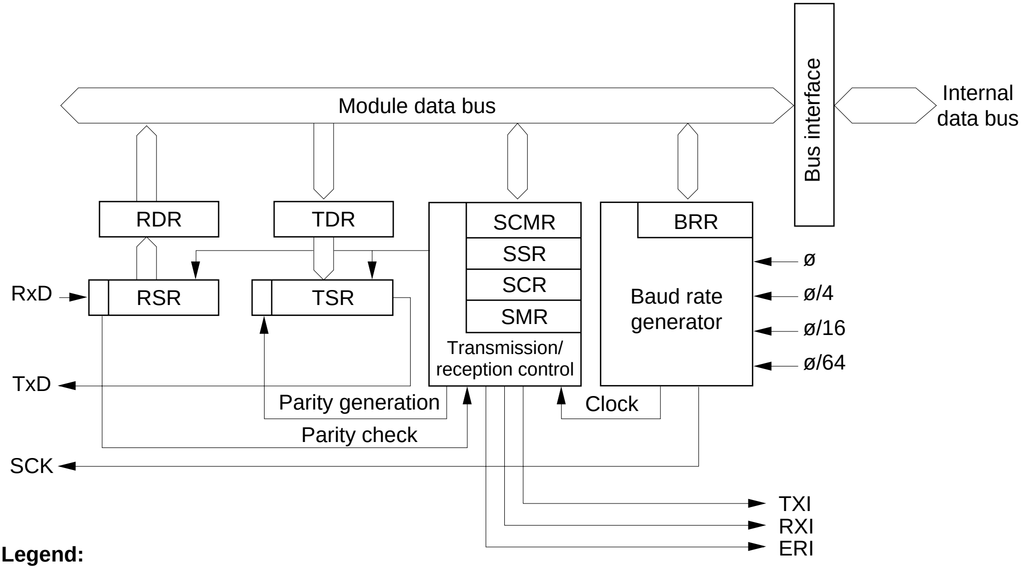
H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
3 channels	2 channels	3 channels	3 channels	2 channels

Features

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and checking
  - Transmission of error signal (parity error) in receive mode
  - Error signal detection and automatic data retransmission in transmit mode
  - Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit data empty, receive data full, and transmit/receive error) that can issue requests independently
  - The transmit data empty interrupt and receive data full interrupt can activate the DMA controller (DMAC)* or data transfer controller (DTC) to execute data transfer

Note: * The H8S/2355 Series and H8S/2345 Series do not have a DMAC.

Block Diagram



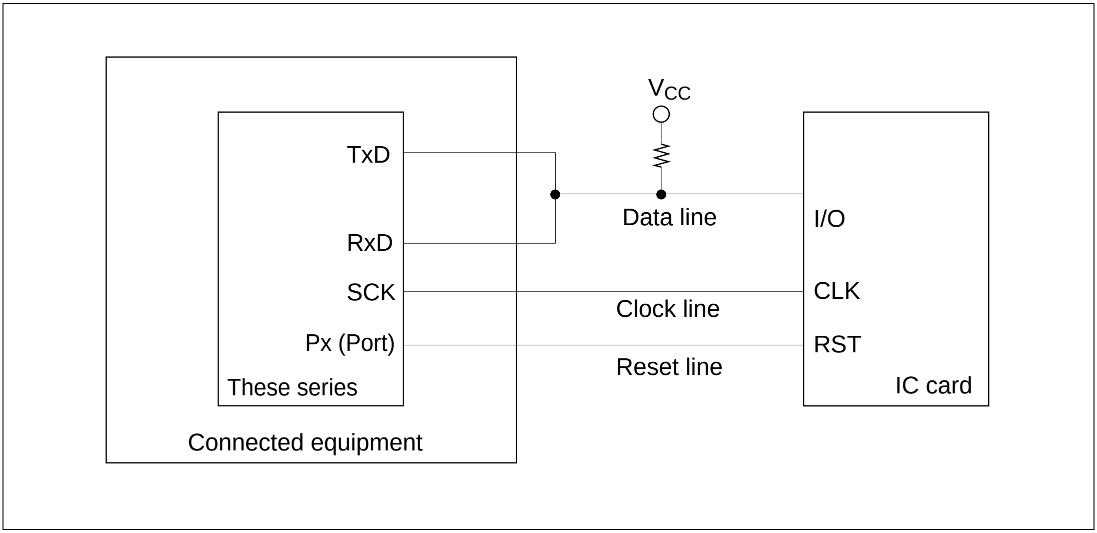
Legend:

SCMR	: Smart card mode register
RSR	: Receive shift register
RDR	: Receive data register
TSR	: Transmit shift register
TDR	: Transmit data register
SMR	: Serial mode register
SCR	: Serial control register
SSR	: Serial status register
BRR	: Bit rate register

Outline of Operation

The main functions of the smart card interface are as follows.

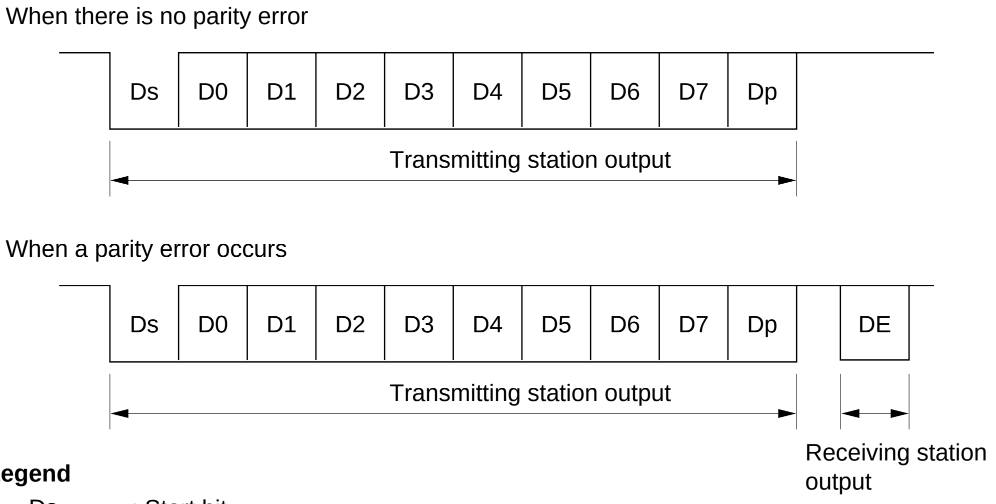
- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer.
- Only asynchronous communication is supported; there is no clocked synchronous communication function.



Schematic Diagram of Smart Card Interface Pin Connections

Data Format

When receiving, a parity check is carried out on each frame, and if an error is detected an error signal is returned to the transmitting side and data retransmission is requested. If an error signal is sampled during transmission, the same data is retransmitted.



When there is no parity error
Ds
D0
D1
D2
D3
D4
D5
D6
D7
Dp
Transmitting station output
When a parity error occurs
Ds
D0
D1
D2
D3
D4
D5
D6
D7
Dp
DE
Transmitting station output
Receiving station output

Legend

Ds	: Start bit
D0 to D7	: Data bits
Dp	: Parity bit
DE	: Error signal

Smart Card Interface Data Format

3.10 A/D Converter

The H8S/2000 Series has an on-chip A/D converter with 10-bit precision. A/D converters are of two kinds, type 1 and type 2. Analog signals can be input on up to eight channels by the program.

3.10.1 Type 1

Product Series with On-Chip

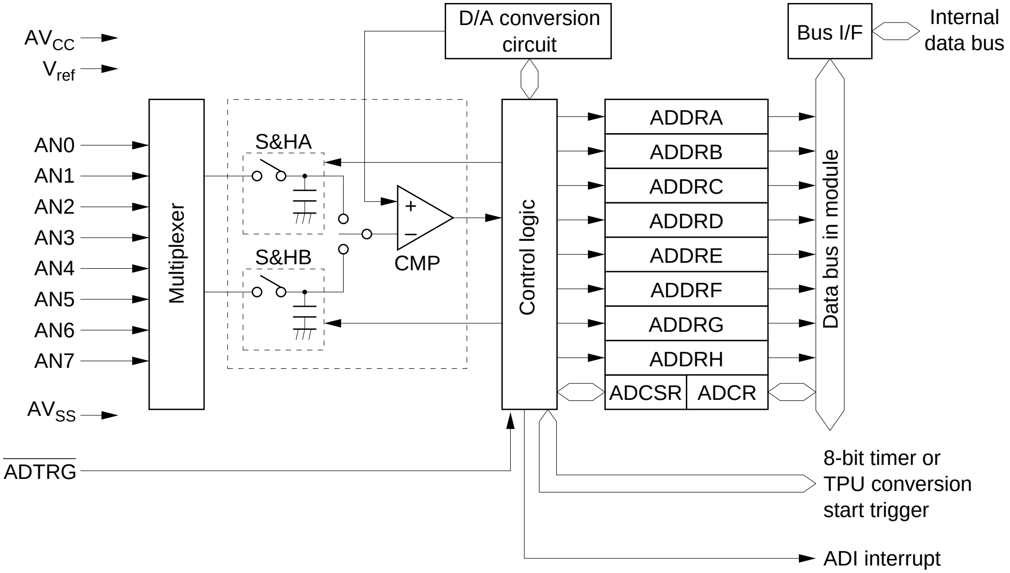
H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	—	—	—	—

Features

- 10-bit resolution
- Input channels: 8 channels
- Settable analog conversion voltage range
  - Conversion of analog voltages from 0 V to V_{ref}, with the reference voltage pin (V_{ref}) as the analog reference voltage
- High-speed conversion
  - Minimum conversion time: 2.2 µs per channel (at 20 MHz operation)
    1.0 µs per channel in continuous conversion
- Variety of conversion modes
  - Choice of select mode or group mode
  - Choice of single mode or scan mode
  - Buffer operation possible
  - Simultaneous 2-channel sampling possible
- Three kinds of conversion start
  - Choice of software or timer conversion start trigger (TPU or 8-bit timer), or $\overline{\text{ADTRG}}$ pin
- Eight data registers
  - Conversion results held in a data register for each channel
- Sample and hold function
- A/D conversion end interrupt generation
  - A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion

- Module stop mode can be set
  — The initial setting is for A/D converter operation to be halted. Register access is enabled by clearing module stop mode.

