H8S/2148 Series,
H8S/2144 Series,
H8S/2138 Series,
H8S/2134 Series

Overview

HITACHI

ADE-802-219
Rev. 1.1
12/97
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

Hitachi's H8S family of single-chip microcomputers comprises a number of new series offering the high performance and low power consumption of the existing H8 Series, which is widely used for machine control, etc., together with significantly greater ease of use.

The H8S/2000 Series features CPU object-level compatibility with the H8/300H Series, H8/300 Series, and H8/300L Series within the H8 Series.

Series	Features
H8S/2000	Upward-compatible with the H8/300H Series and H8/300 Series; twice the performance at the same frequency
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 supporting modules; low voltage, low power consumption

This document gives an overview of the new H8S/2000 Series products with on-chip flash memory (featuring single-power-supply programming): the 100-pin H8S/2148 Series and H8S/2144 Series, and the 80-pin H8S/2138 Series and H8S/2134 Series.

Intended Readership: This Overview is intended for readers who have a basic understanding of microcomputers, and are looking for information on the features and functions of the H8S/2148, H8S/2144, H8S/2138, and H8S/2134 Series. Readers undertaking system design using these products, or requiring more detailed information on their use, should refer to the relevant Hardware Manuals and the H8S/2600 and H8S/2000 Series Programming Manual.

Related Documents

Contents	Title	Document No.
H8S/2148 Series and H8S/2144 Series hardware	H8S/2148 Series and H8S/2144 Series Hardware Manual	TBD (Scheduled publication: 1/98)
H8S/2138 Series and H8S/2134 Series hardware	H8S/2138 Series and H8S/2134 Series Hardware Manual	TBD (Scheduled publication: 1/98)
H8S/2000 Series execution instructions	H8S/2600 Series and H8S/2000 Series Programming Manual	ADE-602-083A

The product specifications in this Overview are subject to change without notice. The relevant Hardware Manual must be used when undertaking product design.

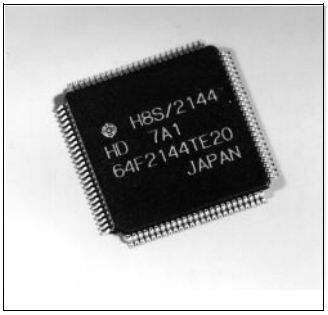
On-Chip Supporting Modules

Series	H8S/2148 Series (100-Pin)	H8S/2144 Series (100-Pin)	H8S/2138 Series (80-Pin)	H8S/2134 Series (80-Pin)
Product names	H8S/2148, 2147	H8S/2144, 2143, 2142	H8S/2138, 2137	H8S/2134, 2133, 2132, 2130
Bus controller (BSC)	Available (16-bit)	Available (16-bit)	Available (8-bit)	Available (8-bit)
Data transfer controller (DTC)	Available	—	Available	—
8-bit PWM timer (PWM)	×16	—	×16	—
14-bit PWM timer (PWMX)	×2	×2	×2	×2
16-bit free-running timer (FRT)	×1	×1	×1	×1
8-bit timer (TMR)	×4	×3	×4	×3
Timer connection	Available	—	Available	—
Watchdog timer (WDT)	×2	×2	×2	×2
Serial communication interface (SCI)	×3	×3	×3	×3
I^2C bus interface (IIC)	×2 (option)	—	×2 (option)	—
Keyboard buffer controller (PS/2 compatible)	×3	—	—	—
Host interface (HIF)	Available	—	Available	—
D/A converter	×2	×2	×2	×2
A/D converter	×8	×8	×8	×8
Comparator	×16	×16	×8	×8

Contents

Section 1 Features of H8S/2148 Series and H8S/2144 Series

1.1 Features of H8S/2148 Series and H8S/2144 Series



H8S/2148 Series and H8S/2144 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 comprehensive on-chip supporting modules needed for control systems, simplifying the implementation of sophisticated, high-performance systems.

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 realtime control
  - — 20 MHz maximum operating frequency (20 MHz oscillation frequency)
  - — High-speed arithmetic operations (at 20 MHz operation)
    - 8/16/32-bit register-register add/subtract: 50 ns
    - 16 × 16-bit register-register multiply: 1000 ns
    - 32 ÷ 16-bit register-register divide: 1000 ns
- Instruction set suitable for high-speed operation
  - — Sixty-five types of basic instructions
  - — 8/16/32-bit transfer instructions
  - — Unsigned/signed 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

- 2-state or 3-state access space can be designated for external expansion areas
- Number of program wait states can be set for external expansion areas

Data transfer controller (DTC) (H8S/2148 Series)

- Can be 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 free-running timer (FRT: 1 channel)

- One 16-bit free-running counter (also usable for external event counting)
- Two output compare outputs
- Four input capture inputs (with buffer operation capability)

8-bit timer (2 channels: TMR0, TMR1)

Each channel has:

- One 8-bit up-counter (also usable for external event counting)
- Two timer constant registers
- The two channels can be connected

Timer connection and 8-bit timer (2 channels: TMRX, TMRY)
(Timer connection and TMRX provided in H8S/2148 Series)

Input/output and FRT, TMR1, TMRX, TMRY can be inter connected

- Measurement of input signal or frequency-divided waveform pulse width and cycle (FRT, TMR1)
- Output of waveform obtained by modification of input signal edge (FRT, TMR1)
- Determination of input signal duty cycle (TMRX)
- Output of waveform synchronized with input signal (FRT, TMRX, TMRY)
- Automatic generation of cyclical waveform (FRT, TMRY)

Watchdog timer (WDT: 2 channels)

- Watchdog timer or interval timer function selectable
- Subclock operation capability (channel 1 only)

8-bit PWM timer (PWM) (Included in the H8S/2148 Series)

- Up to 16 outputs
- Pulse duty cycle settable from 0 to 100%
- Resolution: 1/256
- 1.25 MHz maximum carrier frequency (20 MHz operation)

14-bit PWM timer (PWMX)

- Up to 2 outputs
- Resolution: 1/16384
- 312.5 kHz maximum carrier frequency (20 MHz operation)

Serial communication interface (SCI: 2 channels, SCI0 and SCI1)

- Asynchronous mode or synchronous mode selectable
- Multiprocessor communication function

SCI with IrDA: 1 channel (SCI2)

- Compatible with IrDA specification version 1.0
- TxD and RxD encoding/decoding in IrDA format

Keyboard buffer controller (three channels) (H8S/2148 Series)

- Compatible with PS/2 interface
- Direct manipulation of transmission output by software
- Receive data input to 8-bit shift register
- Data receive completed interrupt, parity error detection, stop bit monitoring

Host interface (HIF) (H8S/2148 Series)

- 8-bit host interface (ISA) port
- Three host interrupt requests (HIRQ11, HIRQ1, HIRQ12)
- Normal and fast A20 gate output
- Two register sets (comprising two data registers and two status registers)

Keyboard controller

- Matrix keyboard control using keyboard scan with wakeup interrupt and sense port configuration

Comparator

- Comparator using A/D converter
- Resolution: 6 bits
- Input: 16 channels (same pins as keyboard sense port)

A/D converter

- Resolution: 10 bits
- Input: 8 channels

- High-speed conversion: 6.7 µs minimum conversion time (20 MHz operation)
- Single or scan mode selectable
- Sample-and-hold function
- A/D conversion can be activated by external trigger or timer trigger

D/A converter

- Resolution: 8 bits
- Output: 2 channels

I/O ports

- 74 input/output pins (including 24 with LED drive capability)
- 8 input-only pins

Memory

- Flash memory or mask ROM
- High-speed static RAM

Product Name	ROM	RAM
H8S/2148, H8S/2144	128 kbytes	4 kbytes
H8S/2143	96 kbytes	4 kbytes
H8S/2147, H8S/2142	64 kbytes	2 kbytes

Interrupt controller

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- 42 internal interrupt sources
- Three priority levels settable

Power-down state

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

I^2C bus interface (IIC: 2 channels) (option) (H8S/2148 Series)

- Conforms to Philips I^2C bus interface standard

- Single master mode/slave mode
- Arbitration lost condition can be identified
- Supports two slave addresses

Three MCU operating modes

Mode	CPU Operating Mode	Description	On-Chip ROM
1	Normal	Expanded mode with on-chip ROM disabled	Disabled
2	Advanced	Expanded mode with on-chip ROM enabled Single-chip mode	Enabled
3	Normal	Expanded mode with on-chip ROM enabled Single-chip mode	Enabled

On-chip clock pulse generator (1:1 oscillation)

- Built-in duty adjustment circuit

Packages

- 100-pin plastic QFP (FP-100B)
- 100-pin plastic TQFP (TFP-100B)

Product Lineup (Preliminary)

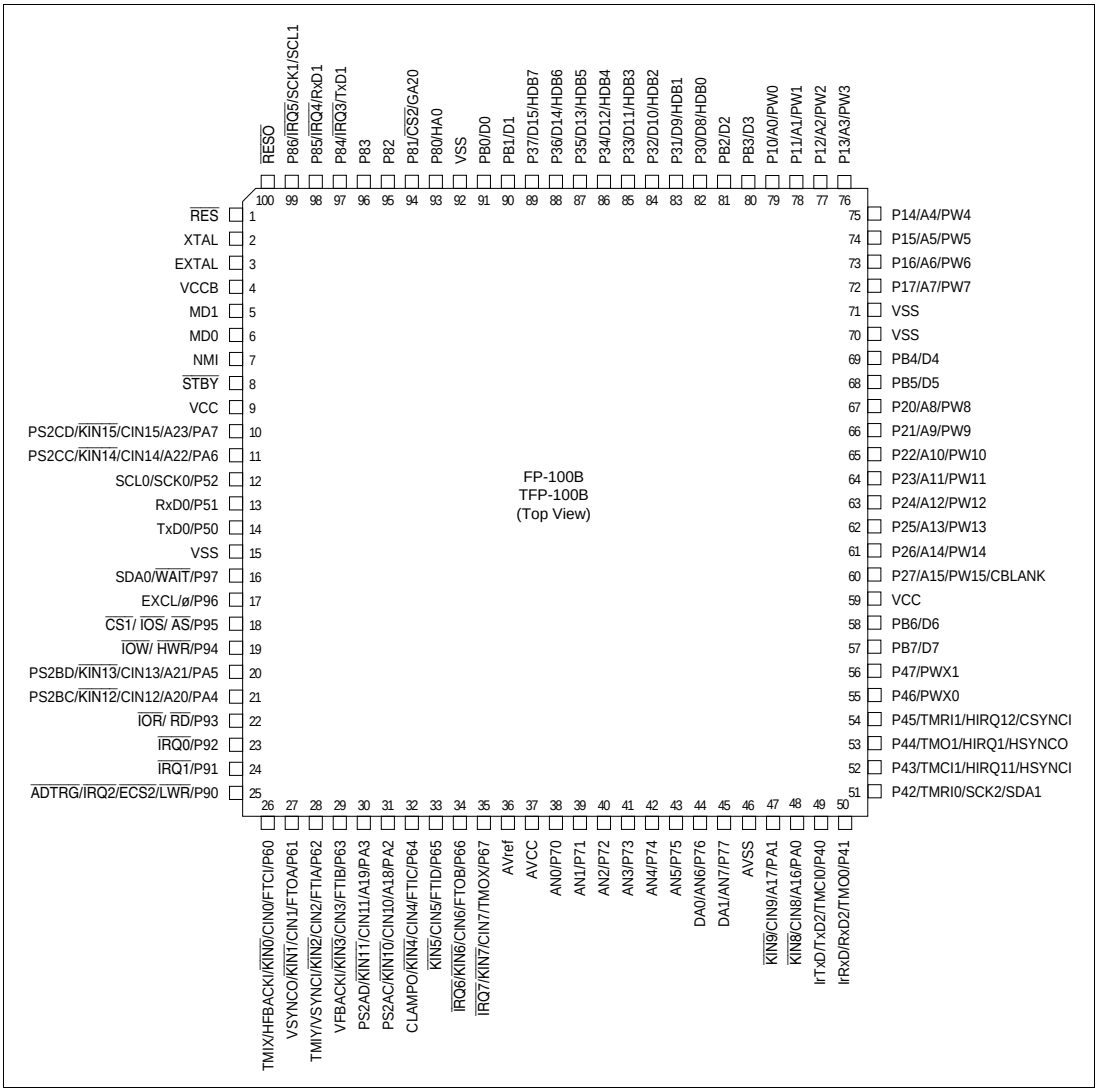
	Product Name		ROM/RAM	
Series	Mask ROM Version	F-ZTAT™ Version	(Bytes)	Package
H8S/2148 Series	HD6432148 HD6432148W*	HD64F2148*	128 k/4 k	FP-100B, TFP-100B
	HD6432147 HD6432147W*	—	64 k/2 k	
H8S/2144 Series	HD6432144	HD64F2144	128 k/4 k	
	HD6432143	—	96 k/4 k	
	HD6432142	HD64F2142	64 k/2 k	

Note: * "W" indicates the I^2C bus option.

1.2 Pin Arrangement and Functions

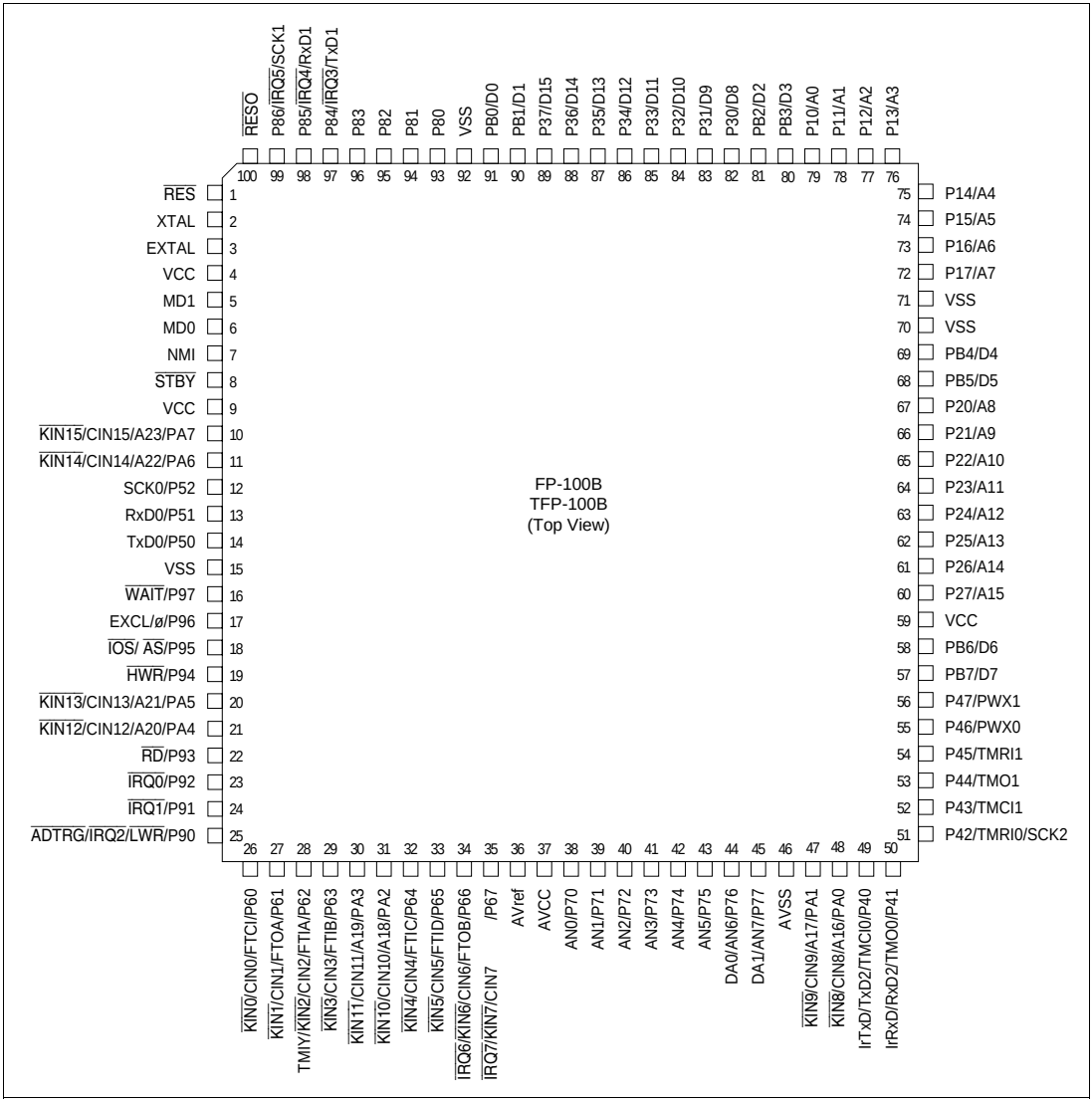
H8S/2148 Series Pin Arrangement

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



H8S/2144 Series Pin Arrangement

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



Pin Functions

Type	Symbol	I/O	Name and Function
Power	VCC	Input	Power supply
	VCCB	Input	I/O buffer power supply (port A)
	VSS	Input	Ground For connection to the power supply (0 V)
Clock	XTAL	Input	For connection to a crystal oscillator
	EXTAL	Input	Crystal oscillator or external clock input
	ø	Output	System clock
	EXCL	Input	External subclock input Inputs 32.768 kHz external subclock
Operating mode control	MD1 MD0	Input	Mode setting
System control	$\overline{\text{RES}}$	Input	Reset input
	$\overline{\text{RESO}}$	Output	Reset output
	$\overline{\text{STBY}}$	Input	Standby
Address bus	A23 to A16	Output	Address bus (advanced) when using 16-Mbyte space
	A15 to A0	Output	Address bus
Data bus	D15 to D8	Input/output	Data bus (upper)
	D7 to D0	Input/output	Data bus (lower)
Bus control	$\overline{\text{WAIT}}$	Input	Wait
	$\overline{\text{RD}}$	Output	Read
	$\overline{\text{HWR}}$	Output	High write
	$\overline{\text{LWR}}$	Output	Low write
	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	Output	Address strobe
Interrupts	NMI	Input	Nonmaskable interrupt
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$	Input	Interrupt requests 0 to 7

Pin Functions (cont)

Type	Symbol	I/O	Name and Function
16-bit free-running timer (FRT)	FTCI	Input	FRT counter clock input
	FTOA	Output	FRT output compare A output
	FTOB	Output	FRT output compare B output
	FTIA	Input	FRT input capture A input
	FTIB	Input	FRT input capture B input
	FTIC	Input	FRT input capture C input
	FTID	Input	FRT input capture D input
8-bit timers (TMR0, TMR1, TMRX, TMRY)	TMO0 TMO1 TMOX	Output	Compare match output
	TMCI0 TMCI1	Input	Counter external clock input
	TMRI0 TMRI1	Input	Counter external reset input
	TMIX TMIY	Input	Counter external clock input and reset input
PWM timer (PWM)	PW15 to PW0	Output	PWM timer output
14-bit PWM timer (PWMX)	PWX0 PWX1	Output	PWM timer output
Serial communi-cation interface (SCI0, SCI1, SCI2)	TxD0 TxD1 TxD2	Output	Transmit data
	RxD0 RxD1 RxD2	Input	Receive data
	SCK0 SCK1 SCK2	Input/output	Serial clock
SCI with IrDA (SCI2)	IrTxD	Output	IrDA transmit data
	IrRxD	Input	IrDA receive data

Pin Functions (cont)

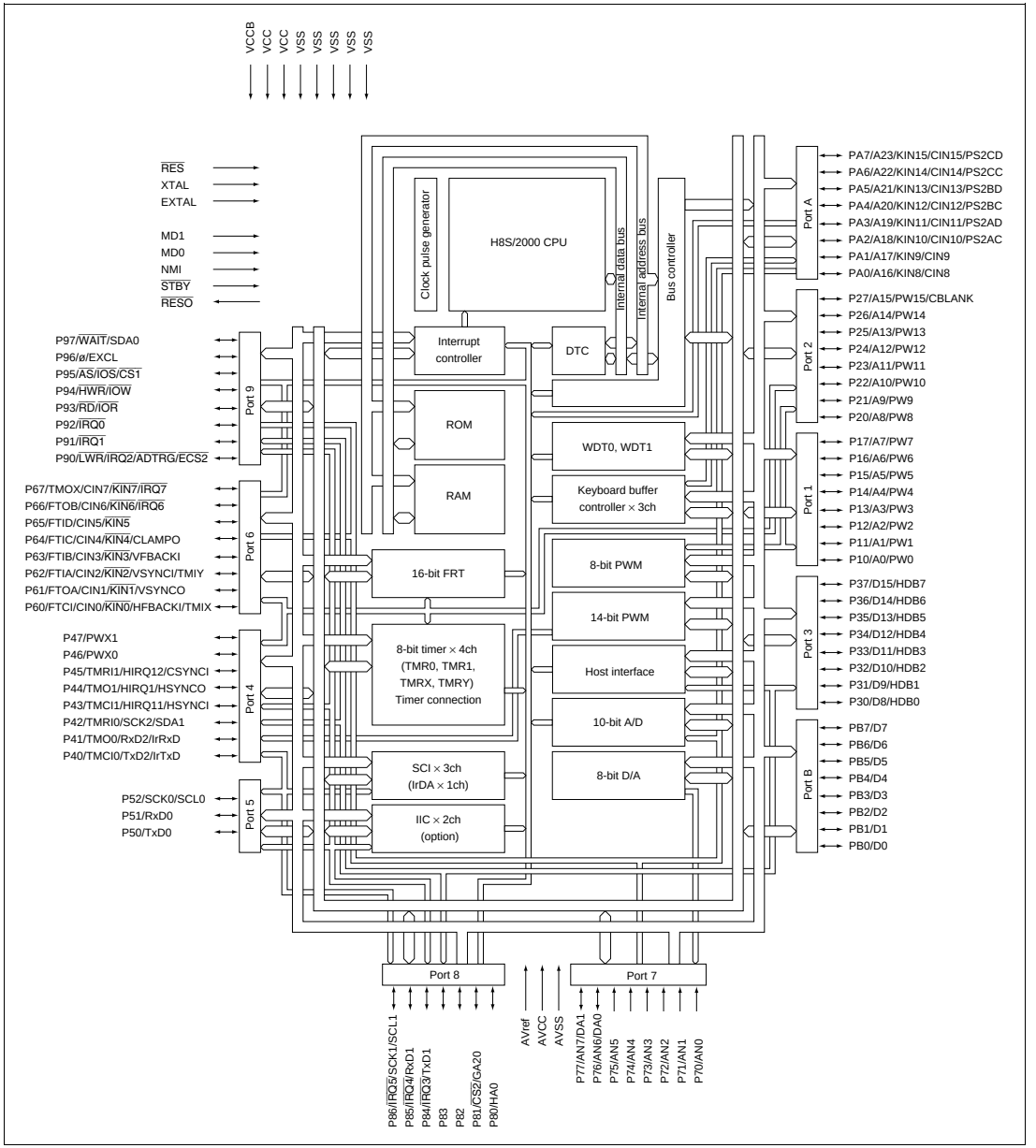
Type	Symbol	I/O	Name and Function
Keyboard buffer controller (PS/2 compatible)	PS2AC PS2BC PS2CC	Input/output	PS2 clock
	PS2AD PS2BD PS2CD	Input/output	PS2 data
Host interface (HIF)	HDB7 to HDB0	Input/output	Host interface data bus
	$\overline{CS1}$, $\overline{CS2}$ $\overline{ECS2}$	Input	Chip select 1, 2
	$\overline{IOR}$	Input	I/O read
	$\overline{IOW}$	Input	I/O write
	HA0	Input	Command/data
	GA20	Output	Gate A20 control signal output
	HIRQ11 HIRQ1 HIRQ12	Output	Host interrupts 11, 1, 12
Keyboard control	$\overline{KIN0}$ to $\overline{KIN15}$	Input	Keyboard input
Comparator	CIN0 to CIN15	Input	Comparator input
A/D converter (ADC)	AN7 to AN0	Input	Analog input
	$\overline{ADTRG}$	Input	A/D conversion external trigger input
D/A converter (DAC)	DA0 DA1	Output	Analog output
A/D converter	AVCC	Input	Analog power supply
D/A converter	AVref	Input	Analog reference voltage
	AVSS	Input	Analog ground Connect to the system power supply (0 V)

Pin Functions (cont)

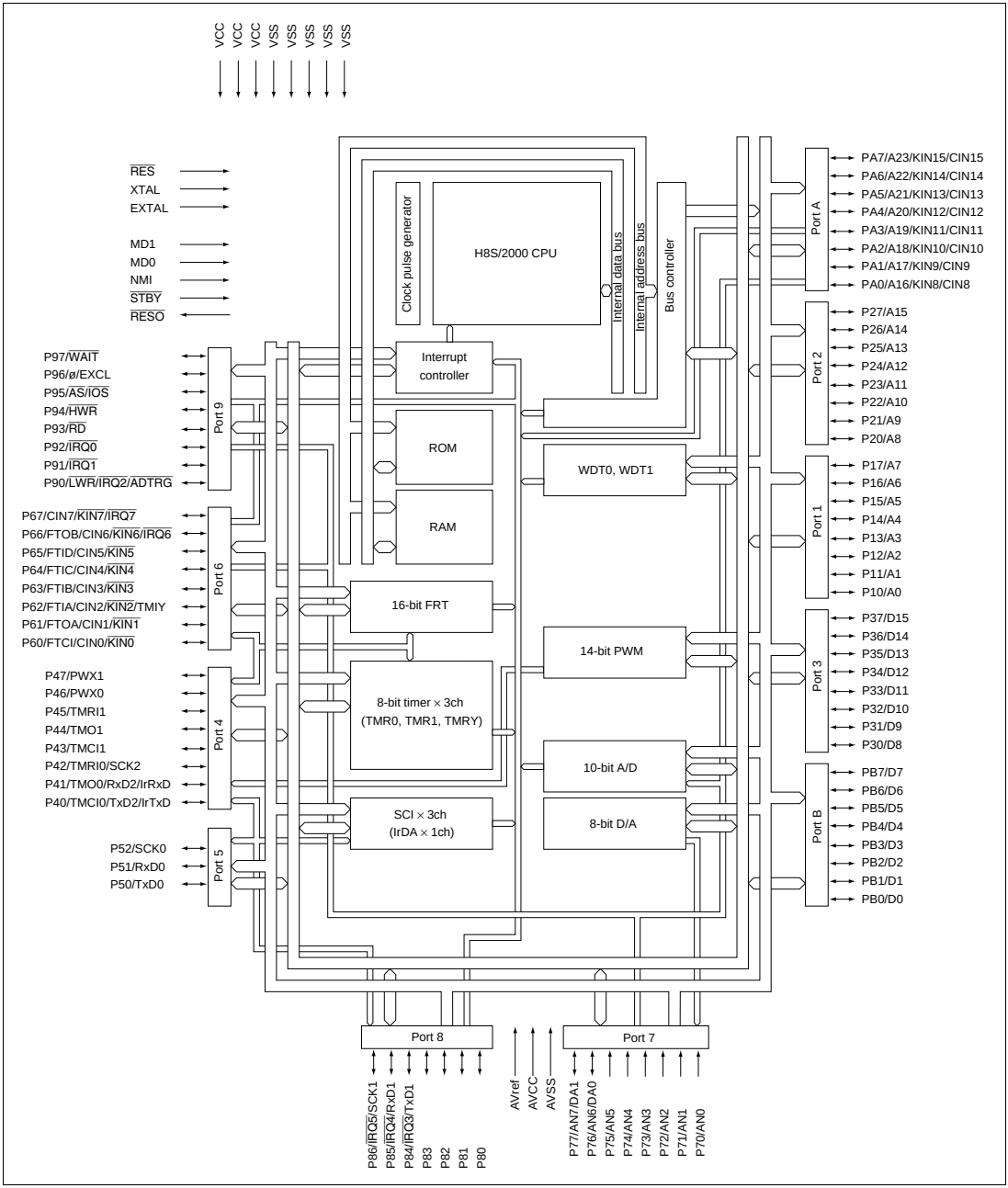
Type	Symbol	I/O	Name and Function
Timer connection	VSYNCI	Input	Timer connection input
	HSYNCI		
	CSYNCI		
	VFBACKI		
	HFBACKI		
	VSYNCO	Output	Timer connection output
	HSYNCO		
	CLAMPO		
	CBLANK		
I^2C bus interface (IIC) (option)	SCL0	Input/output	IIC clock input/output (channels 0, 1)
	SCL1		
	SDA0	Input/output	IIC data input/output (channels 0, 1)
	SDA1		
I/O ports	P17 to P10	Input/output	Port 1
	P27 to P20	Input/output	Port 2
	P37 to P30	Input/output	Port 3
	P47 to P40	Input/output	Port 4
	P52 to P50	Input/output	Port 5
	P67 to P60	Input/output	Port 6
	P77 to P70	Input	Port 7
	P86 to P80	Input/output	Port 8
	P97 to P90	Input/output	Port 9
	PA7 to PA0	Input/output	Port A
	PB7 to PB0	Input/output	Port B

1.3 Internal Block Diagram

Internal Block Diagram of H8S/2148 Series

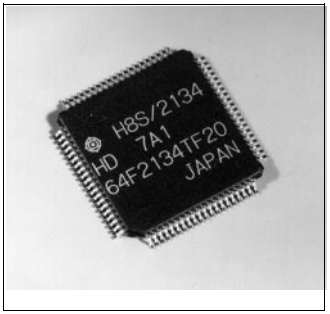


Internal Block Diagram of H8S/2144 Series



Section 2 Features of H8S/2138 Series and H8S/2134 Series

2.1 Features of H8S/2138 Series and H8S/2134 Series



H8S/2138 Series and H8S/2134 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 comprehensive on-chip supporting modules needed for control systems, simplifying the implementation of sophisticated, high-performance systems.