Block Diagram



Legend:

ADCR	: A/D control register	CMP	: Comparator array
ADCSR	: A/D control status register	S&H	: Sample and hold circuit
ADDRA	: A/D data register A	ADDRE	: A/D data register E
ADDRB	: A/D data register B	ADDRF	: A/D data register F
ADDRC	: A/D data register C	ADDRG	: A/D data register G
ADDRD	: A/D data register D	ADDRH	: A/D data register H

Block Diagram of A/D Converter

3.10.2 Type 2

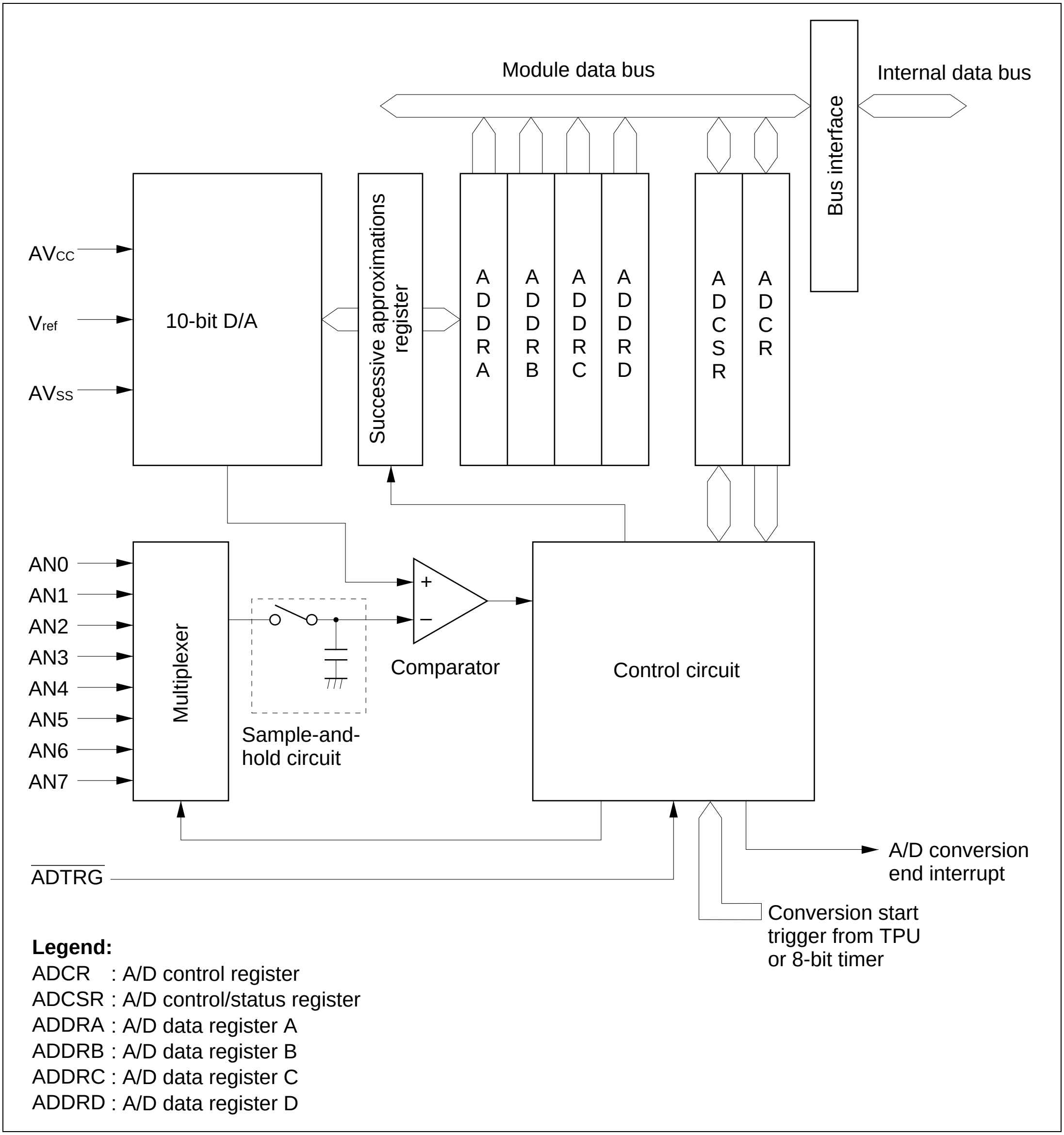
Product Series with On-Chip WDT

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
—	Available	Available	Available	Available

Features

- 10-bit resolution
- Input channels: 8 channels
- Settable analog conversion voltage range
  — The analog conversion voltage range is set with the reference voltage pin (V_{ref}) as the analog reference voltage.
- Conversion time
  — Minimum conversion time: 6.7 µs per channel (at 20 MHz operation)
- Selection of single mode or scan mode
  — Single mode: A/D conversion on one channel
  — Scan mode: Consecutive A/D conversion on one to four channels
- Four data registers
  — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  — Selection of software or timer conversion start trigger (TPU or 8-bit timer*), or $\overline{\text{ADTRG}}$ pin
  Note: * The H8S/2350 Series has no 8-bit timers.
- A/D conversion end interrupt
  — An A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion
- Module stop mode can be set
  — The initial setting is for A/D converter operation to be halted. Register access is enabled by clearing module stop mode.

Block Diagram


Module data bus
Internal data bus
Bus interface
AVCC
Vref
AVSS
10-bit D/A
Successive approximations register
ADDRA
ADDRB
ADDRC
ADDRD
ADCSR
ADCR
AN0
AN1
AN2
AN3
AN4
AN5
AN6
AN7
Multiplexer
Sample-and-hold circuit
+
−
Comparator
Control circuit
ADTRG
A/D conversion end interrupt
Conversion start trigger from TPU or 8-bit timer
Legend:
ADCR : A/D control register
ADCSR : A/D control/status register
ADDRA : A/D data register A
ADDRB : A/D data register B
ADDRC : A/D data register C
ADDRD : A/D data register D

Block Diagram of A/D Converter

3.11 D/A Converter

The D/A converter has 8-bit resolution, and can output analog signals on up to two channels under program control.

Product Series with On-Chip D/A Converter

H8S/2655 Series	H8S/2350 Series	H8S/2355 Series	H8S/2357 Series	H8S/2345 Series
Available	Available	Available	Available	Available

Features

D/A converter features are listed below

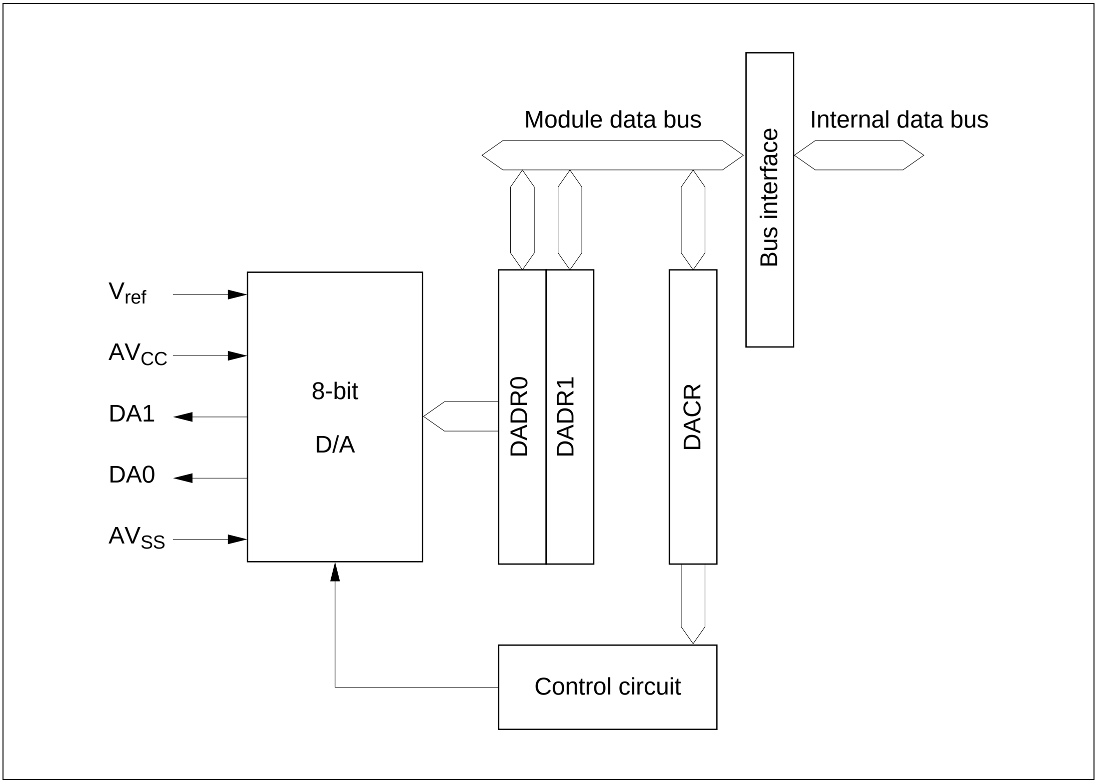
- 8-bit resolution
- Two output channels
- Maximum conversion time of 10 μs (with 20 pF load)
- Output voltage of 0 V to V_{ref}
- D/A output hold function in software standby mode
- Module stop mode can be set
  — The initial setting is for D/A converter operation to be halted. Register access is enabled by clearing module stop mode.

Operation

D/A converter operation is enabled by setting the D/A output enable bit to 1. While this bit is set to 1, DADR contents are constantly converted and output to the corresponding pin.

The output value is: $\frac{\text{DADR contents}}{256} \times V_{ref}$

Block Diagram



Block Diagram of D/A Converter

3.12 RAM

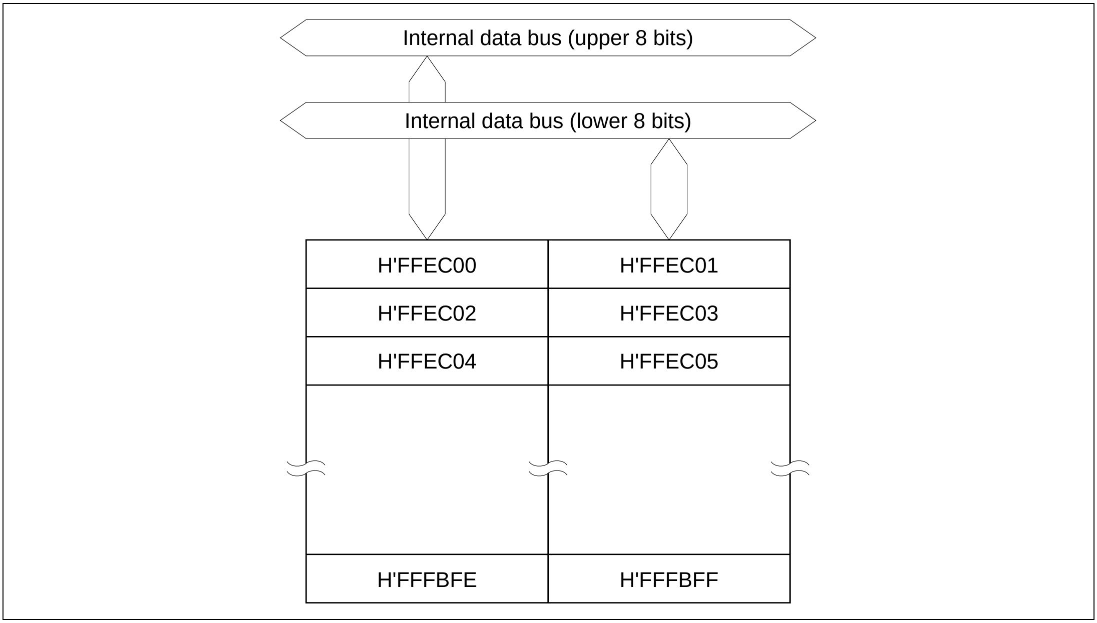
The on-chip RAM is connected to the CPU by a 16-bit data bus, enabling both byte data and word data to be accessed in one state. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAME) in the system control register (SYSCR). The RAM size varies from product to product. Refer to the hardware manual for the relevant product.