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 realtime control
  - — 20 MHz maximum operating frequency (20 MHz oscillation frequency)
  - — High-speed arithmetic operations (at 20 MHz operation)
    - 8/16/32-bit register-register add/subtract: 50 ns
    - 16 × 16-bit register-register multiply: 1000 ns
    - 32 ÷ 16-bit register-register divide: 1000 ns
- Instruction set suitable for high-speed operation
  - — Sixty-five types of basic instructions
  - — 8/16/32-bit transfer instructions
  - — Unsigned/signed 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

- 2-state or 3-state access space can be designated for external expansion areas
- Number of program wait states can be set for external expansion areas

Data transfer controller (DTC) (H8S/2138 Series)

- Can be 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 free-running timer (FRT: 1 channel)

- One 16-bit free-running counter (also usable for external event counting)
- Two output compare outputs
- Four input capture inputs (with buffer operation capability)

8-bit timer (2 channels: TMR0, TMR1)

Each channel has:

- One 8-bit up-counter (also usable for external event counting)
- Two timer constant registers
- The two channels can be connected

Timer connection and 8-bit timer (2 channels: TMRX, TMRY)
(Timer connection and TMRX provided in H8S/2138 Series)

Input/output and FRT, TMR1, TMRX, TMRY can be inter connected

- Measurement of input signal or frequency-divided waveform pulse width and cycle (FRT, TMR1)
- Output of waveform obtained by modification of input signal edge (FRT, TMR1)
- Determination of input signal duty cycle (TMRX)
- Output of waveform synchronized with input signal (FRT, TMRX, TMRY)
- Automatic generation of cyclical waveform (FRT, TMRY)

Watchdog timer (WDT: 2 channels)

- Watchdog timer or interval timer function selectable
- Subclock operation capability (channel 1 only)

8-bit PWM timer (PWM) (Included in the H8S/2138 Series)

- Up to 16 outputs
- Pulse duty cycle settable from 0 to 100%
- Resolution: 1/256
- 1.25 MHz maximum carrier frequency (20 MHz operation)

14-bit PWM timer (PWMX)

- Up to 2 outputs
- Resolution: 1/16384
- 312.5 kHz maximum carrier frequency (20 MHz operation)

Serial communication interface (SCI: 2 channels, SCI0 and SCI1)

- Asynchronous mode or synchronous mode selectable
- Multiprocessor communication function

SCI with IrDA: 1 channel (SCI2)

- Compatible with IrDA specification version 1.0
- TxD and RxD encoding/decoding in IrDA format

Host interface (HIF) (H8S/2138 Series)

- 8-bit host interface (ISA) port
- Three host interrupt requests (HIRQ11, HIRQ1, HIRQ12)
- Normal and fast A20 gate output
- Two register sets (comprising two data registers and two status registers)

Keyboard controller

- Matrix keyboard control using keyboard scan with wakeup interrupt and sense port configuration

Comparator

- Comparator using A/D converter
- Resolution: 6 bits
- Input: 8 channels (same pins as keyboard sense port)

A/D converter

- Resolution: 10 bits
- Input: 8 channels
- High-speed conversion: 6.7 μs minimum conversion time (20 MHz operation)
- Single or scan mode selectable
- Sample-and-hold function
- A/D conversion can be activated by external trigger or timer trigger

D/A converter

- Resolution: 8 bits
- Output: 2 channels

I/O ports

- 58 input/output pins (including 24 with LED drive capability)
- 8 input-only pins

Memory

- Flash memory or mask ROM
- High-speed static RAM

Product Name	ROM	RAM
H8S/2138, H8S/2134	128 kbytes	4 kbytes
H8S/2133	96 kbytes	4 kbytes
H8S/2137, H8S/2132	64 kbytes	2 kbytes
H8S/2130	32 kbytes	2 kbytes

Interrupt controller

- Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$)
- 39 internal interrupt sources
- Three priority levels settable

Power-down state

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

I^2C bus interface (IIC: 2 channels) (option) (H8S/2138 Series)

- Conforms to Philips I^2C bus interface standard
- Single master mode/slave mode
- Arbitration lost condition can be identified
- Supports two slave addresses

Three MCU operating modes

Mode	CPU Operating Mode	Description	On-Chip ROM
1	Normal	Expanded mode with on-chip ROM disabled	Disabled
2	Advanced	Expanded mode with on-chip ROM enabled Single-chip mode	Enabled
3	Normal	Expanded mode with on-chip ROM enabled Single-chip mode	Enabled

On-chip clock pulse generator (1:1 oscillation)

- Built-in duty adjustment circuit

Packages

- 80-pin plastic QFP (FP-80A)
- 80-pin plastic TQFP (TFP-80C)

Product Lineup (Preliminary)

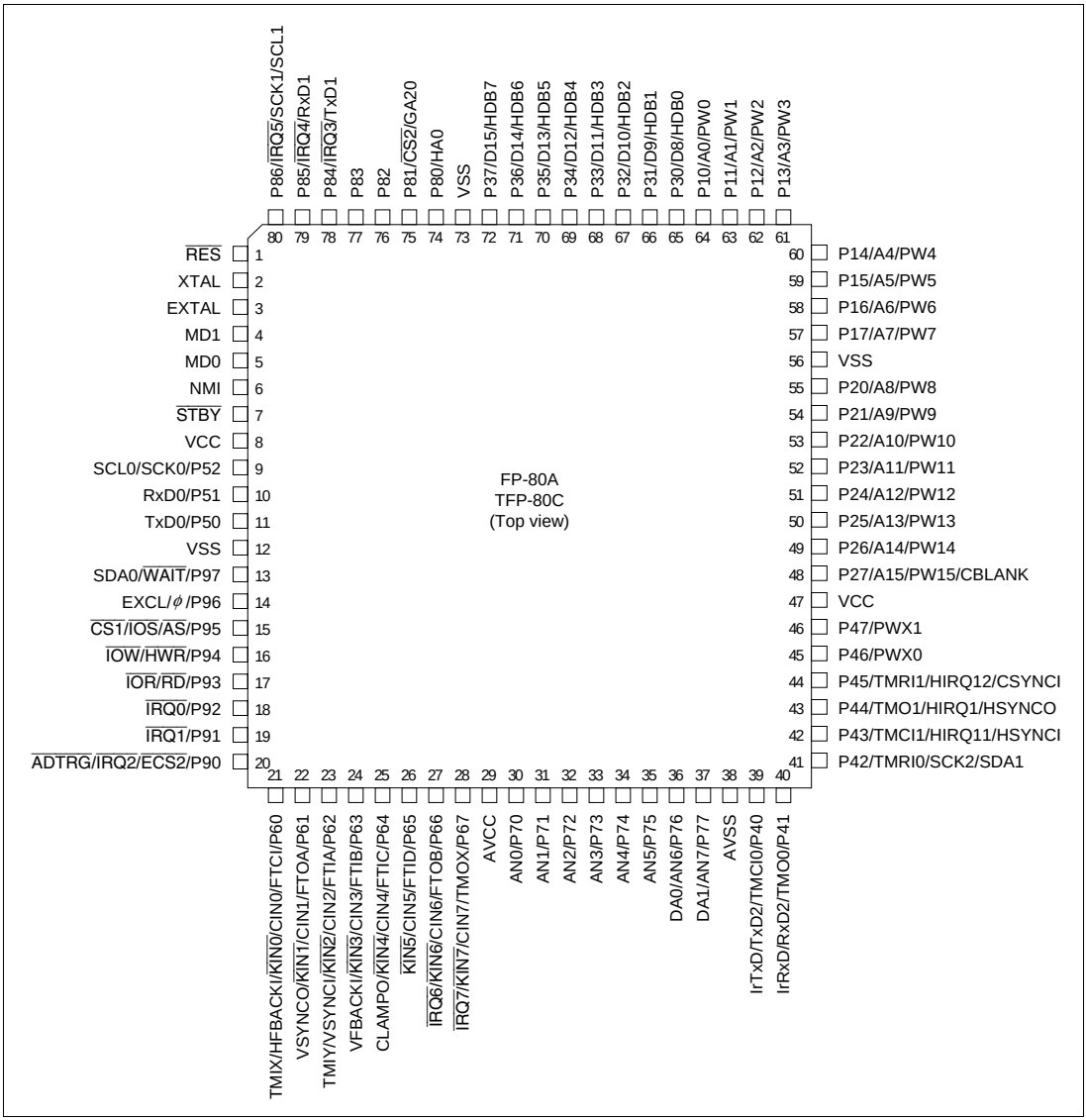
	Product Name		ROM/RAM	
Series	Mask ROM Version	F-ZTAT™ Version	(Bytes)	Package
H8S/2138 Series	HD6432138 HD6432138W*	HD64F2138*	128 k/4 k	FP-80A, TFP-80C
	HD6432137 HD6432137W*	—	64 k/2 k	
H8S/2134 Series	HD6432134	HD64F2134	128 k/4 k	
	HD6432133	—	96 k/4 k	
	HD6432132	HD64F2132	64 k/2 k	
	HD6432130	—	32 k/2 k	

Note: * “W” indicates the I^2C bus option.

2.2 Pin Arrangement and Functions

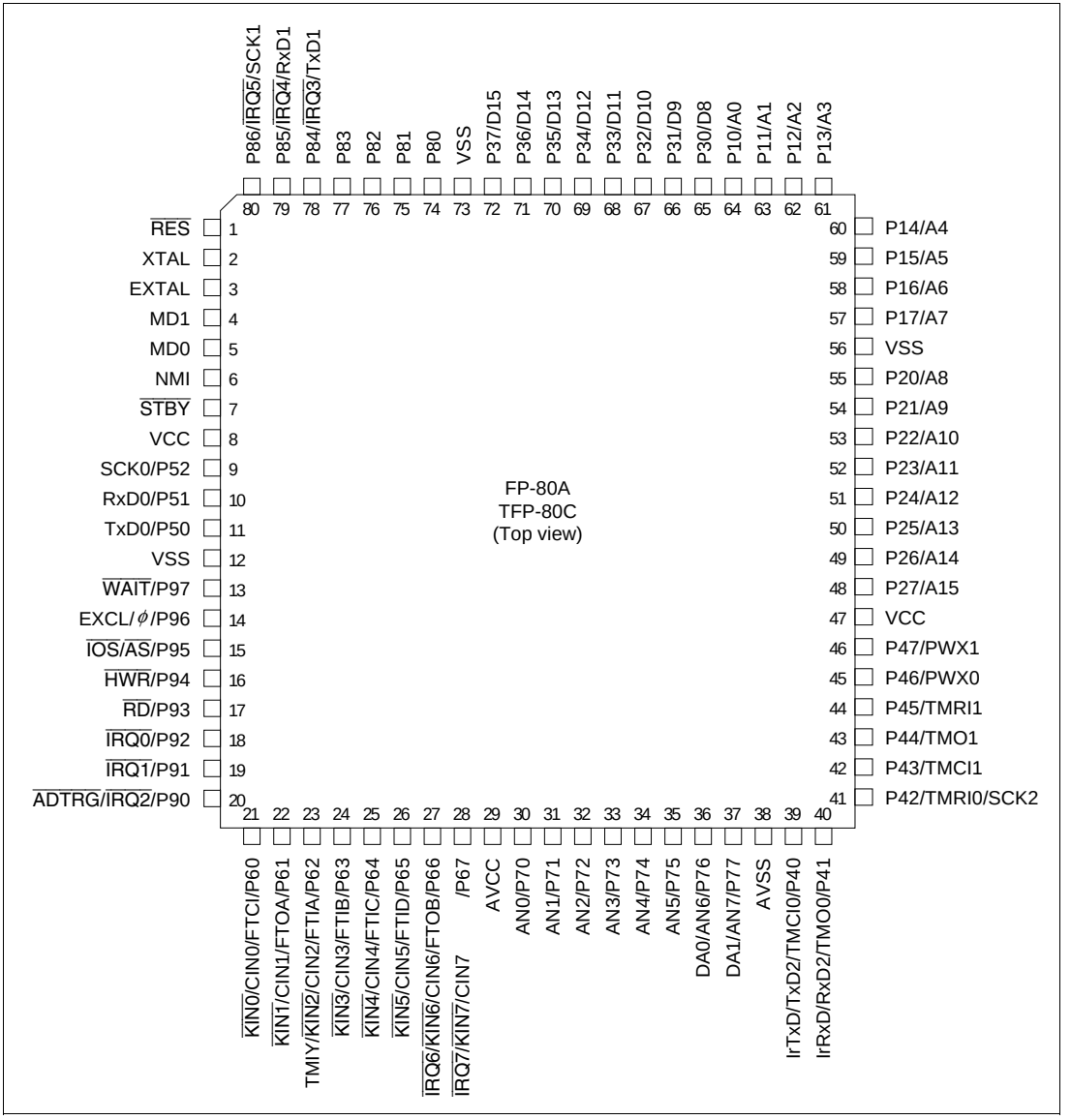
H8S/2138 Series Pin Arrangement

- 80-pin plastic QFP (FP-80A), 80-pin plastic TQFP (TFP-80C)



H8S/2134 Series Pin Arrangement

- 80-pin plastic QFP (FP-80A), 80-pin plastic TQFP (TFP-80C)



Pin Functions

Type	Symbol	I/O	Name and Function
Power	VCC	Input	Power supply
	VSS	Input	Ground For connection to the power supply (0 V)
Clock	XTAL	Input	For connection to a crystal oscillator
	EXTAL	Input	Crystal oscillator or external clock input
	ϕ	Output	System clock
	EXCL	Input	External subclock input Inputs 32.768 kHz external subclock
Operating mode control	MD1 MD0	Input	Mode setting
System control	$\overline{\text{RES}}$	Input	Reset input
	$\overline{\text{STBY}}$	Input	Standby
Address bus	A15 to A0	Output	Address bus
Data bus	D15 to D8	Input/output	Data bus (upper)
Bus control	$\overline{\text{WAIT}}$	Input	Wait
	$\overline{\text{RD}}$	Output	Read
	$\overline{\text{HWR}}$	Output	High write
	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	Output	Address strobe
Interrupts	NMI	Input	Nonmaskable interrupt
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$	Input	Interrupt requests 0 to 7
16-bit free-running timer (FRT)	FTCI	Input	FRT counter clock input
	FTOA	Output	FRT output compare A output
	FTOB	Output	FRT output compare B output
	FTIA	Input	FRT input capture A input
	FTIB	Input	FRT input capture B input
	FTIC	Input	FRT input capture C input
	FTID	Input	FRT input capture D input
8-bit timers (TMR0, TMR1, TMRX, TMRY)	TMO0 TMO1 TMOX	Output	Compare match output
	TMCI0 TMCI1	Input	Counter external clock input

Pin Functions (cont)

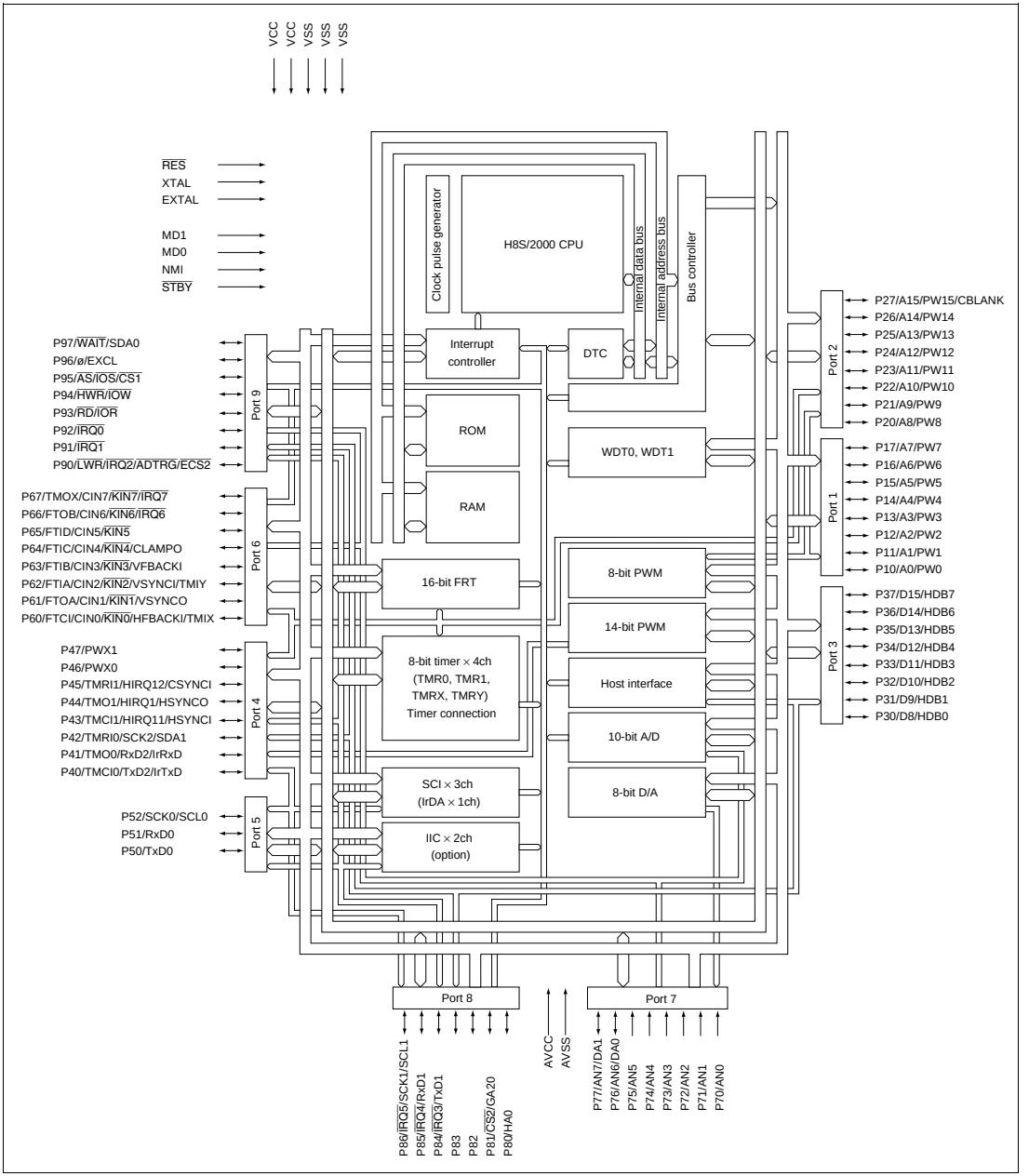
Type	Symbol	I/O	Name and Function
8-bit timers (TMR0, TMR1, TMRX, TMRY)	TMRI0 TMRI1	Input	Counter external reset input
	TMIX TMIY	Input	Counter external clock input and reset input
PWM timer (PWM)	PW15 to PW0	Output	PWM timer output
14-bit PWM timer (PWMX)	PWX0 PWX1	Output	PWM timer output
Serial communication interface (SCI0, SCI1, SCI2)	TxD0 TxD1 TxD2	Output	Transmit data
	RxD0 RxD1 RxD2	Input	Receive data
	SCK0 SCK1 SCK2	Input/output	Serial clock
SCI with IrDA (SCI2)	IrTxD	Output	IrDA transmit data
	IrRxD	Input	IrDA receive data
Host interface (HIF)	HDB7 to HDB0	Input/output	Host interface data bus
	$\overline{CS1}$, $\overline{CS2}$ $\overline{ECS2}$	Input	Chip select 1, 2
	$\overline{IOR}$	Input	I/O read
	$\overline{IOW}$	Input	I/O write
	HA0	Input	Command/data
	GA20	Output	Gate A20 control signal output
	HIRQ11 HIRQ1 HIRQ12	Output	Host interrupts 11, 2, 12
Keyboard control	$\overline{KIN0}$ to $\overline{KIN7}$	Input	Keyboard input
Comparator	CIN0 to CIN7	Input	Comparator input
A/D converter (ADC)	AN7 to AN0	Input	Analog input
	$\overline{ADTRG}$	Input	A/D conversion external trigger input

Pin Functions (cont)

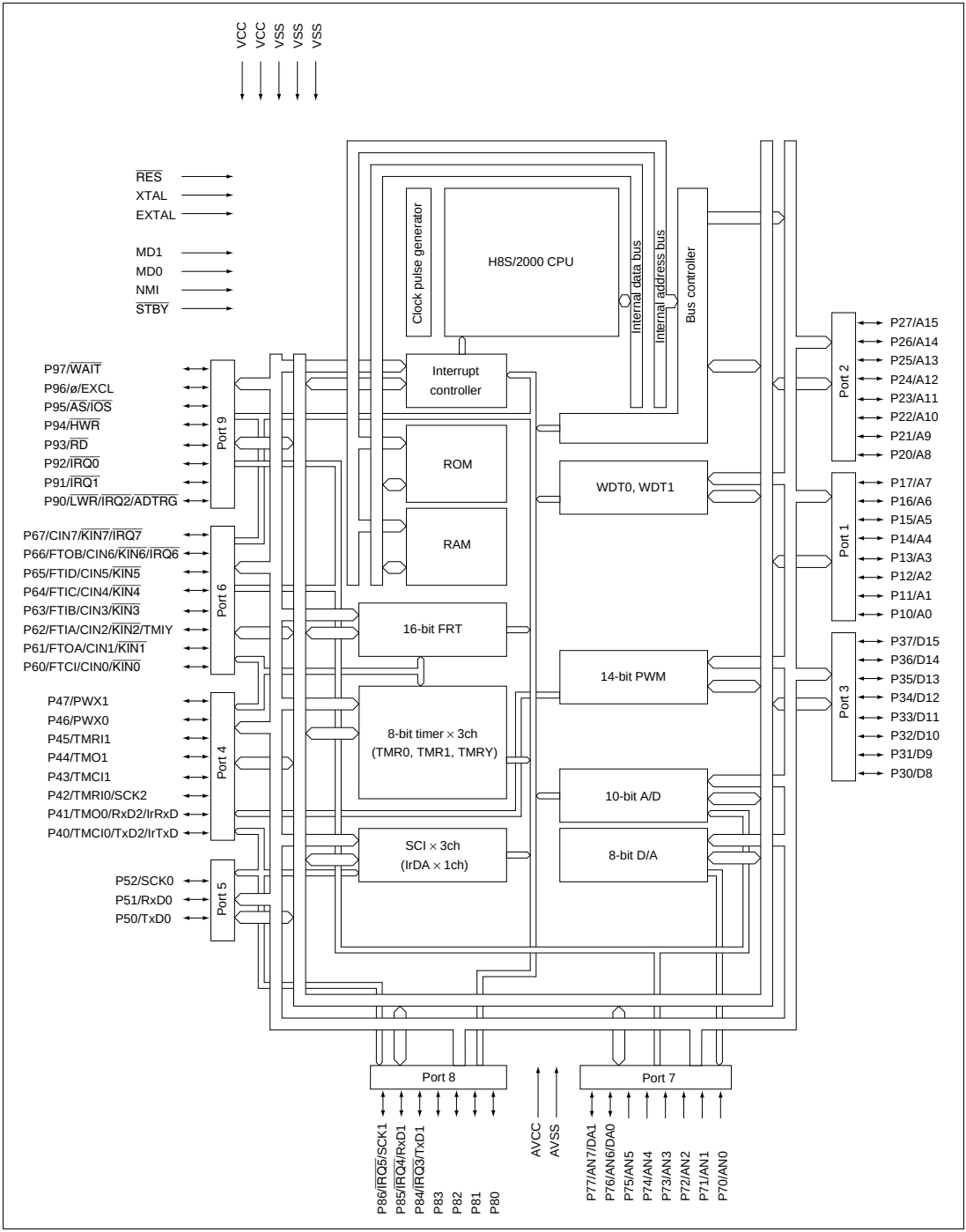
Type	Symbol	I/O	Name and Function
D/A converter (DAC)	DA0	Output	Analog output
	DA1		
A/D converter	AVCC	Input	Analog power supply
D/A converter	AVSS	Input	Analog ground
			Connect to the system power supply (0 V)
Timer connection	VSYNCI	Input	Timer connection input
	HSYNCI		
	CSYNCI		
	VFBACKI		
	HFBACKI		
	VSYNCO	Output	Timer connection output
	HSYNCO		
	CLAMPO		
	CBLANK		
I^2C bus interface (IIC) (option)	SCL0	Input/output	IIC clock input/output (channels 0, 1)
	SCL1		
	SDA0	Input/output	IIC data input/output (channels 0, 1)
	SDA1		
I/O ports	P17 to P10	Input/output	Port 1
	P27 to P20	Input/output	Port 2
	P37 to P30	Input/output	Port 3
	P47 to P40	Input/output	Port 4
	P52 to P50	Input/output	Port 5
	P67 to P60	Input/output	Port 6
	P77 to P70	Input	Port 7
	P86 to P80	Input/output	Port 8
	P97 to P90	Input/output	Port 9

2.3 Internal Block Diagram

Internal Block Diagram of H8S/2138 Series



Internal Block Diagram of H8S/2134 Series



Section 3 CPU

3.1 Overview

The H8S/2000 CPU is 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 H8S/2000 CPU has 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 and H8/300H CPUs
  - 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-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- 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: 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
- 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.

Differences from H8/300 CPU: In comparison to the H8/300 CPU, the H8S/2000 CPU has 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/2000 CPU has 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.

3.2 Register Configuration

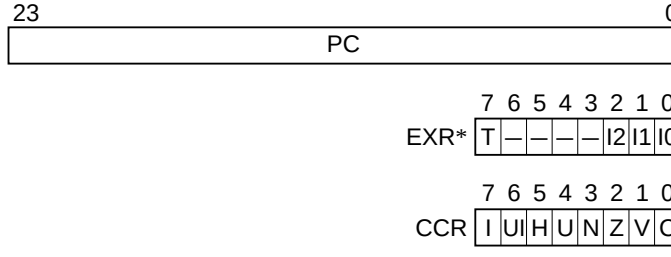
The H8S/2000 CPU has general registers and control registers. The eight 32-bit general registers all have identical functions and can be used as either address registers or data registers. The control registers are the 24-bit program counter (PC), 8-bit extended register (EXR), and 8-bit condition code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

CPU Registers

General Registers (Rn) and Extended Registers (En)

	15 — 07	07 — 0	
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)



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		
UI:	User bit or interrupt mask bit		

Note: * In these series, this bit does not affect operation.

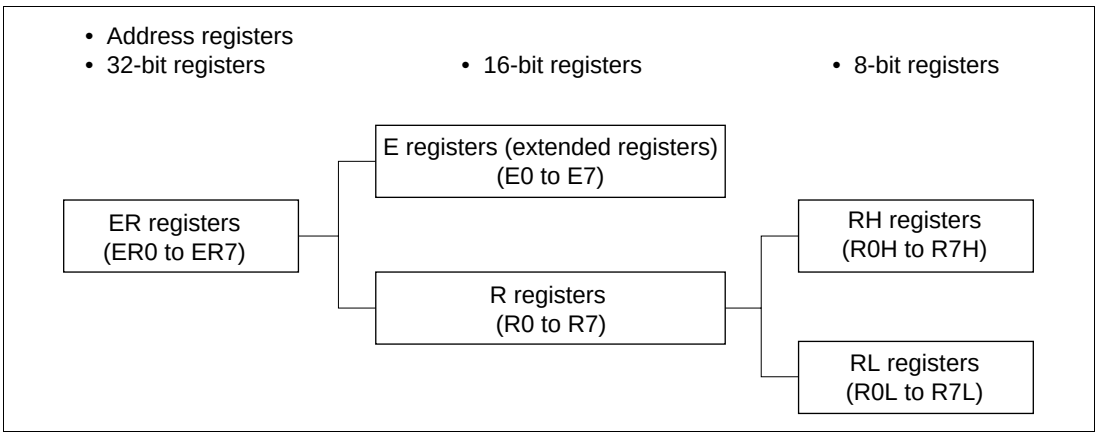
General Registers: The CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general 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



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).

- **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. In these series, this register does not affect operation.

  — Bit 7—Trace Bit (T): This bit is reserved. In these series, this bit does not affect operation.

  — Bits 6 to 3—Reserved: These bits are reserved. They are always read as 1.

  — Bits 2 to 0—Interrupt Mask Bits (I2 to I0): This bit is reserved. In these series, this bit does not affect operation.

- **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 3.9, 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 3.9, 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:

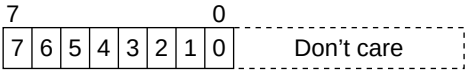
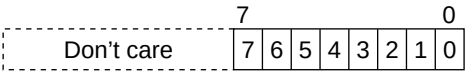
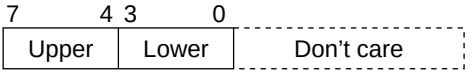
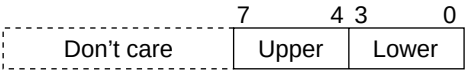
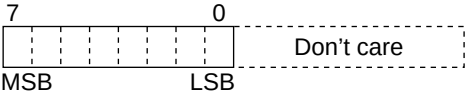
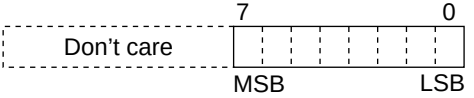
    - Add instructions, to indicate a carry
    - Subtract instructions, to indicate a borrow
    - Shift and rotate instructions, to store the carry

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

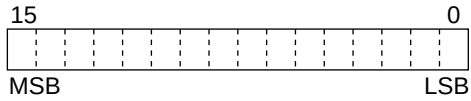
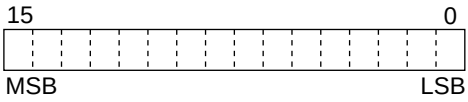
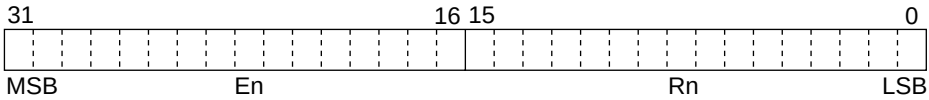
3.3 Data Formats

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

Data Type	Register Number	Data Format
1-bit data	RnH	
1-bit data	RnL	
4-bit BCD data	RnH	
4-bit BCD data	RnL	
Byte data	RnH	
Byte data	RnL	

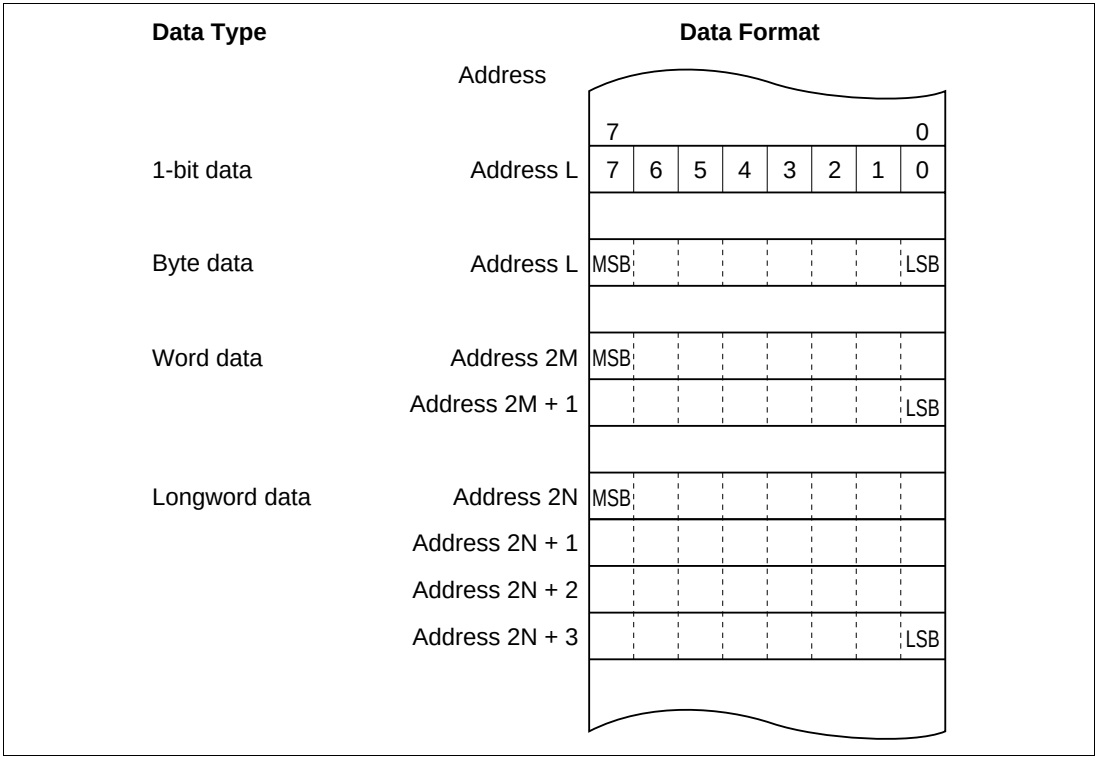
General Register Data Formats (cont)