Product Series with On-Chip RAM

Series	**H8S/2655**		**H8S/2350**		**H8S/2355**		**H8S/2357**		**H8S/2345**	
Type Name	H8S/2655	H8S/2653	H8S/2350	H8S/2351	H8S/2355	H8S/2353	H8S/2357	H8S/2352*	H8S/2345	H8S/2343
RAM (Bytes)	4k	4k	2k	2k	4k	2k	8k	8k	4k	2k

Note: * Under development


Internal data bus (upper 8 bits)
Internal data bus (lower 8 bits)
H'FFEC00
H'FFEC01
H'FFEC02
H'FFEC03
H'FFEC04
H'FFEC05
H'FFFBFE
H'FFFBFF

Block Diagram of RAM (Example of 4-Kbyte RAM)

3.13 ROM

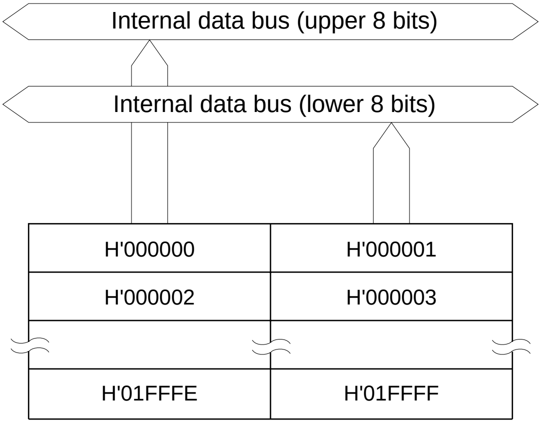
The ROM is connected to the CPU by a 16-bit data bus, enabling both byte data and word data to be accessed in one state. This makes possible rapid instruction fetches and high-speed processing.

With the flash memory versions, on-board erasing and programming is possible in addition to erasing and programming using a dedicated PROM programmer.

Product Series with On-Chip ROM

Series	H8S/2655		H8S/2350		H8S/2355		H8S/2357		H8S/2345	
Type Name	H8S/2655	H8S/2653	H8S/2350	H8S/2351	H8S/2355	H8S/2353	H8S/2357	H8S/2352	H8S/2345	H8S/2343
ROM (Bytes)	128k	64k	—	64k	128k	64k	128k	—	128k	64k
ROM Type	ZTAT, mask ROM		Mask ROM		ZTAT, mask ROM		ZTAT, mask ROM, F-ZTAT		ZTAT, mask ROM, F-ZTAT*	

Note: * Under development


Internal data bus (upper 8 bits)
Internal data bus (lower 8 bits)
H'000000
H'000001
H'000002
H'000003
H'01FFFE
H'01FFFF

ROM Block Diagram (Example of 128-Kbyte ROM)

PROM Programming

PROM versions suspend their microcomputer functions when placed in PROM mode, enabling the on-chip PROM to be programmed. PROM programming can be done using the same specifications as for the HN27C101 EPROM (V_{PP} = 12.5 V). Use of a 120/128/100-pin-to-32-pin socket adapter enables programming with a commercial PROM programmer. The address range is H'00000 to H'1FFFF. Page programming is not supported.

Features of Flash Memory

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode

- Programming/erase methods
  - 32 bytes programmed simultaneously
  - Block erase
  - Block erasing can be performed as required on 1-kbyte, 28-kbyte, 16-kbyte, 8-kbyte, and 32-kbyte blocks

- Programming/erase times
  - Programming time: 10 ms (typ.) per 32-byte programming operation
    300 µs (typ.) per byte
  - Erase time: 100 ms (typ.) per block

- Reprogramming up to 100 times

- On-board programming modes
  - Boot mode
  - User program mode

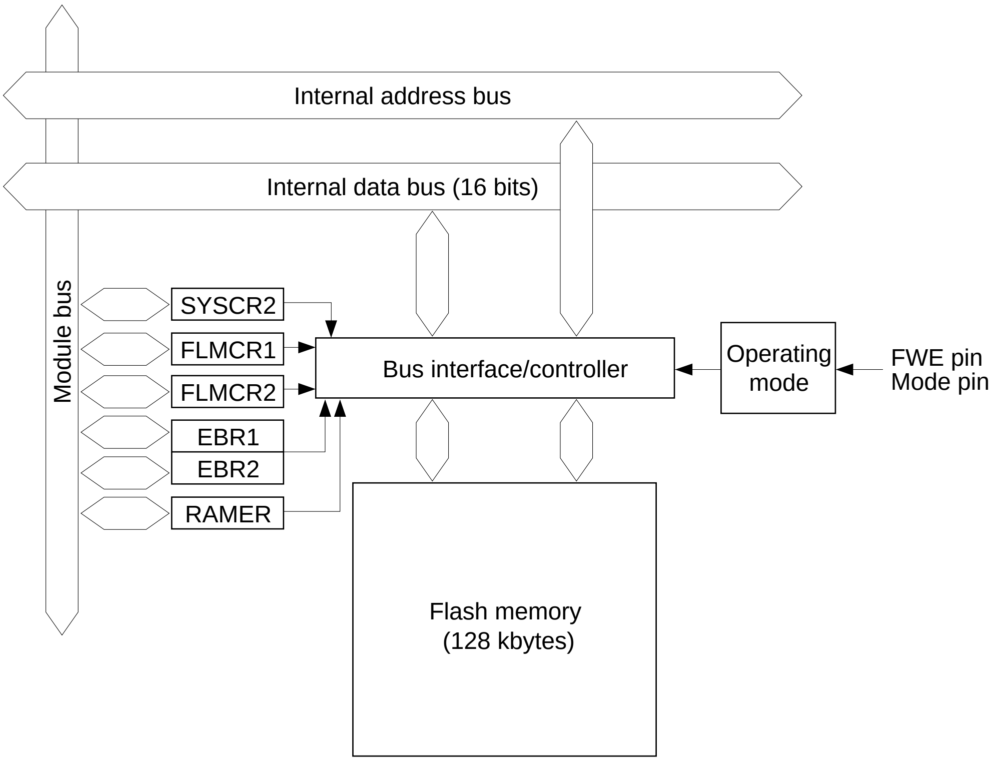
- Automatic bit rate adjustment
  - With data transfer in boot mode, the bit rate of the chip can be automatically adjusted to match the transfer bit rate of the host.

- Flash memory emulation by RAM
  - Real-time programming of flash memory can be emulated by overlapping part of RAM onto flash memory.

- Protect modes
  - There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase/verify operations

- Writer mode
  - Programming, erasing, and verifying can be carried out using a general-purpose PROM programmer

Block Diagram



Legend:

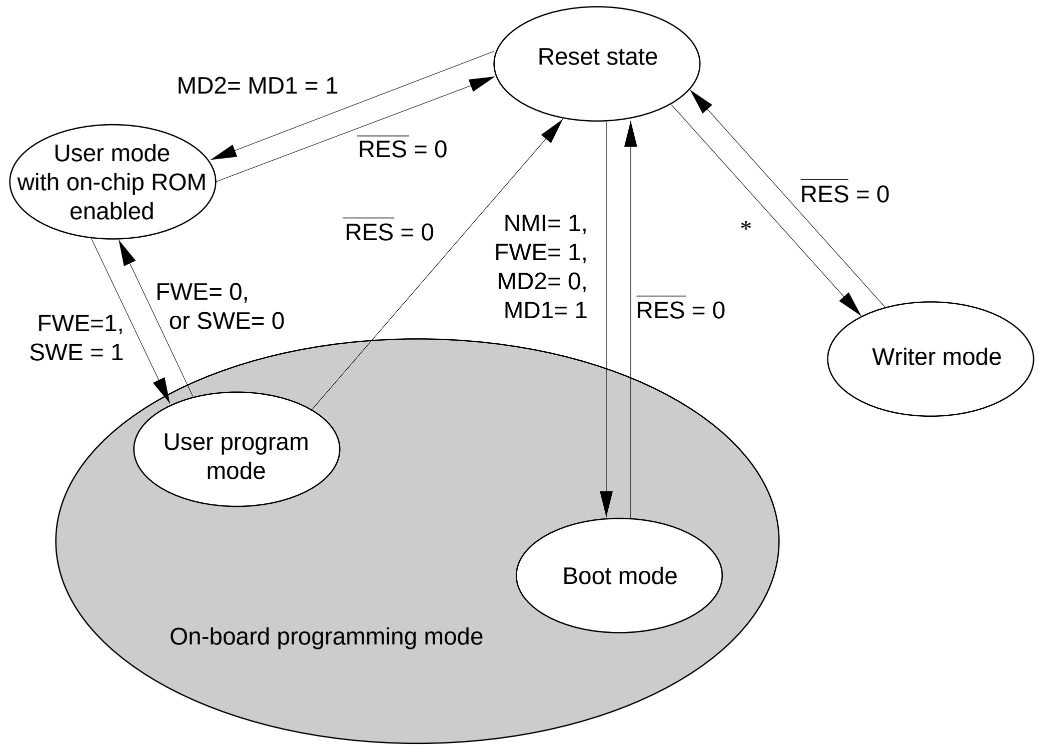
SYSCR2:	System control register 2
FLMCR1:	Flash memory control register 1
FLMCR2:	Flash memory control register 2
EBR1:	Erase block register 1
EBR2:	Erase block register 2
RAMER:	RAM emulation register

Block Diagram of Flash Memory (Example of H8S/2357 F-ZTAT™)

Flash Memory Operating Modes

Mode Transitions: When the mode pins are set in the reset state and a reset-start is executed, the MCU enter one of the operating modes shown below. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and writer mode.