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



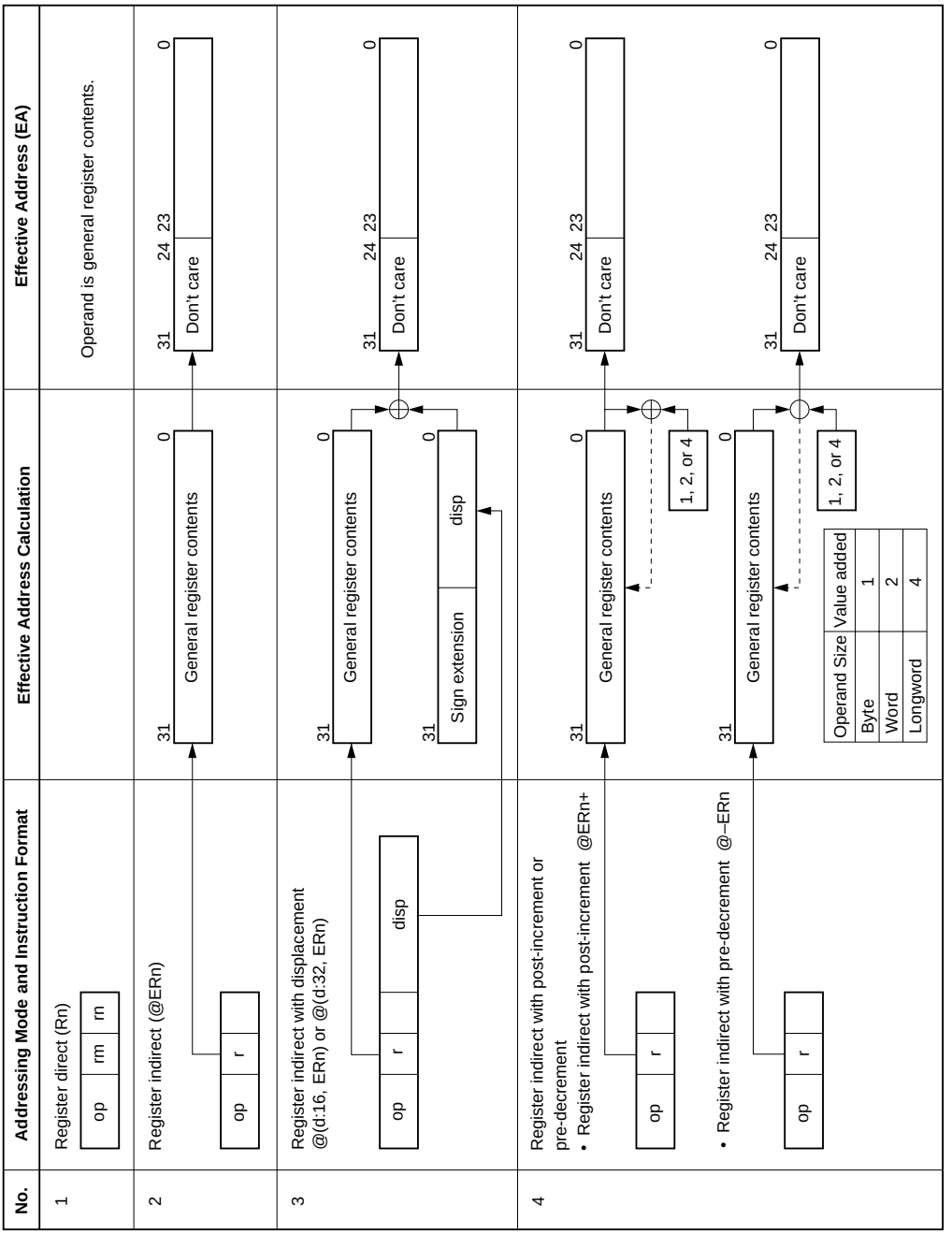
3.4 Addressing Modes

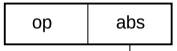
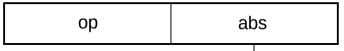
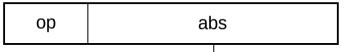
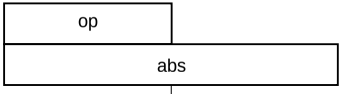
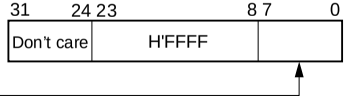
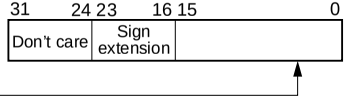
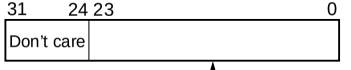
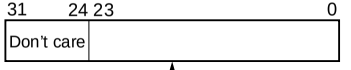
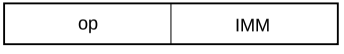
The H8S/2000 CPU supports eight addressing modes.

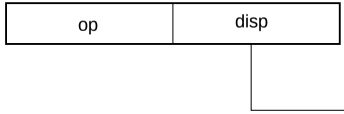
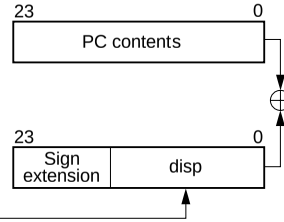
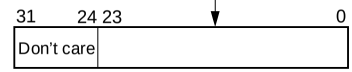
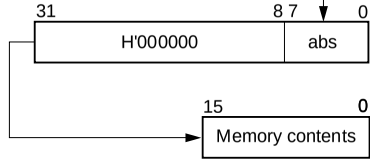
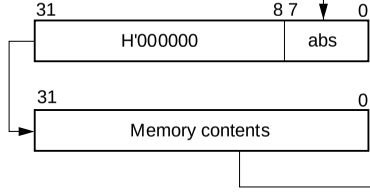
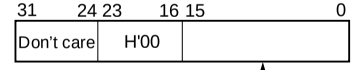
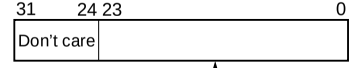
Addressing Modes

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: 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)
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 	 	 

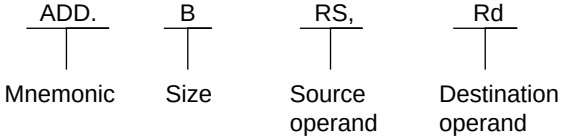
3.5 Instruction Set

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
- 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	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						Rd8→@(d:16,ERd)	—	—	↕	↕	0	—	3	
	MOV.B Rs,@(d:32,ERd)	B				8						Rd8→@(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]	

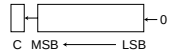
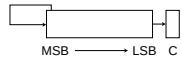
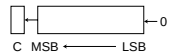
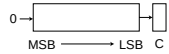
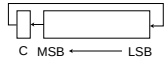
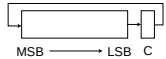
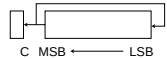
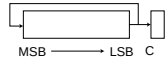
- Arithmetic instructions

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	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)	—	—	—	—	—	—		12
	MULXU.W Rs,ERd	W		2								Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—		20
MULXS	MULXS.B Rs,Rd	B		4								Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—		13
	MULXS.W Rs,ERd	W		4								Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—		21
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

- 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	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	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	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	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	Branch Condition	Condition Code						Number of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|			I	H	N	Z	V	C	Normal	Advanced
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then	Always	—	—	—	—	—	—	2	
	BRA d:16(BT d:16)	—							4			PC←PC+d		—	—	—	—	—	—	3	
	BRN d:8(BF d:8)	—							2			else next;	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	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	Advanced
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 R5+1→R5 R6+1→R6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n*2	
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 R5+1→R5 R6+1→R6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n*2	

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: 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] Set to 1 if the divisor is negative: otherwise cleared to 0.
[7] Set to 1 if the divisor is zero; otherwise cleared to 0.
[8] Set to 1 if the quotient is negative; otherwise cleared to 0.
[9] 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	4	2	4	6 + 2m	2	3 + m
Branch instruction read							
Stack operation							
Byte data access		2		2	3 + m		
Word data access		4		4	6 + 2m		
Internal operation	1	1	1	1	1	1	1

Legend
m: Number of wait states inserted in external device access

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)
(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).

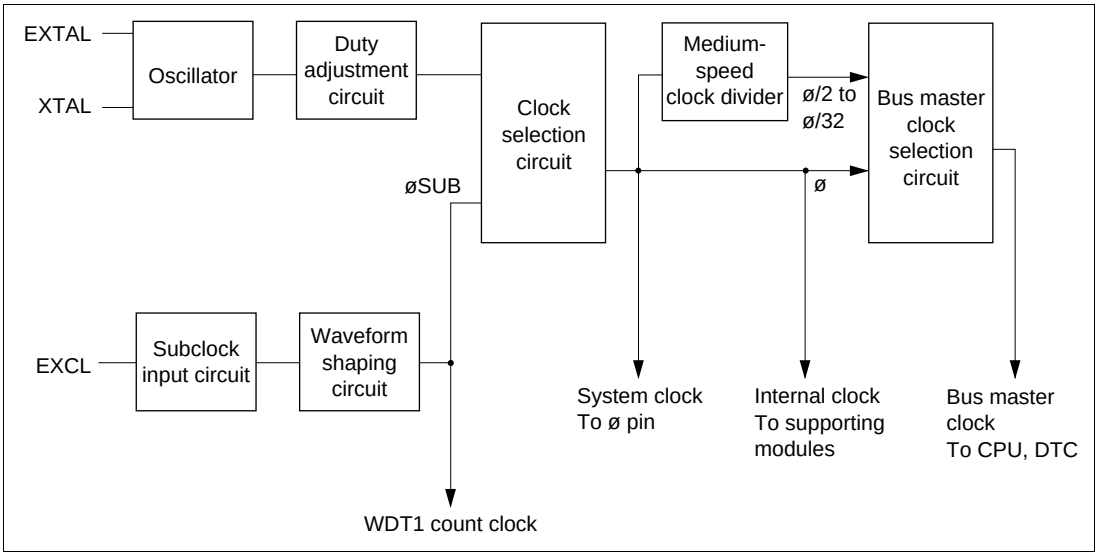
3.6 Basic Timing

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." The memory cycle or 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.

The following methods can be used to generate the system clock (ø):

1. Input an external clock of the same frequency as the system clock to the EXTAL pin
2. Connect a crystal oscillator of the same frequency as the system clock to the EXTAL and XTAL pins
3. Input an external clock of the same frequency as the subclock to the EXCL pin

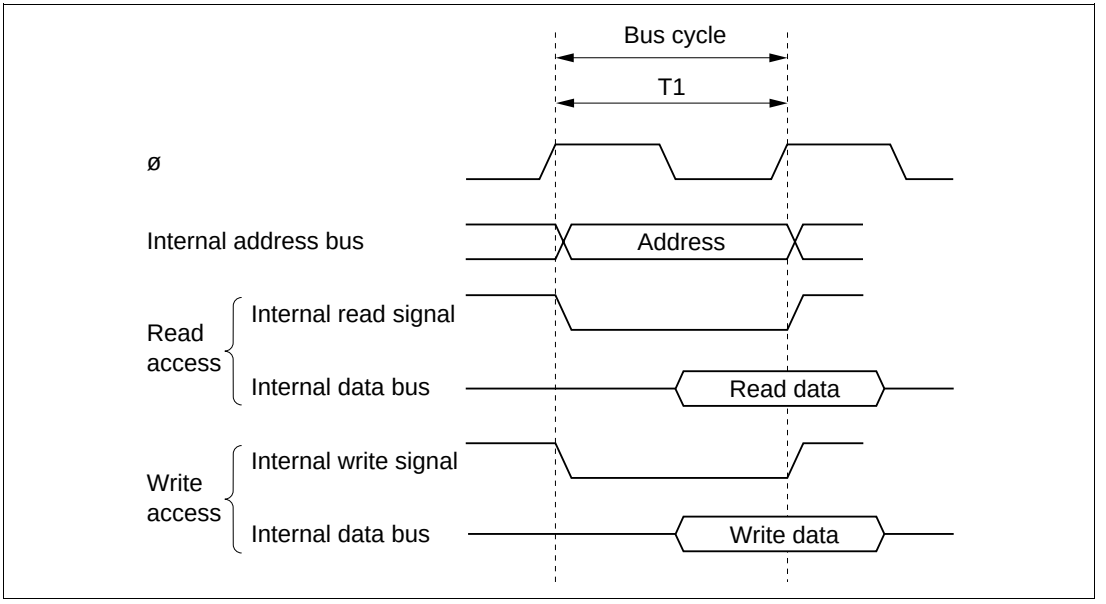


Basic Clock Timing

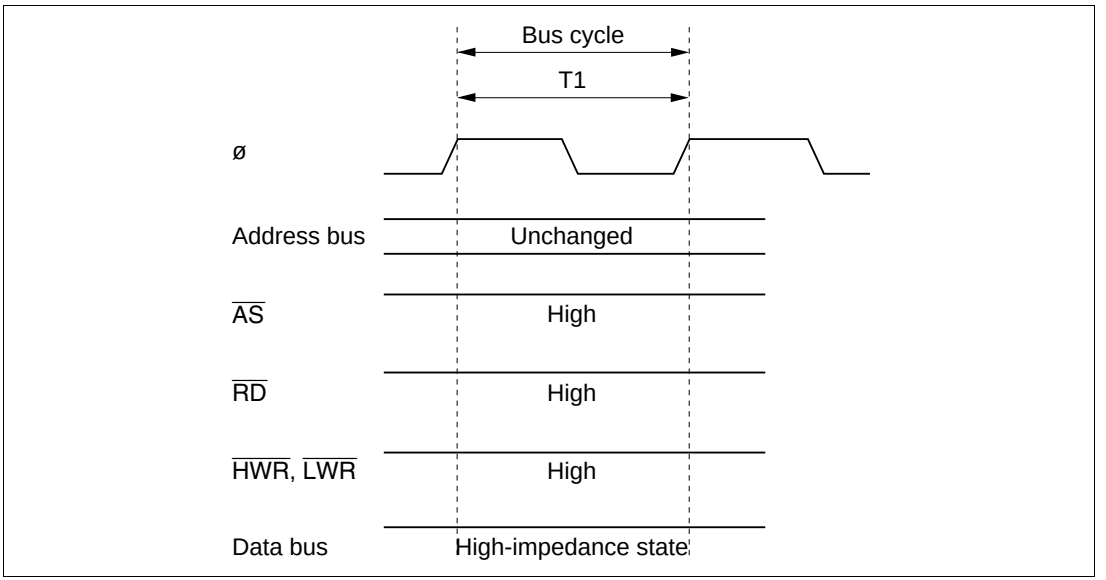
CPU Read/Write Cycles: The CPU operates on the basis of the ø clock. One ø clock cycle is called a state, and a bus cycle consists of one, two, or three states. Different access methods are used for on-chip memory, on-chip supporting modules, and external address space. Access to the external address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

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)

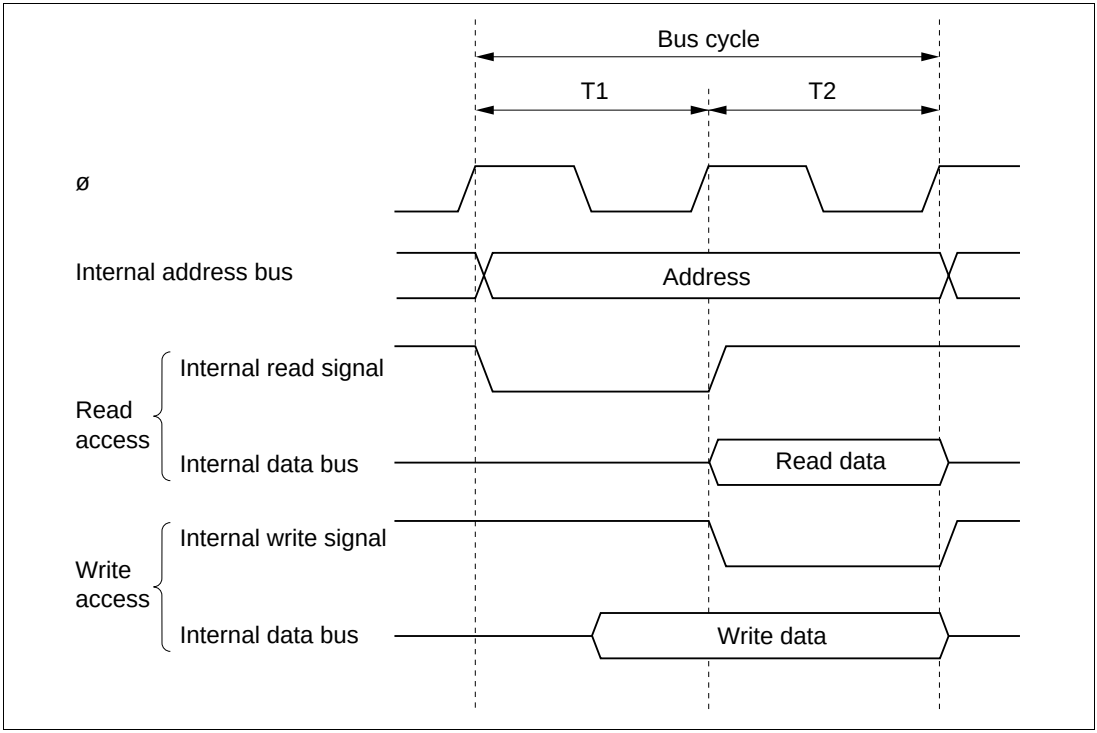


- Pin States during On-Chip Memory Access

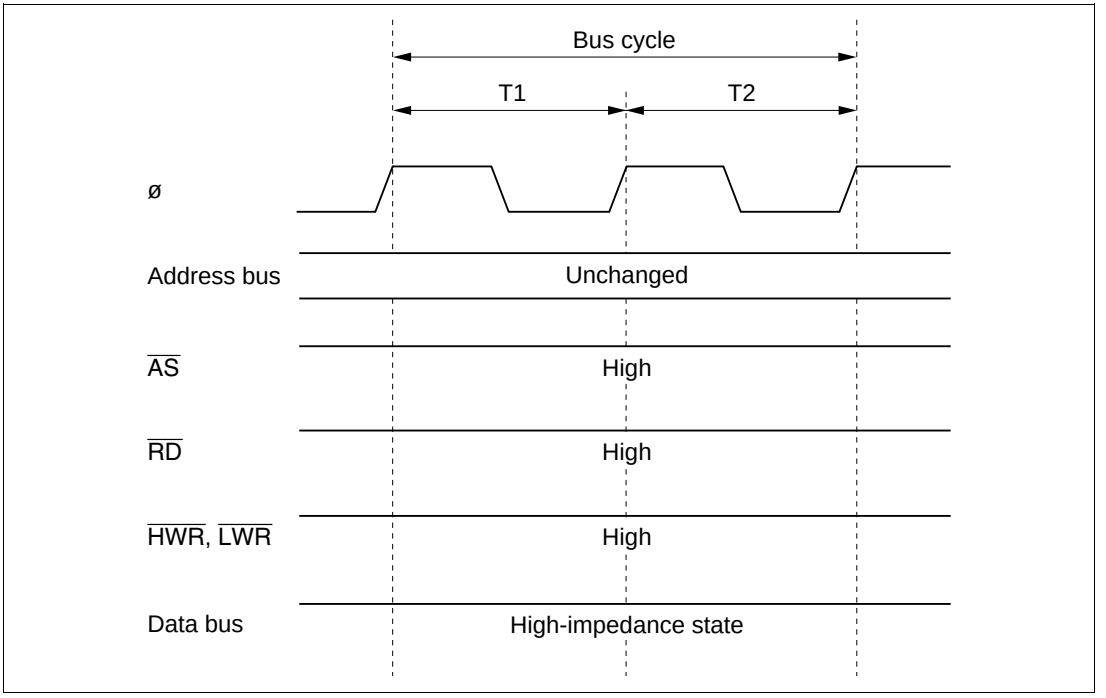


On-Chip Supporting Module Access Timing: The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed.

- 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 4.1, Bus Controller (BSC).

3.7 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

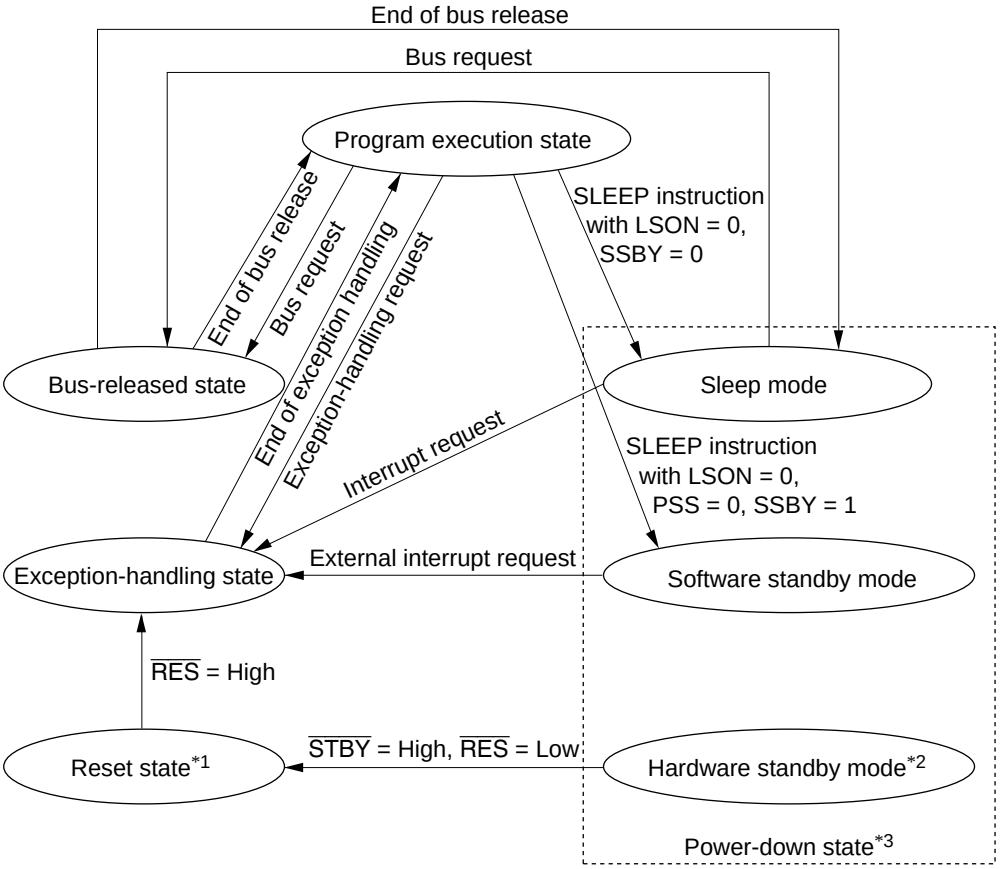
Program Execution State: State in which the CPU executes the program sequentially

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, subsleep mode, watch mode). The power-down state also includes medium-speed mode, module stop mode, and subactive mode.

State Transition Diagram



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition to the reset state can also be caused by watchdog timer overflow.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
3. There is also a watch mode, subactive mode, and subsleep mode. For details, see section 5, Power-Down State.

3.8 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 $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Trace*	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
	Direct transition	Started by a direct transition resulting from execution of a SLEEP instruction
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA)

Note: * Traces are enabled only in interrupt control modes 2 and 3. (They cannot be used in these series.) Trace exception handling is not executed after execution of an RTE instruction.

Exception Handling Operation: Exceptions originate from various sources. Trap instruction exception handling is always accepted in the program execution state. Trap instructions and interrupts are handled as follows:

1. The program counter (PC) and condition code register (CCR) 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	Vector Address*[1] 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 | 103	H'0030 to H'0031 | H'00CE to H'00CF	H'0060 to H'0063 | H'019C to H'019F

Notes: 1. Lower 16 bits of the address.

2. For details of internal interrupt vectors, see section 3.9, Interrupts.

3.9 Interrupts

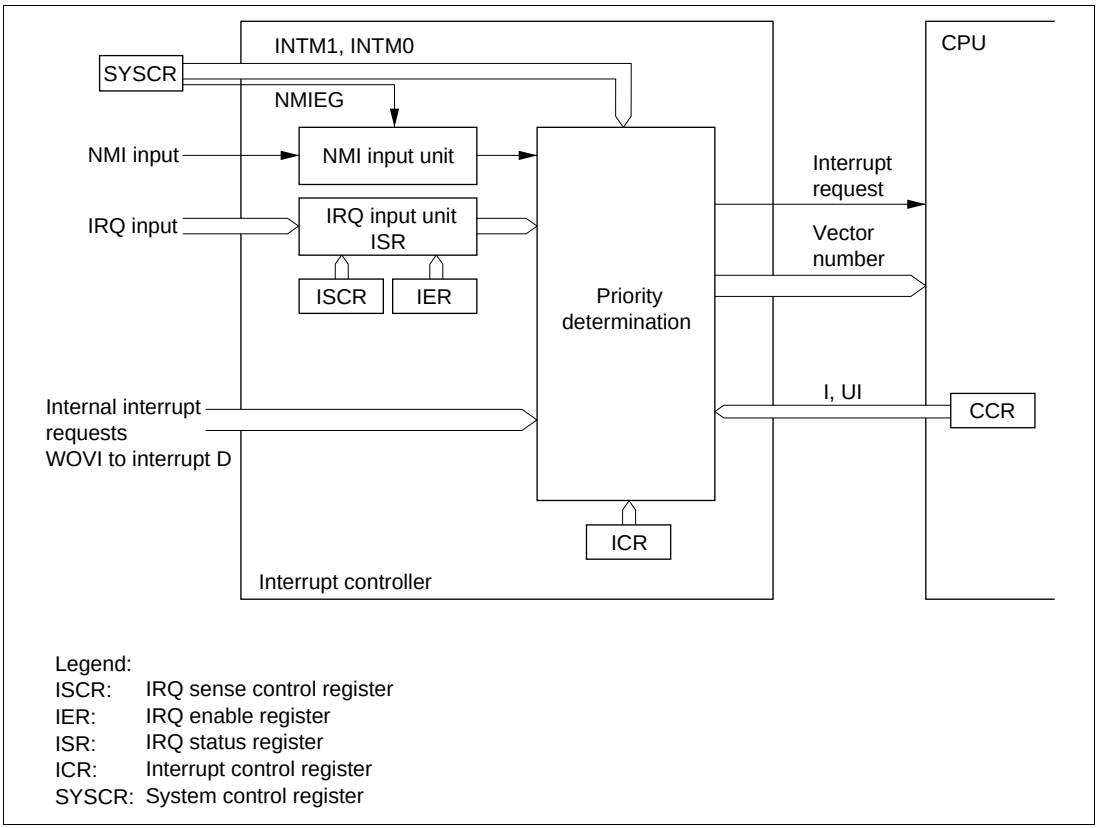
Interrupts are controlled by the interrupt controller. A separate vector number is assigned to each interrupt.

Interrupt Control: Either of two interrupt control modes 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 the masking conditions set by the I and UI bits in CCR.

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

Block Diagram of Interrupt Controller

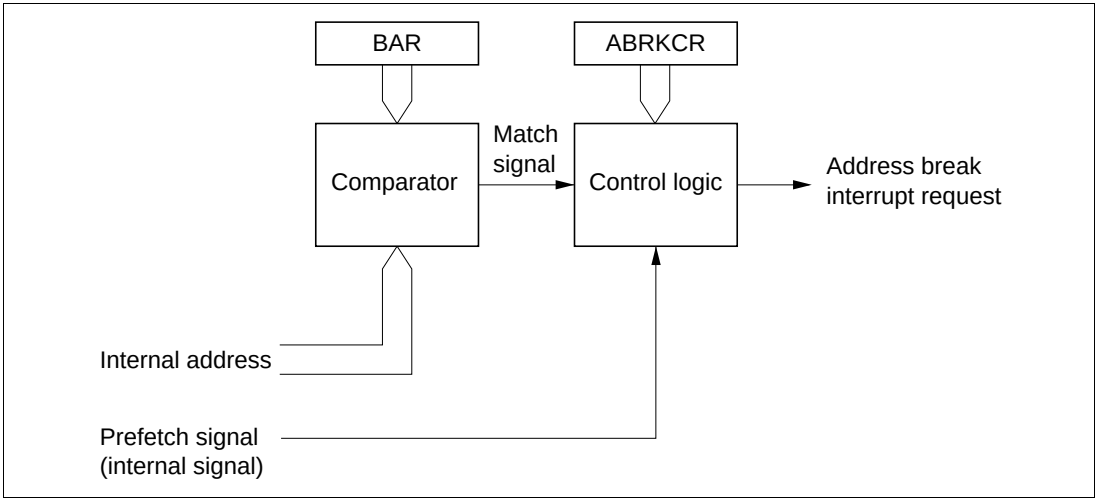


Legend:
ISCR: IRQ sense control register
IER: IRQ enable register
ISR: IRQ status register
ICR: Interrupt control register
SYSCR: System control register

Address Breaks: With these series, it is possible to identify the prefetch of a specific address by the CPU and generate an address break interrupt. When an address break interrupt is generated, address break interrupt exception handling is executed.

This function can be used to detect the beginning of execution of a bug location in the program, and branch to a correction routine.

- Block Diagram of Address Break Function

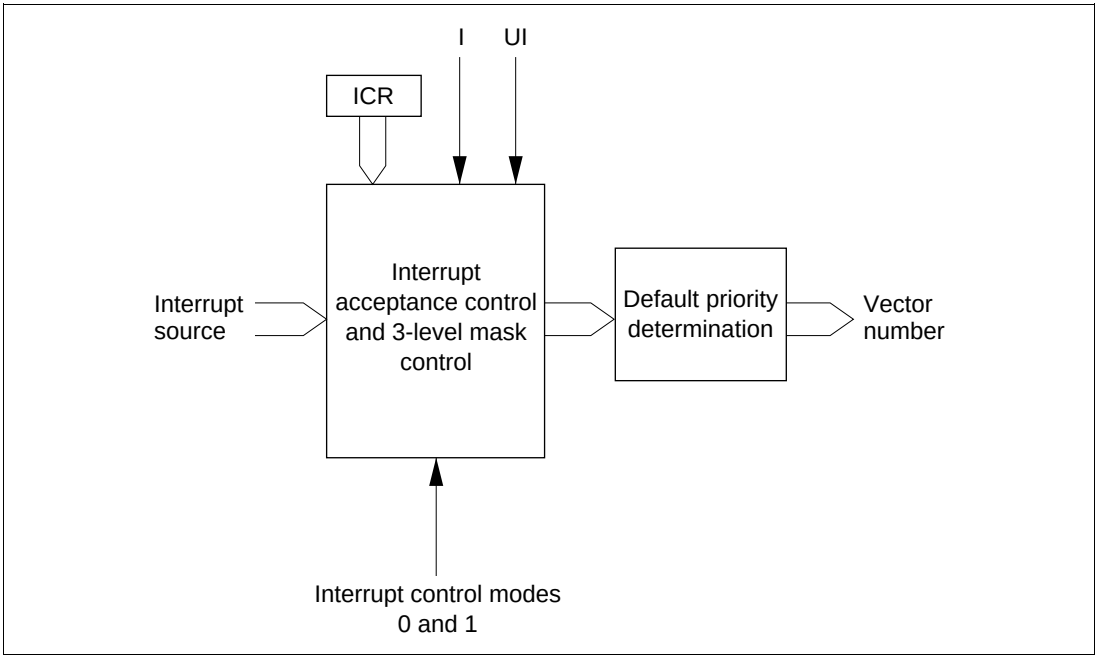


Interrupt Control Modes

Interrupt Control Mode	SYSCR INTM1*	SYSCR INTM0	Priority Setting Register	Interrupt Mask Bits	Description
0	0	0	ICR	I	Interrupt mask control is performed by the I bit Priority can be set with ICR
1		1	ICR	I, UI	3-level interrupt mask control is performed by the I and UI bits Priority can be set with ICR

Note: * INTM1 cannot be set to 1.