Reset state
MD2= MD1 = 1
$\overline{\text{RES}}$ = 0
User mode with on-chip ROM enabled
$\overline{\text{RES}}$ = 0
NMI= 1, FWE= 1, MD2= 0, MD1= 1
$\overline{\text{RES}}$ = 0
*
$\overline{\text{RES}}$ = 0
FWE= 0, or SWE= 0
FWE=1, SWE = 1
Writer mode
User program mode
Boot mode
On-board programming mode

Note: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

* MD2= MD1= MD0= 0, P66= 1, P65= P64= 0

Mode Transitions

Section 4 Power-Down Modes

In addition to the normal program execution state, the H8S/2000 series have power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

Overview

The H8S/2000 series operating modes are as follows:

1. High-speed mode
2. Medium-speed mode
3. Sleep mode
4. Module stop mode
5. Software standby mode
6. Hardware standby mode

Of these, 2 to 6 are power-down modes. Sleep mode is a CPU mode, medium-speed mode is a CPU and bus master mode, and module stop mode is an on-chip supporting module mode (including bus masters other than the CPU). A combination of certain of these modes can be set.

- Medium-Speed Mode

  When bits SCK2 to SCK0 in the system clock control register (SCKCR) are set to 1 in medium-speed mode is entered as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock (*ϕ*/2, *ϕ*/4, *ϕ*/8, *ϕ*/16, or *ϕ*/32) specified by bits SCK2 to SCK0. The bus masters other than the CPU (the DMAC and DTC) also operate in medium-speed mode. However, on-chip supporting modules other than the bus masters operate on the high-speed clock (*ϕ*).

  In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. (For example, if ø/4 is selected as the operating clock, on-chip memory is accessed in four states, and internal I/O registers in eight states.)

  Medium-speed mode is cleared by clearing bits SCK2 to SCK0 to 0. High-speed mode is restored at the end of the current bus cycle.

- Sleep Mode

  If a SLEEP instruction is executed when the SSBY bit in the standby control register (SBYCR) is cleared to 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other supporting modules do not stop.

  Sleep mode is cleared by a reset or any interrupt, and the CPU returns to the normal program execution state via the exception handling state.

- Module Stop Mode

  Module stop mode allows individual on-chip supporting modules to be stopped.

  When the MSTP bit corresponding to a particular supporting module in the module stop control register (MSTPCR) is set to 1, operation of the specified module stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

  When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI and A/D converter (type 2 only) are retained.

  After reset release, all modules except the DMAC and DTC are in module stop mode. Registers of modules set to module stop mode cannot be read or written.

- Software Standby Mode

  If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, in software standby mode is entered. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, on-chip RAM data, the states of on-chip supporting modules other than the SCI, and A/D converter (type 2 only), and the states of I/O ports, are retained.

  Software standby mode is cleared by a reset or a external interrupt. After the elapse of the oscillation stabilization time, the program execution state is returned to via the exception handling state.

  As the oscillator is stopped in this mode, power consumption is extremely low.

- Hardware Standby Mode

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any state.

  In hardware standby mode, all functions enter the reset state and stop operation, resulting in extremely low power consumption. As long as the specified voltage is supplied, on-chip RAM data is retained. I/O ports go to the high-impedance state.

  Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is entered and clock oscillation is started. The $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin is subsequently driven high, the program execution state is returned to via the reset exception handling state.

  In this mode, as in software standby mode, power consumption is extremely low since the oscillator is stopped.

Operation Modes

Operating Mode	Transition Condition	Clearing Condition	Oscillator	CPU		Modules		I/O Ports
					Registers		Registers	
High speed mode	Control register		Functions	High speed	Functions	High speed	Functions	High speed
Medium-speed mode	Control register		Functions	Medium speed	Functions	High/ medium speed *1	Functions	High speed
Sleep mode	Instruction	Interrupt	Functions	Halted	Retained	High speed	Functions	High speed
Module stop mode	Control register		Functions	High/ medium speed	Functions	Halted	Retained/ reset *2	Retained
Software standby mode	Instruction	External interrupt	Halted	Halted	Retained	Halted	Retained/ reset *2	Retained
Hardware standby mode	Pin		Halted	Halted	Undefined	Halted	Reset	High impedance

Notes: 1. The bus master operates on the medium-speed clock, and other on-chip supporting modules on the high-speed clock.

2. The SCI, and A/D converter (type 2 only) are reset, and other on-chip supporting modules retain their state.

Section 5 Guide to Products

5.1 H8S/2655 Series Features

The H8S/2655 Series uses an H8S/2600 CPU core and offers high-speed arithmetic/logic and multiply-and-accumulate instruction execution plus powerful on-chip supporting functions, including a high-speed A/D converter. These features provide easy implementation of sophisticated, high-performance systems. The H8S/2655 Series also includes a high-functionality bus controller and multifunction, multichannel high-speed data transfer functions that make it possible to control large-scale systems with large external memory.

Product Lineup

Product Name	ROM (Bytes)	RAM (Bytes)	ROM Type*
H8S/2655	128k	4k	MZ
H8S/2653	64k	4k	M

Note: * M: Mask ROM version
Z: ZTAT version

High-Performance H8S/2600 CPU

- General register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)

- High-speed operation suitable for real-time control
  - All frequently used instructions execute in 1 to 2 states
  - Maximum operating frequency: 20 MHz
  - 8/16/32-bit register-register add/subtract: 50 ns
  - 8 × 8-bit register-register multiply: 150 ns (at 20 MHz operation)
  - 16 ÷ 8-bit register-register divide: 600 ns
  - 16 × 16-bit multiply-and-accumulate 200 ns
  - 32 ÷ 16-bit register-register divide: 1000 ns

- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction

- Two CPU operating modes
  — Normal mode: H8S/300 Series compatible, maximum 64-kbyte address space
  — Advanced mode: Maximum 16-Mbyte address space

Bus Controller

- Maximum 8-Mbyte DRAM or pseudo-SRAM directly connectable (or use of interval timer possible)

DMA Controller (DMAC)

Data Transfer Controller (DTC)

Six 16-Bit Timer-Pulse Unit (TPU) Channels

Programmable Pulse Generator (PPG)

Two 8-Bit Timer Channels

One Watchdog Timer (WDT) Channel

Three Serial Communication Interface (SCI) Channels

A/D Converter

- Resolution: 10 bits
- Input: 8 channels
- High-speed conversion: 2.2 µs minimum conversion time (at 20 MHz operation)
- Selection of select mode or group mode and single or scan mode

D/A Converter

- Resolution: 8 bits
- Output: 2 channels

Thirteen I/O ports

- 87 I/O pins, 8 input-only pins

Interrupt Controller

- Nine external interrupt pins (NMI, $\overline{IRQ}_0$ to $\overline{IRQ}_7$)
- 52 internal interrupt sources
- Selection of four interrupt control modes

Power-Down state

- Medium-speed mode
- Sleep mode
- Module stop mode
- Software standby mode
- Hardware standby mode

Seven MCU Operating Modes

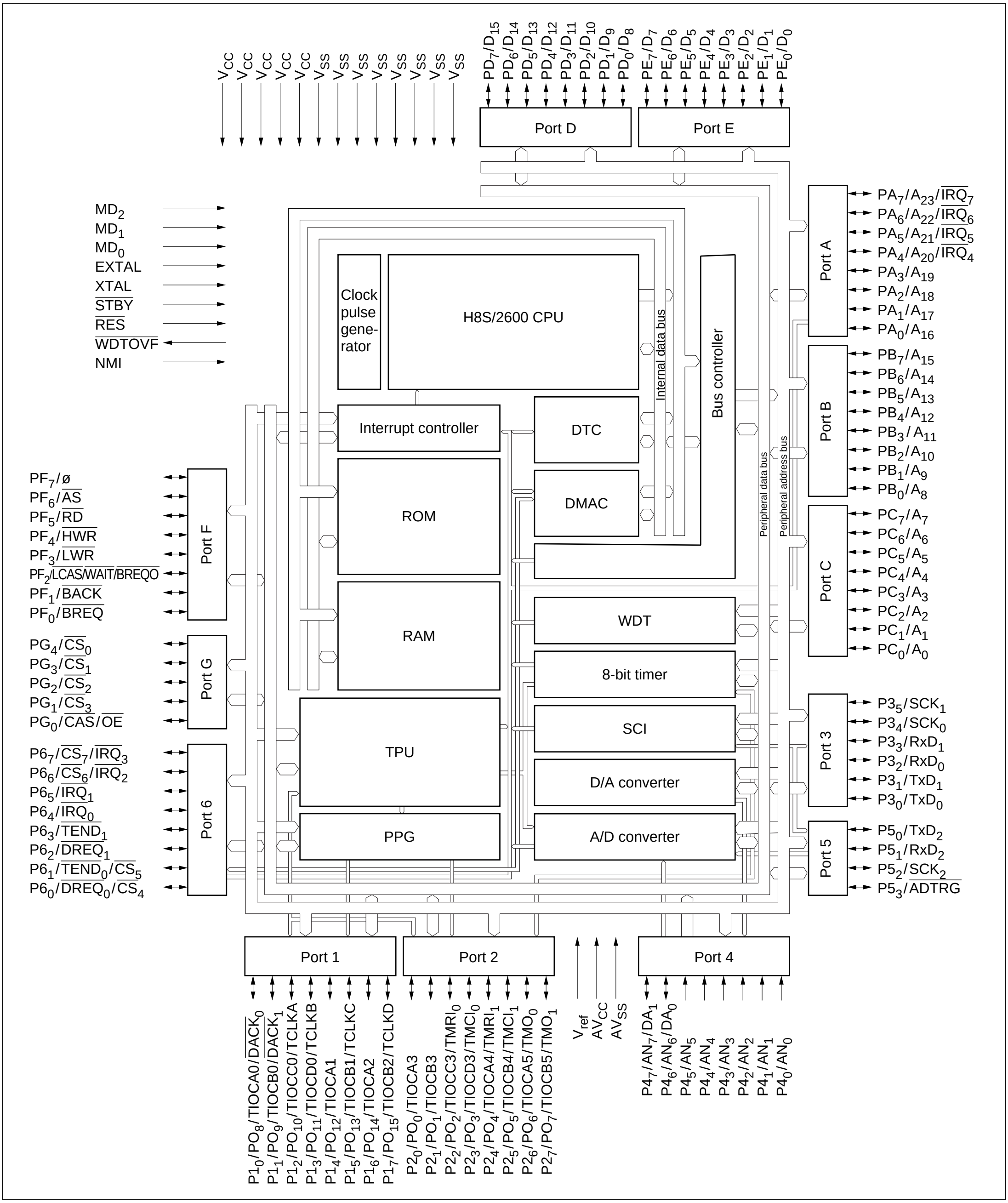
				External Data Bus	
Mode	CPU Operating Mode	Description	On-Chip ROM	Initial Value	Maximum Value
1	Normal	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
2		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
3		Single-chip mode	Enabled	—	—
4	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5		Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
6		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7		Single-chip mode	Enabled	—	—

On-Chip Clock Pulse Generator (1:1 oscillation)

Packages

- 120-pin plastic TQFP (TFP-120)
- 128-pin plastic QFP (FP-128)

Internal Block Diagram of H8S/2655 Series



5.2 H8S/2350 Series Features

The H8S/2350 Series uses an H8S/2000 CPU core and includes a high-functionality bus controller and multifunction, multichannel high-speed data transfer functions that make it possible to control large-scale systems with large external memory. ROMless and on-chip ROM versions are available. The H8S/2350 Series is pin-compatible with the H8S/2655 Series.

Product Lineup

Product Name	ROM (Bytes)	RAM (Bytes)	ROM Type*
H8S/2351	64k	2k	M
H8S/2350	—	2k	L

Note: * M: Mask ROM version
L: ROMless version

High-Performance H8S/2000 CPU

- General register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)

- High-speed operation suitable for real-time control

— All frequently used instructions execute in 1 to 2 states	
— Maximum operating frequency:	20 MHz
— 8/16/32-bit register-register add/subtract:	50 ns
— 8 × 8-bit register-register multiply:	600 ns
— 16 ÷ 8-bit register-register divide:	600 ns
— 16 × 16-bit register-register multiply:	1000 ns
— 32 ÷ 16-bit register-register divide:	1000 ns

- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions

- Two CPU operating modes
  - Normal mode: H8/300 Series compatible, maximum 64-kbyte address space
  - Advanced mode: Maximum 16-Mbyte address space

Bus Controller

- Maximum 8-Mbyte DRAM directly connectable (or use of interval timer possible)

DMA Controller (DMAC)

Data Transfer Controller (DTC)

Six 16-Bit Timer-Pulse Unit (TPU) Channels

Programmable Pulse Generator (PPG)

One Watchdog Timer (WDT) Channel

Two Serial Communication Interface (SCI) Channels

A/D Converter

- Resolution: 10 bits
- Input: 8 channels
- Conversion time: 6.7 µs (at 20 MHz operation)
- Selection of single or scan mode

D/A Converter

- Resolution: 8 bits
- Output: 2 channels

Thirteen I/O Ports*

- 87 I/O pins*, 8 input-only pins

  Note: * Ports B, C, and D, and pins PF_6 to PF_3, cannot be used as I/O ports in the H8S/2350.

Interrupt Controller

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- 42 internal interrupt sources
- Selection of two interrupt control modes

Power-Down State

- Medium-speed mode
- Sleep mode
- Module stop mode
- Software standby mode
- Hardware standby mode

Seven MCU operating modes

				External Data Bus	
Mode	**CPU Operating Mode**	**Description**	**On-Chip ROM**	**Initial Value**	**Maximum Value**
1	Normal	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
2*		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
3*		Single-chip mode	Enabled	—	—
4	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5		Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
6*		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7*		Single-chip mode	Enabled	—	—

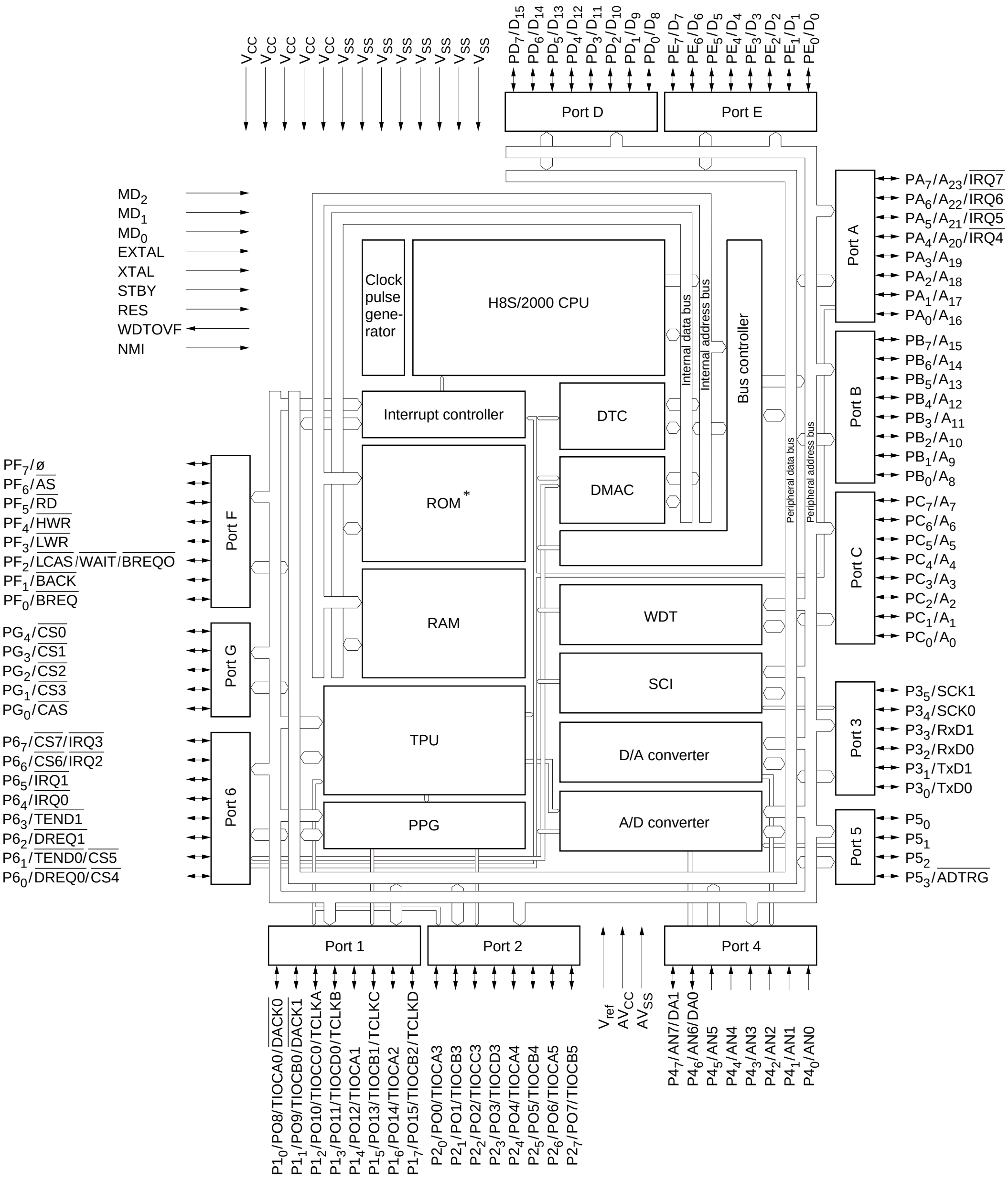
Note: * Applies to the H8S/2351 only.

On-Chip Clock Pulse Generator (1:1 oscillation)

Packages

- 120-pin plastic TQFP (TFP-120)
- 128-pin plastic QFP (FP-128)

Internal Block Diagram of H8S/2350 Series



Note: * Applies to the H8S/2351 only.

5.3 H8S/2355 Series Features

The H8S/2355 Series uses an H8S/2000 CPU core and includes large-capacity ROM and RAM and powerful on-chip supporting functions that enable high-speed, high-precision control of both small- and large-scale systems. The H8S/2355 Series is pin-compatible with the H8S/2655 Series.

Product Lineup

Product Name	ROM (Bytes)	RAM (Bytes)	ROM Type*
H8S/2355	128k	4k	MZ
H8S/2353	64k	2k	M

Note: * M: Mask ROM version
Z: ZTAT version

High-Performance H8S/2000 CPU

- General register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)

- High-speed operation suitable for real-time control
  - All frequently used instructions execute in 1 to 2 states
  - Maximum operating frequency: 20 MHz
  - 8/16/32-bit register-register add/subtract: 50 ns
  - 8 × 8-bit register-register multiply: 600 ns
  - 16 ÷ 8-bit register-register divide: 600 ns
  - 16 × 16-bit register-register multiply: 1000 ns
  - 32 ÷ 16-bit register-register divide: 1000 ns

- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions

- Two CPU operating modes
  - Normal mode: H8/300 Series compatible, maximum 64-kbyte address space
  - Advanced mode: Maximum 16-Mbyte address space

Bus Controller

Data Transfer Controller (DTC)

Six 16-Bit Timer-Pulse Unit (TPU) Channels

Two 8-Bit Timer Channels

One Watchdog Timer (WDT) Channel

Three Serial Communication Interface (SCI) Channels

A/D Converter

- Resolution: 10 bits
- Input: 8 channels
- High-speed conversion: 6.7 µs (at 20 MHz operation)
- Selection of single or scan mode

D/A Converter

- Resolution: 8 bits
- Output: 2 channels

Thirteen I/O Ports

- 87 I/O pins, 8 input-only pins

Interrupt Controller

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- 47 internal interrupt sources
- Selection of two interrupt control modes

Power-Down State

- Medium-speed mode
- Sleep mode
- Module stop mode
- Software standby mode
- Hardware standby mode