- Block Diagram of Interrupt Control Operation



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 Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
NMI	External pin	7	H'000E	H'00001C		High
IRQ0		16	H'0020	H'000040	ICRA7	
IRQ1		17	H'0022	H'000044	ICRA6	
IRQ2 IRQ3		18 19	H'0024 H'0026	H'000048 H'00004C	ICRA5	
IRQ4 IRQ5		20 21	H'0028 H'002A	H'000050 H'000054	ICRA4	
IRQ6, KIN7 to KIN0 IRQ7, KIN15 to KIN8		22 23	H'002C H'002E	H'000058 H'00005C	ICRA3	
SWDTEND (software activation interrupt end)	DTC	24	H'0030	H'000060	ICRA2	
WOVI0 (interval timer)	Watchdog timer 0	25	H'0032	H'000064	ICRA1	
WOVI1 (interval timer)	Watchdog timer 1	26	H'0034	H'000068	ICRA0	
PC break	—	27	H'0036	H'00006C		
ADI (A/D conversion end)	A/D	28	H'0038	H'000070	ICRB7	
Reserved	—	29 to 47	H'003A to H'005E	H'000074 to H'0000BC		
ICIA (input capture A) ICIB (input capture B) ICIC (input capture C) ICID (input capture D) OCIA (output compare A) OCIB (output compare B) FOVI (overflow) Reserved	Free-running timer	48 49 50 51 52 53 54 55	H'0060 H'0062 H'0064 H'0066 H'0068 H'006A H'006C H'006E	H'0000C0 H'0000C4 H'0000C8 H'0000CC H'0000D0 H'0000D4 H'0000D8 H'0000DC	ICRB6	
Reserved	—	56 to 63	H'0070 to H'007E	H'0000E0 to H'0000FC		Low

Interrupt Sources, Vector Addresses, and Interrupt Priorities (cont)

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
CMIA0 (compare-match A)	8-bit timer	64	H'0080	H'000100	ICRB3	High
CMIB0 (compare-match B)	channel 0	65	H'0082	H'000104		
OVI0 (overflow)		66	H'0084	H'000108		
Reserved		67	H'0086	H'00010C		
CMIA1 (compare-match A)	8-bit timer	68	H'0088	H'000110	ICRB2	
CMIB1 (compare-match B)	channel 1	69	H'008A	H'000114		
OVI1 (overflow)		70	H'008C	H'000118		
Reserved		71	H'008E	H'00011C		
CMIAY (compare-match A)	8-bit timer	72	H'0090	H'000120	ICRB1	
CMIBY (compare-match B)	channels	73	H'0092	H'000124		
OVIY (overflow)	Y, X	74	H'0094	H'000128		
ICIX (input capture X)		75	H'0096	H'00012C		
IBF1 (IDR1 reception completed)	Host interface	76	H'0098	H'000130	ICRB0	
IBF2 (IDR2 reception completed)		77	H'009A	H'000134		
Reserved		78	H'009C	H'000138		
Reserved		79	H'009E	H'00013C		
ERI0 (receive error 0)	SCI	80	H'00A0	H'000140	ICRC7	
RXI0 (reception completed 0)	channel 0	81	H'00A2	H'000144		
TXI0 (transmit data empty 0)		82	H'00A4	H'000148		
TEI0 (transmission end 0)		83	H'00A6	H'00014C		
ERI1 (receive error 1)	SCI	84	H'00A8	H'000150	ICRC6	
RXI1 (reception completed 1)	channel 1	85	H'00AA	H'000154		
TXI1 (transmit data empty 1)		86	H'00AC	H'000158		
TEI1 (transmission end 1)		87	H'00AE	H'00015C		
ERI2 (receive error 2)	SCI	88	H'00B0	H'000160	ICRC5	
RXI2 (reception completed 2)	channel 2	89	H'00B2	H'000164		
TXI2 (transmit data empty 2)		90	H'00B4	H'000168		
TEI2 (transmission end 2)		91	H'00B6	H'00016C		
IICI0 (1-byte transmission/ reception completed) DDCSWI (format switch)	IIC channel 0 (option)	92 93	H'00B8 H'00BA	H'000170 H'000174	ICRC4	
IICI1 (1-byte transmission/ reception completed)	IIC channel 1 (option)	94 95	H'00BC H'00BE	H'000178 H'00017C	ICRC3	
Reserved						Low

Interrupt Sources, Vector Addresses, and Interrupt Priorities (cont)

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
PS2IA (reception completed A)	Keyboard	96	H'00C0	H'000180	ICRB0	High
PS2IB (reception completed B)	buffer	97	H'00C2	H'000184		
PS2IC (reception completed C)	controller	98	H'00C4	H'000188		
Reserved		99	H'00C6	H'00018C		
Reserved	—	100 to 103	H'00C8 to H'00CE	H'000190 to H'00019C		Low

3.10 MCU Operating Modes

These series have three operating modes, determined by the setting of the mode pins (MD1 and MD0).

Mode 1: The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is disabled.

Ports 1 and 2 function as an address bus, ports 3 and B* function as a data bus, and part of port 9 carries bus control signals.

Note: * Port B is not provided in the H8S/2138 Series or H8S/2134 Series.

Mode 2: The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. After a reset, single-chip mode is set, and the EXPE bit must be set to 1 in MDCR in order to use external addresses. (In the H8S/2138 Series or H8S/2134 Series there are a maximum of 16 address outputs, and therefore external addresses can be specified correctly only when the I/O strobe function of the AS/IOS pin is used.)

When the EXPE bit is set to 1 in MDCR, ports 1, 2, and A* function as input ports after a reset. They can be designated as address outputs by setting the corresponding data direction register (DDR) bits to 1. Port 3 and B* function as a data bus, and part of port 9 carries bus control signals.

Note: * Ports A and B are not provided in the H8S/2138 Series or H8S/2134 Series.

Mode 3: The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is enabled. After a reset, single-chip mode is set, and the EXPE bit must be set to 1 in MDCR in order to use external addresses.

When the EXPE bit is set to 1 in MDCR, ports 1 and 2 function as input ports after a reset. They can be designated as address outputs by setting the corresponding data direction register (DDR) bits to 1. Port 3 and B* function as a data bus, and part of port 9 carries bus control signals.

The amount of on-chip ROM that can be used is limited to 56 kbytes.

Note: * Port B is not provided in the H8S/2138 Series or H8S/2134 Series.

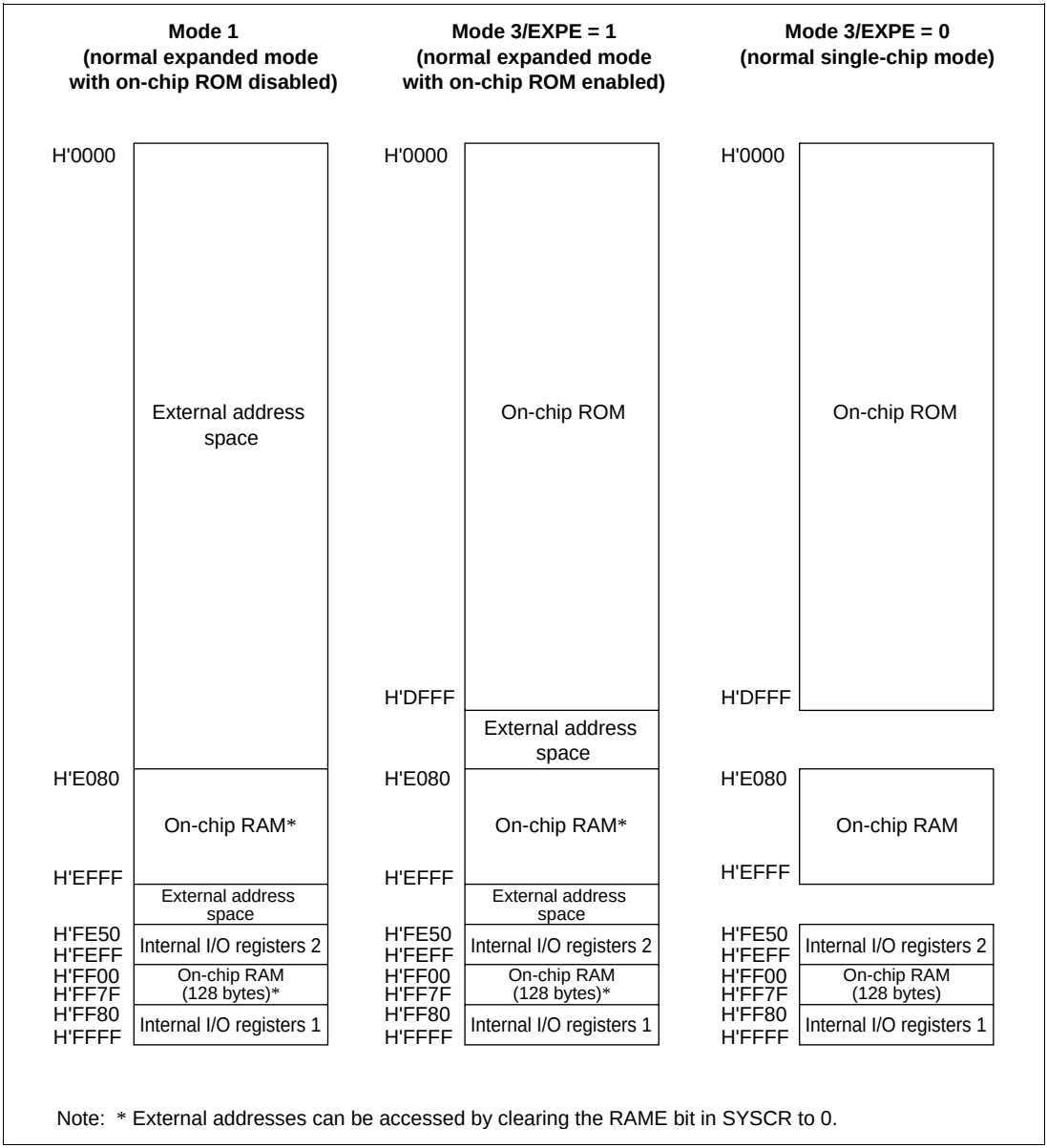
MCU Operating Mode Selection

MCU Operating Mode	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM
0	0	0	—	—	—
1		1	Normal	On-chip ROM disabled, expanded mode	Disabled
2	1	0	Advanced	On-chip ROM enabled, expanded mode Single-chip mode	Enabled
3		1	Normal	On-chip ROM enabled, expanded mode Single-chip mode	

3.11 Address Maps

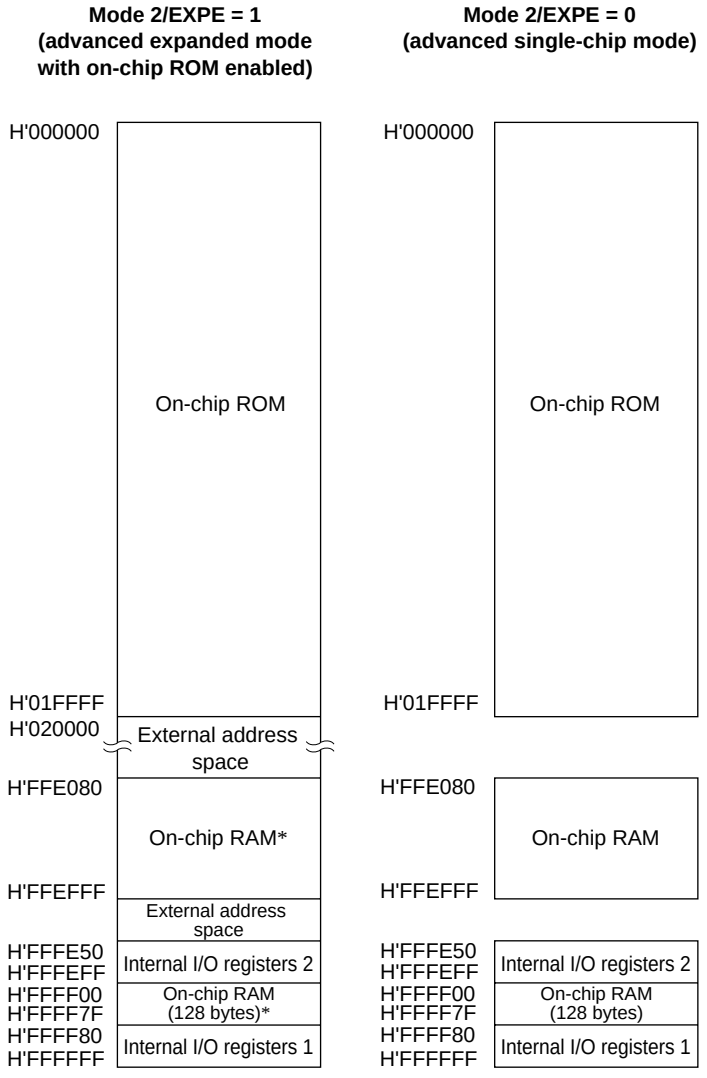
This section shows the address maps in each operating mode.

The address space is 64 kbytes in modes 1 and 3 (normal modes) and 16 Mbytes in mode 2 (advanced mode).



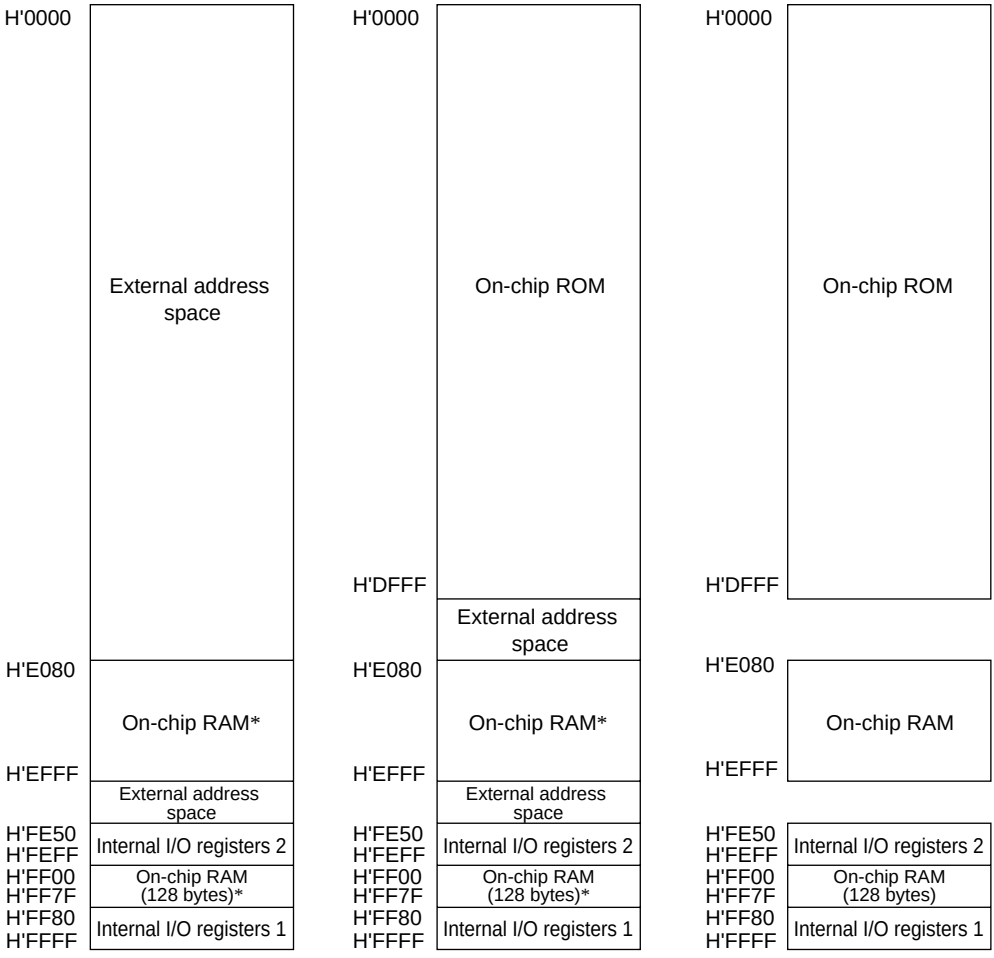
Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2148, H8S/2144, H8S/2138 and H8S/2134 Address Map in Each Operating Mode