Seven MCU Operating Modes

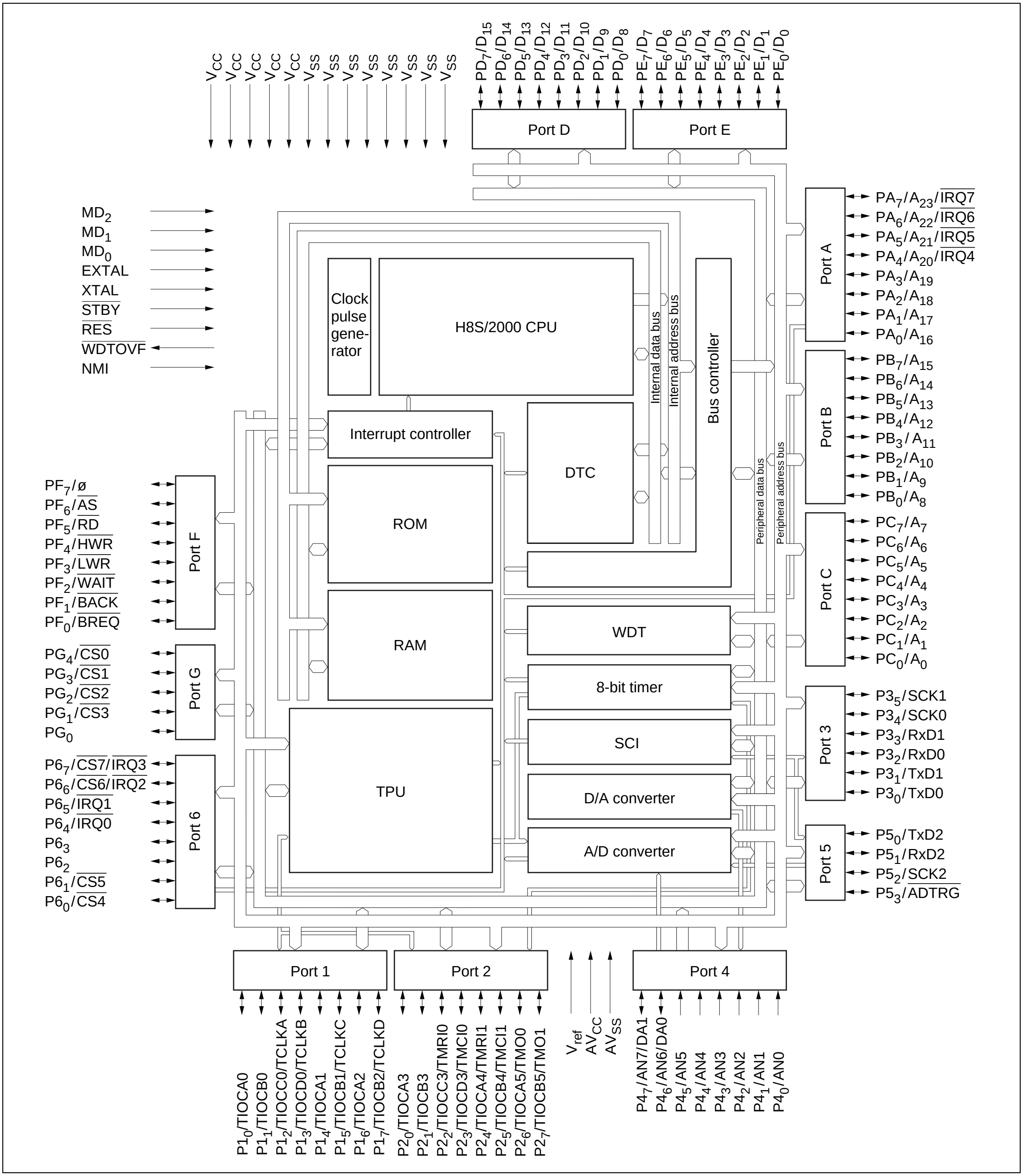
				External Data Bus	
Mode	CPU Operating Mode	Description	On-Chip ROM	Initial Value	Maximum Value
1	Normal	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
2		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
3		Single-chip mode	Enabled	—	—
4	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5		Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
6		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7		Single-chip mode	Enabled	—	—

On-Chip Clock Pulse Generator (1:1 oscillation)

Packages

- 120-pin plastic TQFP (TFP-120)
- 128-pin plastic QFP (FP-128)

Internal Block Diagram of H8S/2355 Series



5.4 H8S/2357 Series Features

The H8S/2357 Series uses an H8S/2000 CPU core, is pin-compatible with the H8S/2655 Series, and includes a variety of on-chip supporting functions. A single-power-supply flash memory version, ZTAT version, and mask ROM version are available. These versions enable users to respond quickly and flexibly to changing application specifications, growing production volumes, and other conditions.

Product Lineup

Product Name	ROM (Bytes)	RAM (Bytes)	ROM Type*
H8S/2357	128k	8k	MZF
H8S/2352	—	8k	L

Note: * M: Mask ROM version
Z: ZTAT version
F: F-ZTAT version
L: ROMless version

High-Performance H8S/2000 CPU

- General register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- High-speed operation suitable for real-time control
  - All frequently used instructions execute in 1 to 2 states
  - Maximum operating frequency: 20 MHz
  - 8/16/32-bit register-register add/subtract: 50 ns
  - 8 × 8-bit register-register multiply: 600 ns
  - 16 ÷ 8-bit register-register divide: 600 ns
  - 16 × 16-bit register-register multiply: 1000 ns
  - 32 ÷ 16-bit register-register divide: 1000 ns
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- CPU operating mode
  - Advanced mode: Maximum 16-Mbyte address space

Bus Controller

- Maximum 8-Mbyte DRAM directly connectable (or use of interval timer possible)

DMA Controller (DMAC)

Data Transfer Controller (DTC)

Six 16-Bit Timer-Pulse Unit (TPU) Channels

Programmable Pulse Generator (PPG)

Two 8-Bit Timer Channels

One Watchdog Timer (WDT) Channel

Three Serial Communication Interface (SCI) Channels

A/D Converter

- Resolution: 10 bits
- Input: 8 channels
- High-speed conversion: 6.7 µs minimum conversion time (at 20 MHz operation)
- Selection of single or scan mode

D/A Converter

- Resolution: 8 bits
- Output: 2 channels

Thirteen I/O Ports

- 87 I/O pins, 8 input-only pins

Interrupt Controller

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- 52 internal interrupt sources
- Selection of two interrupt control modes

Power-Down State

- Medium-speed mode
- Sleep mode
- Module stop mode
- Software standby mode
- Hardware standby mode

Eight MCU Operating Modes (F-ZTAT™ Version)

MCU Operating Mode	FWE	MD$_2$	MD$_1$	MD$_0$	CPU Operating Mode	Description	On-Chip ROMe	External Data Bus	
								Initial Value	Maximum Value
0	0	0	0	0	—	—	—	—	—
1				1					
2			1	0					
3				1					
4		1	0	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5				1				8 bits	16 bits
6			1	0		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7				1		Single-chip mode		—	—
8	1	0	0	0	—	—	—	—	—
9				1					
10			1	0	Advanced	Boot mode	Enabled	8 bits	16 bits
11				1				—	—
12		1	0	0	—	—	—	—	—
13				1					
14			1	0	Advanced	User program mode	Enabled	8 bits	16 bits
15				1				—	—

Four MCU Operating Modes
(ZTAT™ Version, Mask ROM Version, and ROMless Version)

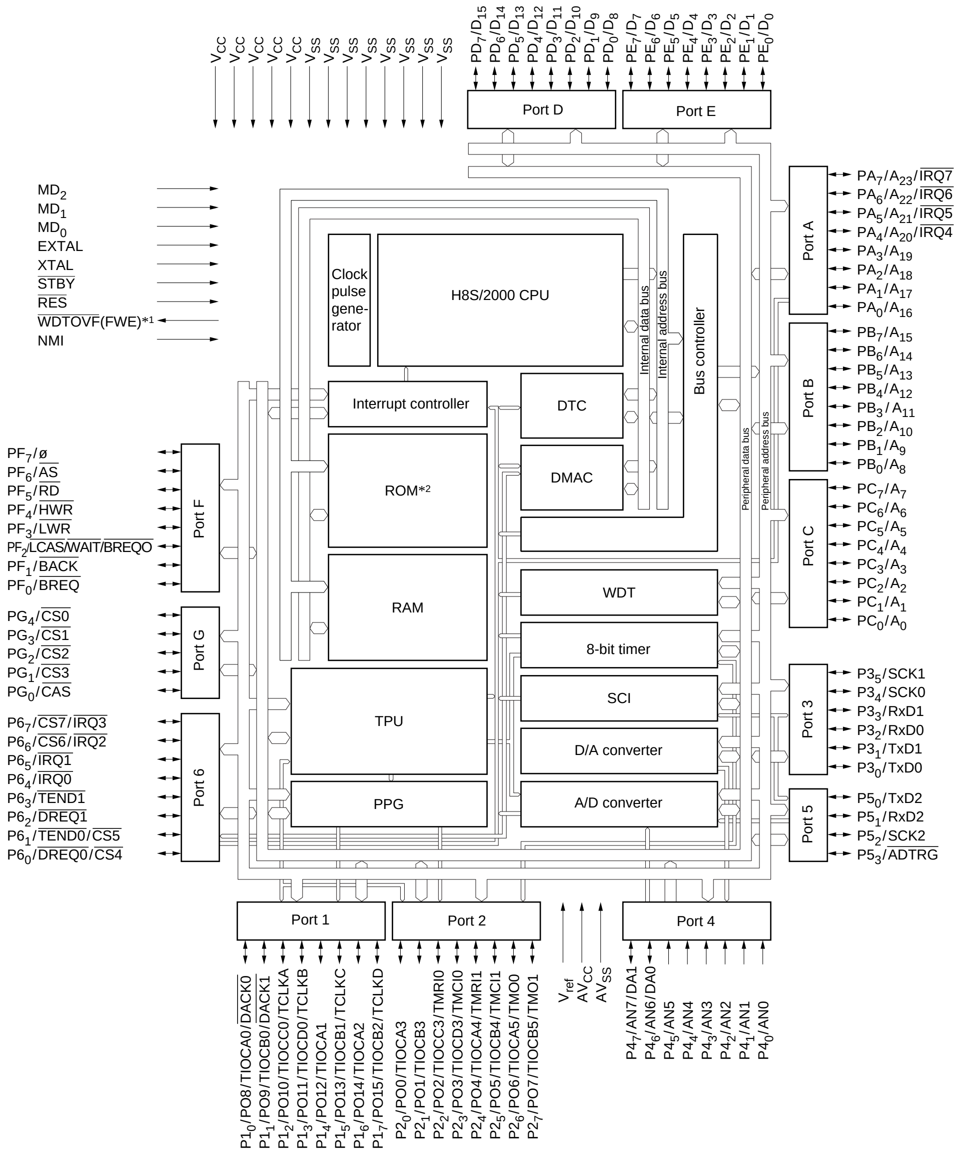
MCU Operating Mode	MD_2	MD_1	MD_0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus Initial Value	External Data Bus Maximum Value
0	0	0	0	—	—	—	—	—
1			1					
2		1	0					
3			1					
4	1	0	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5			1				8 bits	16 bits
6		1	0		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7			1		Single-chip mode		—	—

On-Chip Clock Pulse Generator (1:1 oscillation)

Packages

- 120-pin plastic TQFP (TFP-120)
- 128-pin plastic QFP (FP-128)

Internal Block Diagram of H8S/2357



Notes: 1. This pin has the WDTOVF pin function in the ZTAT, mask ROM, and ROMless versions. In the F-ZTAT version, the WDTOVF pin function is not available, and this pin is the FWE pin.

2. Applies to the H8S/2357 only.

5.5 H8S/2345 Series Features

The H8S/2345 Series is a 100-pin compact shrink version of the H8S/2655 Series. A varied package lineup is supported. A single-power-supply flash memory version, ZTAT version, and mask ROM version are available. These versions enable users to respond quickly and flexibly to changing application specifications, growing production volumes, and other conditions.

Product Lineup

Product Name	ROM (Bytes)	RAM (Bytes)	ROM Type*[1]
H8S/2345	128k	4k	MZF*[2]
H8S/2343	64k	2k	M

Notes: 1. M: Mask ROM version
Z: ZTAT version
F: F-ZTAT version
2. F-ZTAT version: Under development

High-Performance H8S/2000 CPU

- General register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- High-speed operation suitable for real-time control
  - All frequently used instructions execute in 1 to 2 states
  - Maximum operating frequency: 20 MHz
  - 8/16/32-bit register-register add/subtract: 50 ns
  - 8 × 8-bit register-register multiply: 600 ns
  - 16 ÷ 8-bit register-register divide: 600 ns
  - 16 × 16-bit register-register multiply: 1000 ns
  - 32 ÷ 16-bit register-register divide: 1000 ns
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Two CPU operating modes
  - Normal mode: H8/300 Series compatible, maximum 64-kbyte address space (available in ZTAT and mask ROM versions only)
  - Advanced mode: Maximum 16-Mbyte address space

Bus Controller

- Chip select output for areas 0 to 3

Data Transfer Controller (DTC)

Six 16-Bit Timer-Pulse Unit (TPU) Channels

Two 8-Bit Timer Channels

One Watchdog Timer (WDT) Channel

Two Serial Communication Interface (SCI) Channels

A/D Converter

- Resolution: 10 bits
- Input: 8 channels
- High-speed conversion: 6.7 µs (at 20 MHz operation)
- Selection of single or scan mode

D/A Converter

- Resolution: 8 bits
- Output: 2 channels

Eleven I/O Ports

- 71 I/O pins, 8 input-only pins

Interrupt Controller

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- 43 internal interrupt sources
- Selection of two interrupt control modes

Power-Down State

- Medium-speed mode
- Sleep mode
- Module stop mode
- Software standby mode
- Hardware standby mode

Eight MCU Operating Modes (F-ZTAT™ Version)

								External Data Bus	
MCU Operating Mode	FWE	MD_2	MD_1	MD_0	CPU Operating Mode	Description	On-Chip ROMe	Initial Value	Maximum Value
0	0	0	0	0	—	—	—	—	—
1				1					
2			1	0					
3				1					
4		1	0	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5				1				8 bits	16 bits
6			1	0		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7				1		Single-chip mode		—	—
8	1	0	0	0	—	—	—	—	—
9				1					
10			1	0	Advanced	Boot mode	Enabled	8 bits	16 bits
11				1				—	—
12		1	0	0	—	—	—	—	—
13				1					
14			1	0	Advanced	User program mode	Enabled	8 bits	16 bits
15				1				—	—

Seven MCU Operating Modes (ZTAT™ version and Mask ROM version)

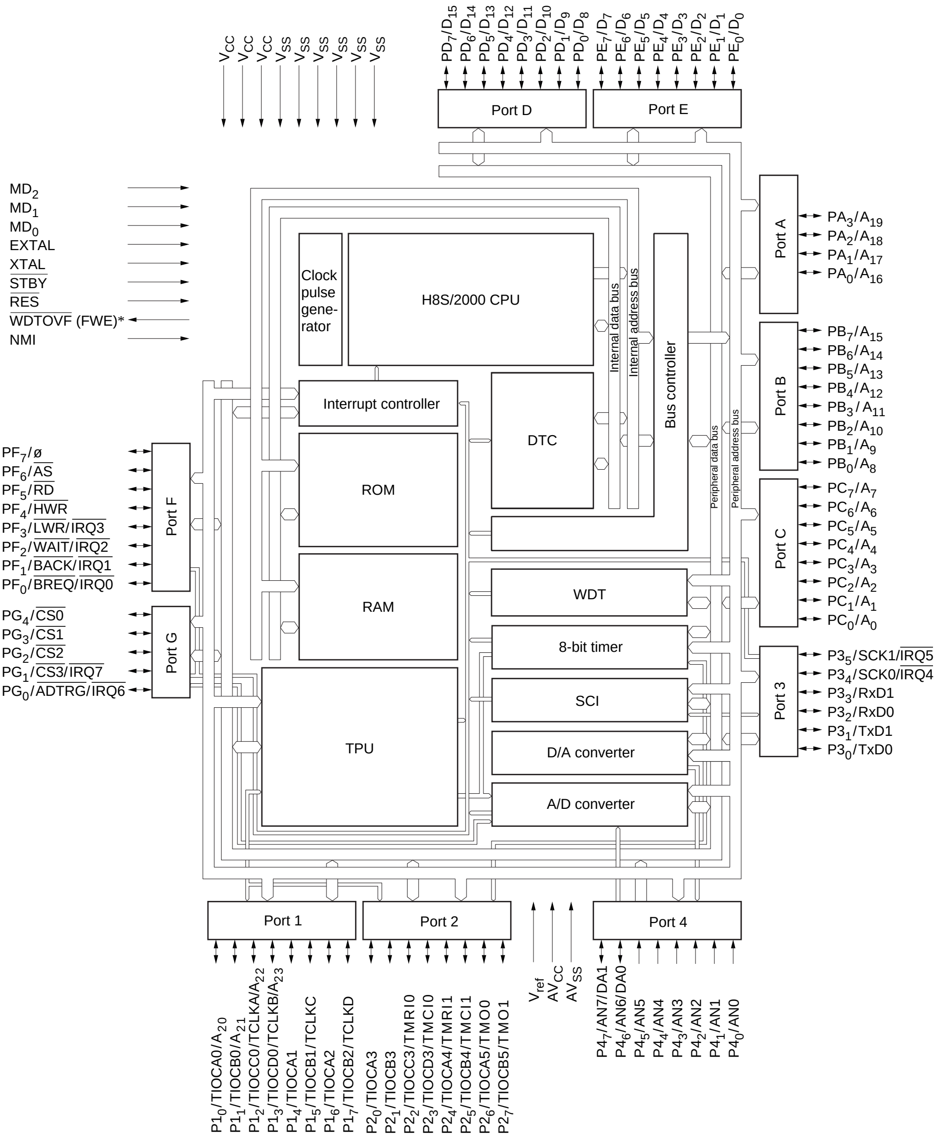
				External Data Bus	
Mode	**CPU Operating Mode**	**Description**	**On-Chip ROM**	**Initial Value**	**Maximum Value**
1	Normal	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
2		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
3		Single-chip mode	Enabled	—	—
4	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
5		Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
6		Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
7		Single-chip mode	Enabled	—	—

On-Chip Clock Pulse Generator (1:1 Oscillation)

Packages

- 100-pin plastic TQFP (TFP-100B, TFP-100G))
- 100-pin plastic QFP (FP-100A, FP-100B)

Internal Block Diagram of H8S/2345 Series



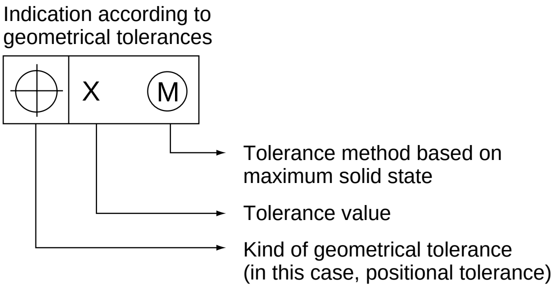
Note: * This pin has the $\overline{\text{WDTOVF}}$ pin function in the ZTAT, mask ROM, and ROMless versions. In the F-ZTAT version, the $\overline{\text{WDTOVF}}$ pin function is not available, and this pin is the FWE pin.

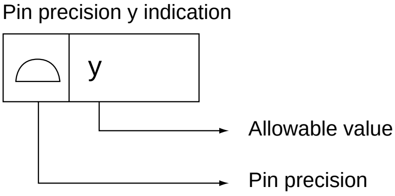
Appendix A List of Documents

		Product Series				
Document		**H8S/2655**	**H8S/2350**	**H8S/2355**	**H8S/2357**	**H8S/2345**
User's manuals	Hardware	ADE-602-094B	ADE-602-111A	ADE-602-112A	ADE-602-146	ADE-602-129
	Software	• H8S/2600 Series, H8S/2000 Series Programming Manual (ADE-602-083A)				
	Emulators	• E6000 H8S/2655, H8S/2245, H8S/2355, H8S/2350 Series Emulators (ADE-702-155)				
	Cross software	• H8S, H8/300 Series Simulator/Debugger (ADE-702-037B) • H8S, H8/300 Series C Compiler (ADE-702-059C) • H8S, H8/300 Series Cross Assembler (ADE-702-039C)				
Application notes		• H8S, H8/300 Series C Compiler Application Note (ADE-502-044) • H8S/2655 Series On-Chip Supporting Modules Application Note (ADE-502-048) • H8S Series Technical Q&A (ADE-502-059)				
Common documentation		Quick Reference Guide to Semiconductor Devices (ADE-301-001O)				

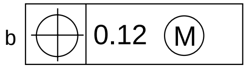
Appendix B Packages

Package Dimension Diagrams (Unit: mm)


Indication according to geometrical tolerances
X
M
Tolerance method based on maximum solid state
Tolerance value
Kind of geometrical tolerance (in this case, positional tolerance)



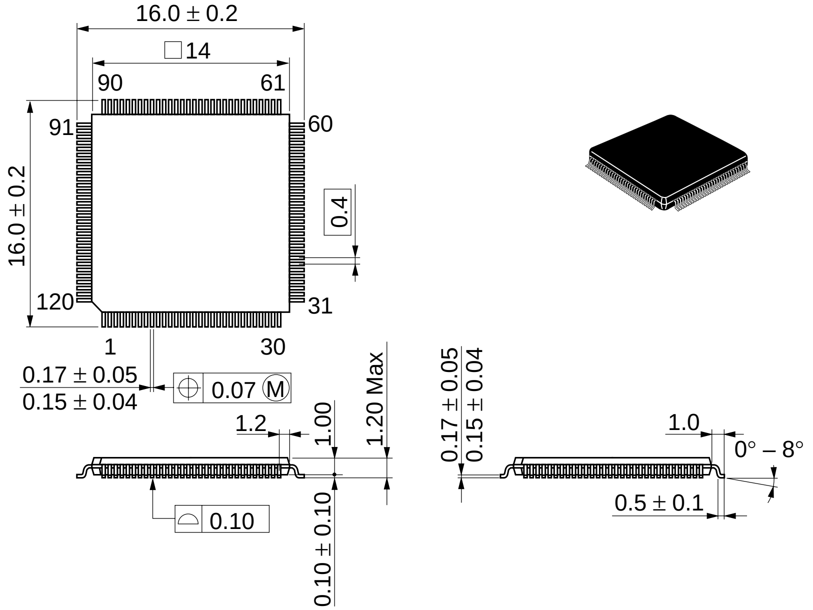
Example:



This indicates that the allowable pin displacement from the true central position is 0.12 mm when pin width b is the maximum dimension. If b is smaller than the maximum dimension, the tolerance can be extended accordingly.

TFP-120

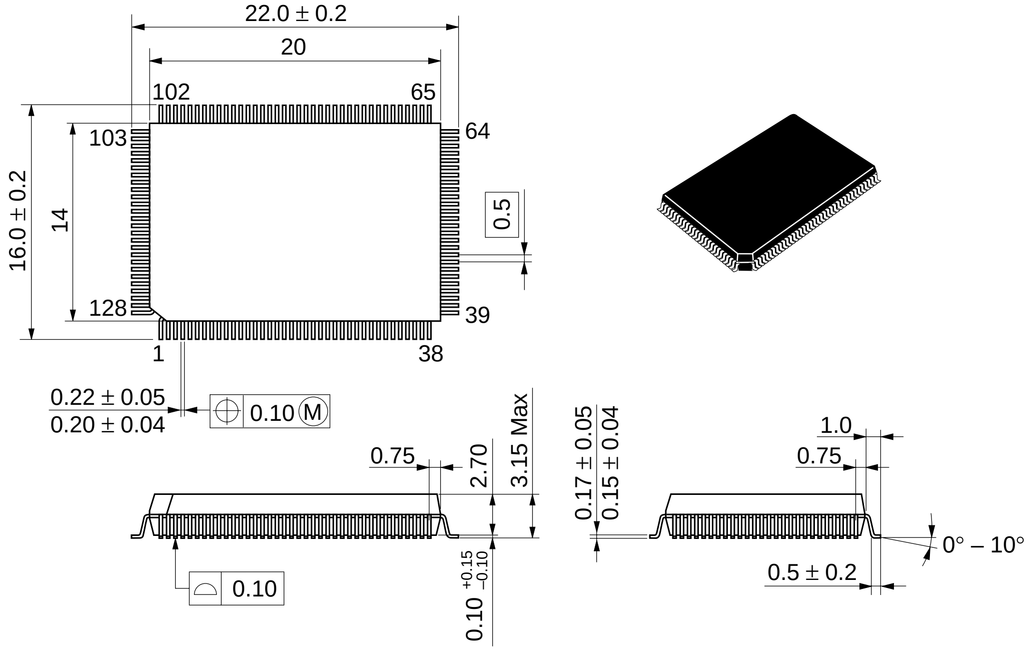
Unit: mm



Dimension including the plating thickness
Base material dimension

FP-128

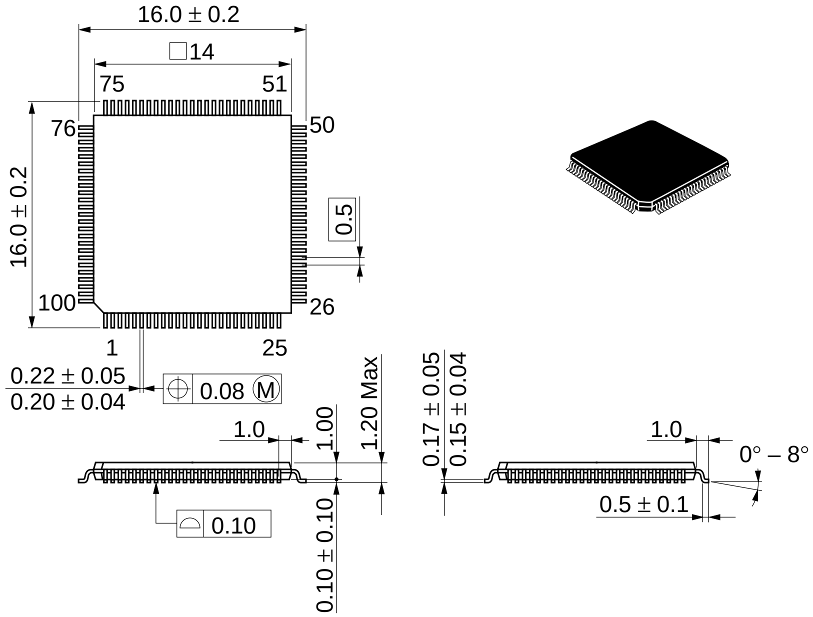
Unit: mm



Dimension including the plating thickness
Base material dimension

TFP-100B

Unit: mm

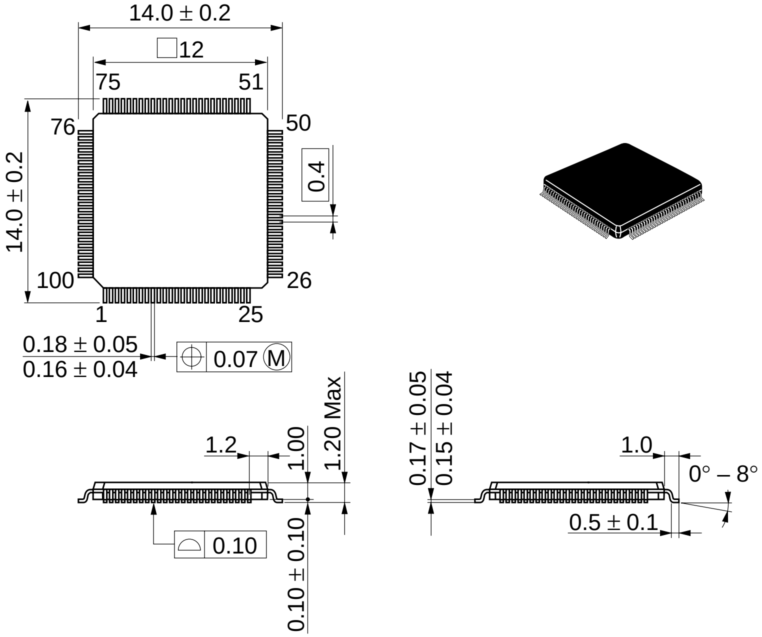


Dimension including the plating thickness

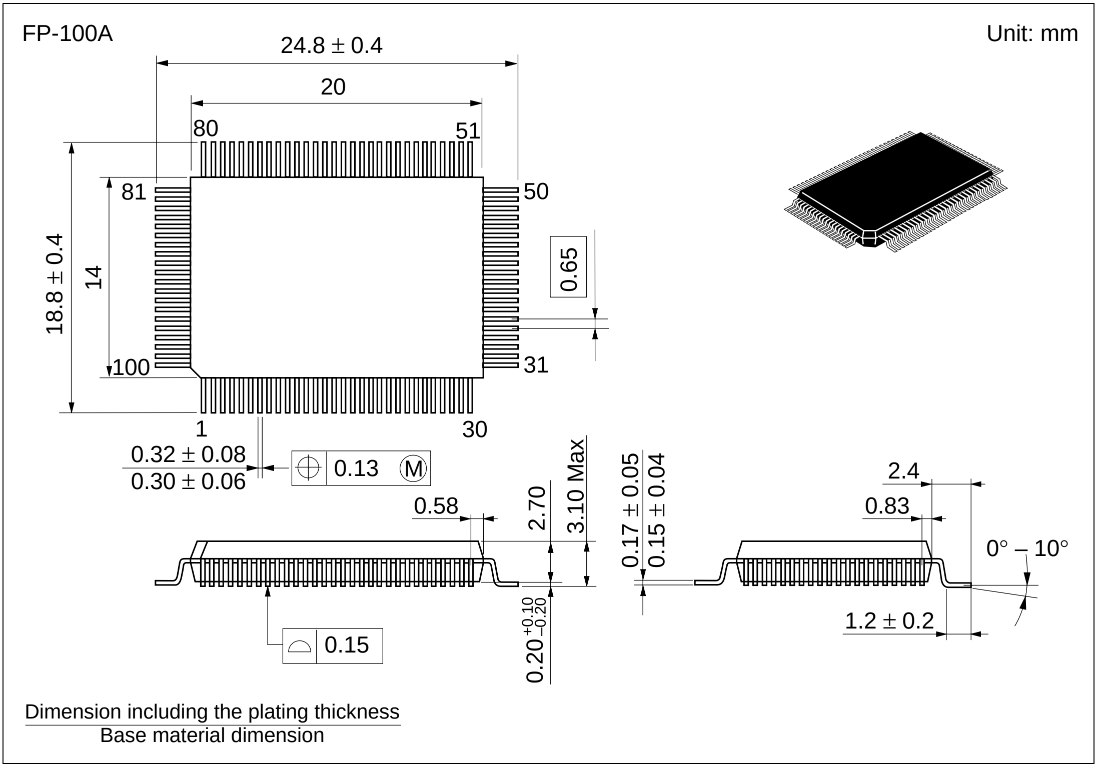
Base material dimension

TFP-100G

Unit: mm

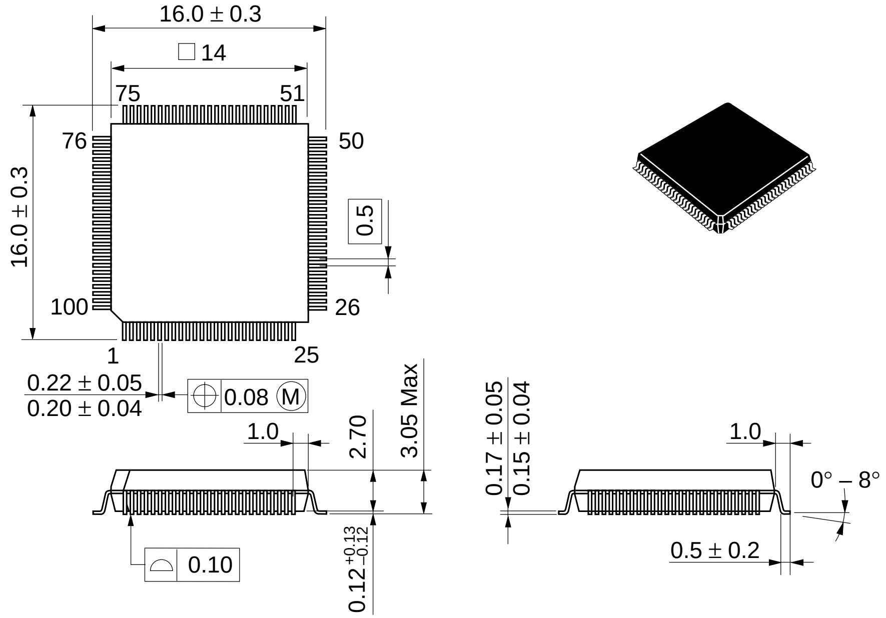


Dimension including the plating thickness
Base material dimension


FP-100A
Unit: mm
24.8 ± 0.4
20
80
51
81
50
18.8 ± 0.4
14
0.65
100
31
1
30
0.32 ± 0.08
0.30 ± 0.06
0.13 Ⓜ
0.58
2.70
3.10 Max
0.17 ± 0.05
0.15 ± 0.04
2.4
0.83
0° – 10°
1.2 ± 0.2
0.15
0.20 +0.10 −0.20
Dimension including the plating thickness
Base material dimension

FP-100B

Unit: mm



Dimension including the plating thickness
Base material dimension

H8S/2000 Series Overview

Publication Date: 1st Edition, March 1998
Published by: Electronic Devices Business Group
Hitachi, Ltd.
Edited by: Technical Documentation Center
Hitachi Microcomputer System Ltd.