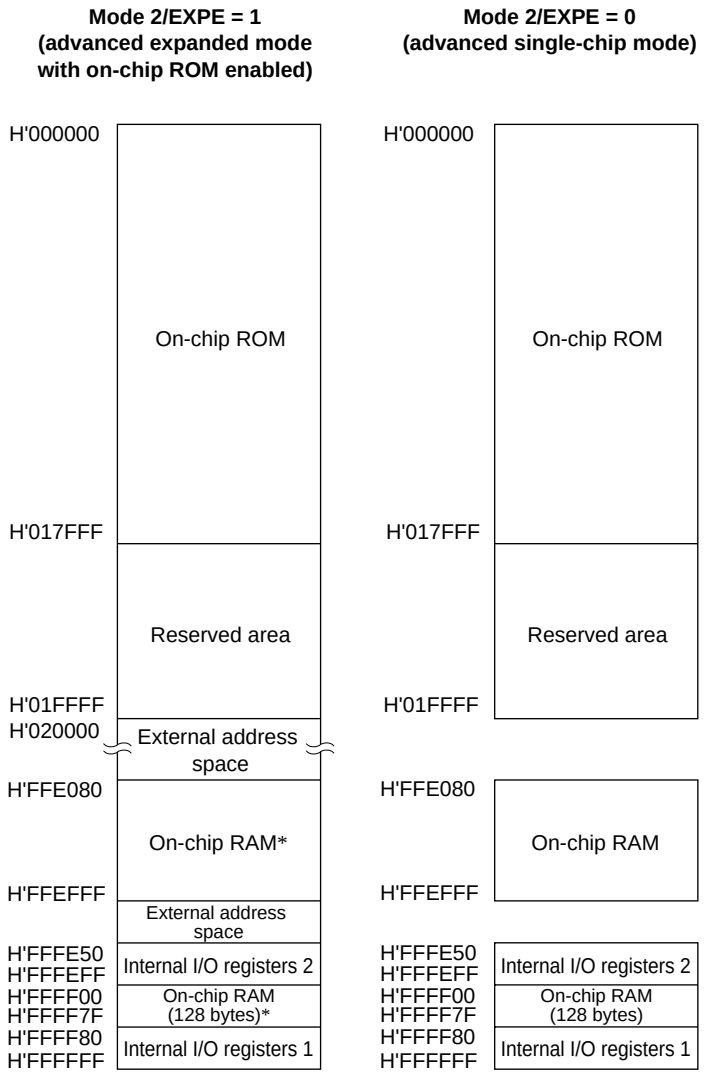
Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2148, H8S/2144, H8S/2138 and H8S/2134 Address Map in Each Operating Mode (cont)



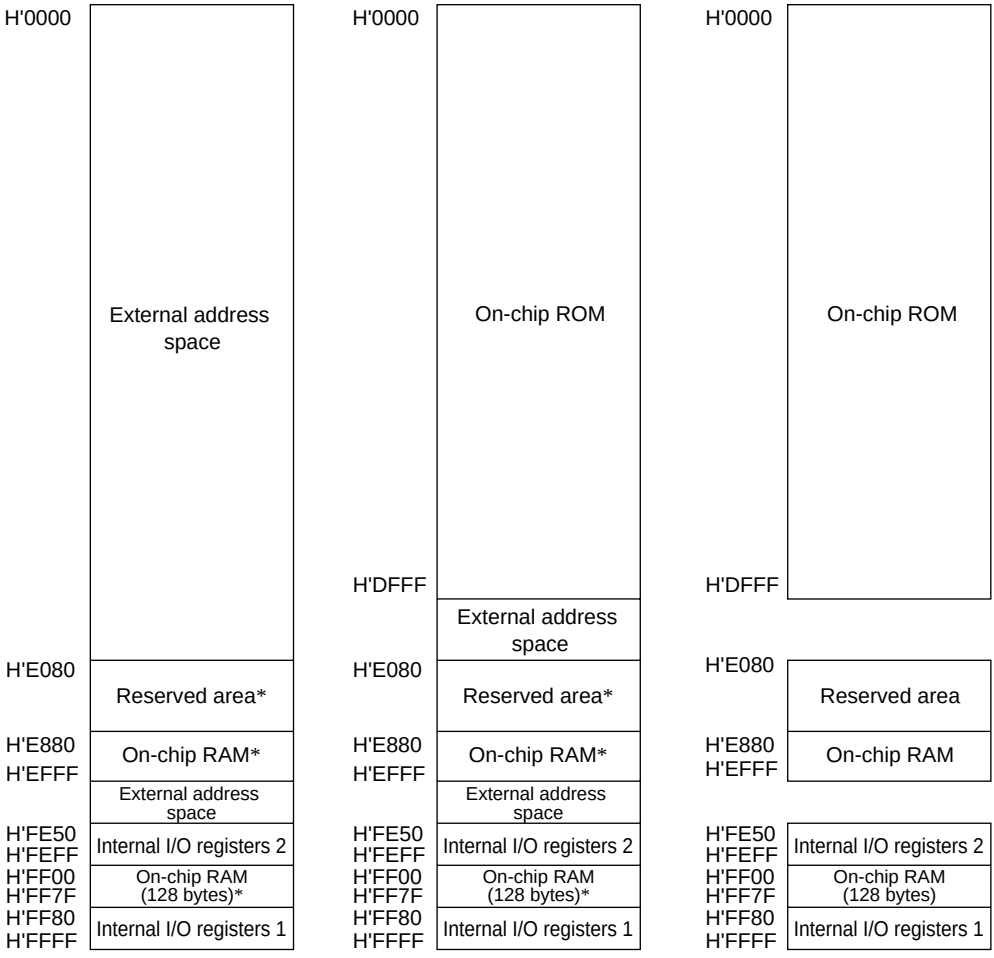
Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2143 and H8S/2133 Address Map in Each Operating Mode



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2143 and H8S/2133 Address Map in Each Operating Mode (cont)

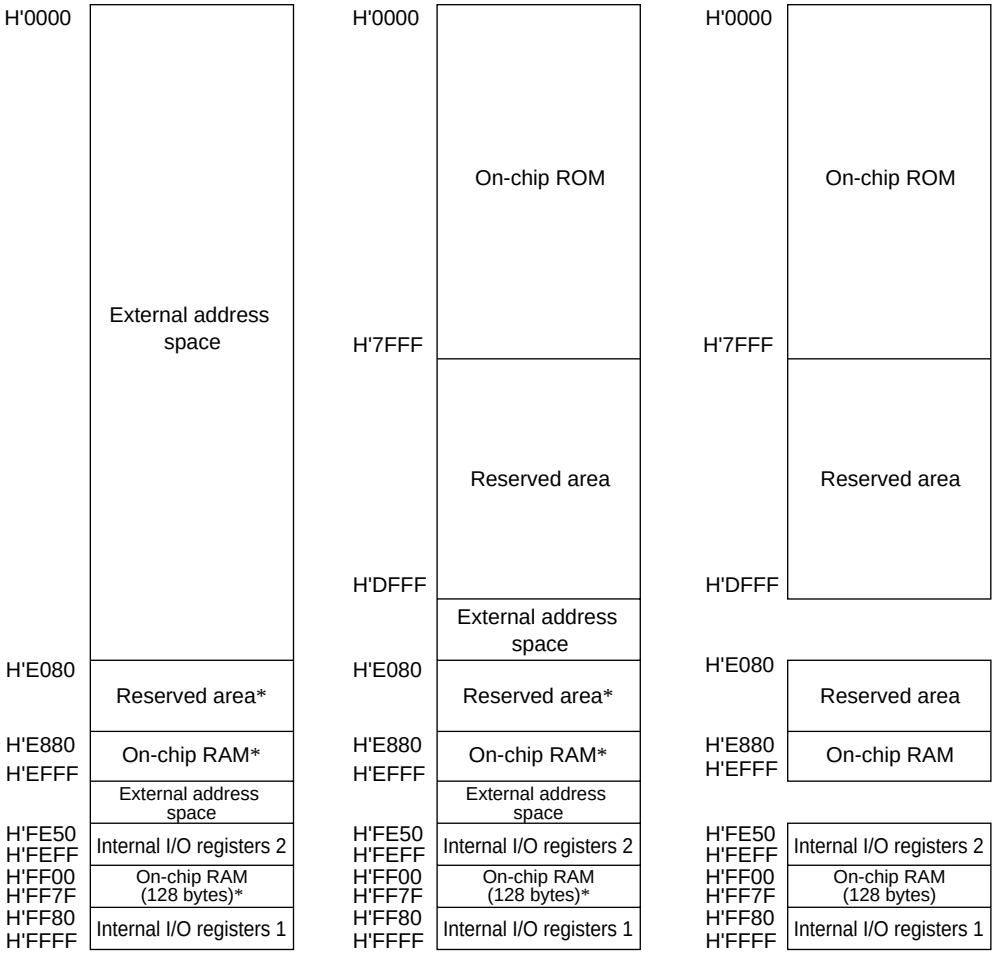


Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2147, H8S/2142, H8S/2137 and H8S/2132 Address Map in Each Operating Mode

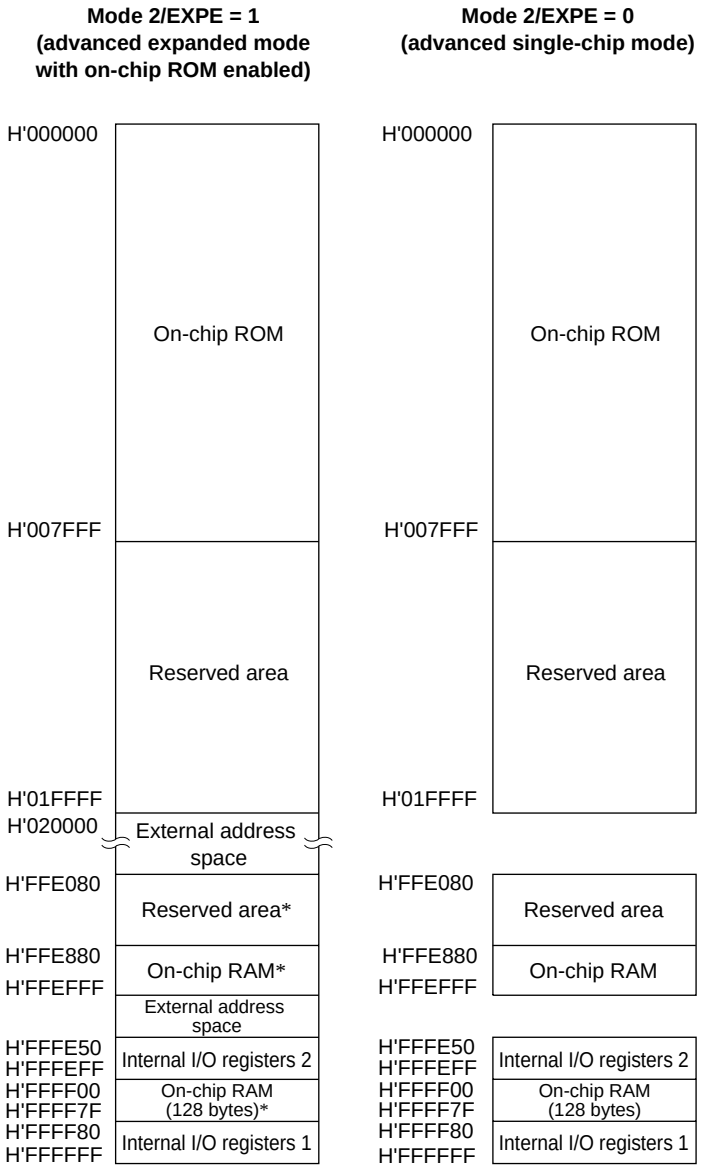
Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2147, H8S/2142, H8S/2137 and H8S/2132 Address Map in Each Operating Mode (cont)



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2130 Address Map in Each Operating Mode



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

H8S/2130 Address Map in Each Operating Mode (cont)

Section 4 Supporting Modules

4.1 Bus Controller (BSC)

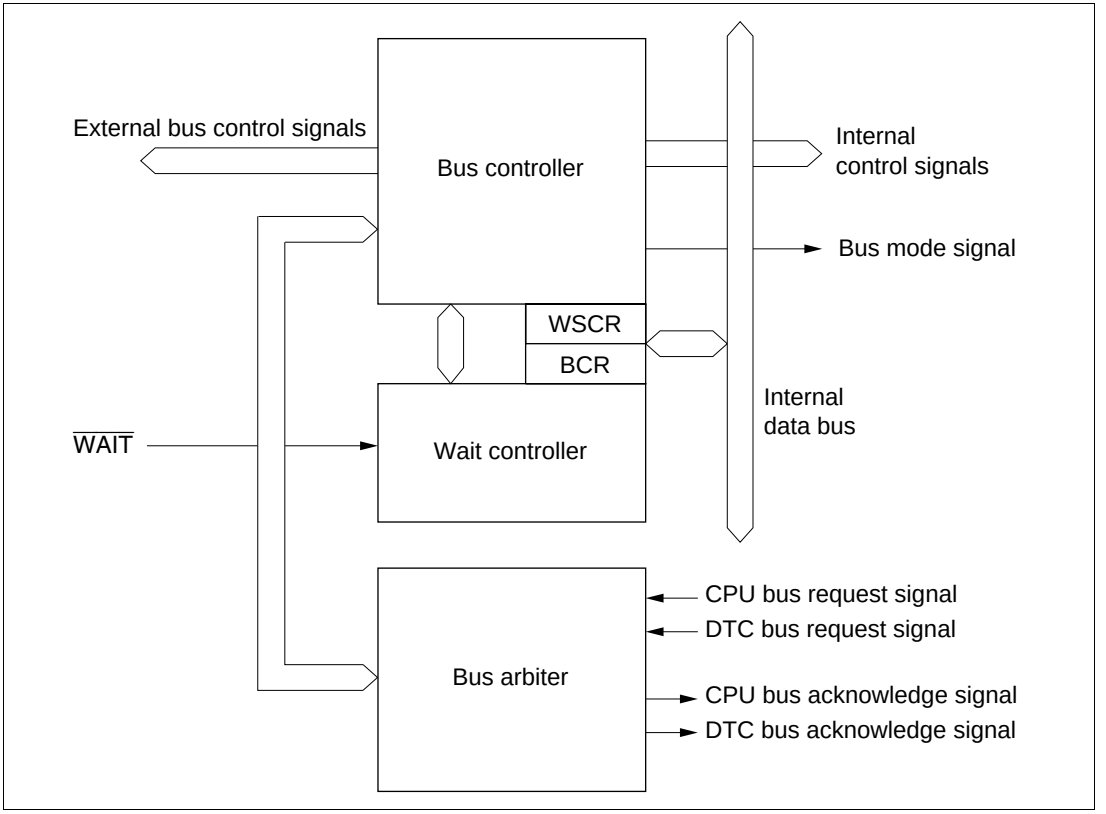
These series have a built-in bus controller (BSC) that allows external address space bus specifications, such as bus width and number of access states, to be set independently for different areas.

The bus controller also has a bus arbitration function, and controls the operation of the internal bus masters: the CPU and data transfer controller (DTC).

Features

- Basic bus interface
  - 2-state access or 3-state access can be selected
  - Program wait states can be inserted
- Burst ROM interface
  - External space can be designated as ROM interface space
  - 1-state or 2-state burst access can be selected
- Idle cycle insertion
  - An idle cycle can be inserted in an external write cycle immediately after an external read cycle
- Bus release
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership between the CPU and DTC

Block Diagram



Block Diagram of Bus Controller

Bus Specifications: The external space bus specifications consist of three elements: bus width, number of access states, and wait mode 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.

Bus Specifications for Each Area (Basic Bus Interface)

						Bus Specifications (Basic Bus Interface)		
ABW	**AST**	**WMS1**	**WMS0**	**WC1**	**WC0**	**Bus Width**	**Access States**	**Program Wait States**
0*	0	—	—	—	—	16	2	0
	1	0	1	—	—	16	3	0
		0	0	0	0		3	0
					1			1
		1	—	1	0			2
					1			3
1	0	—	—	—	—	8	2	0
	1	0	1	—	—	8	3	0
		0	0	0	0		3	0
					1			1
		1	—	1	0			2
					1			3

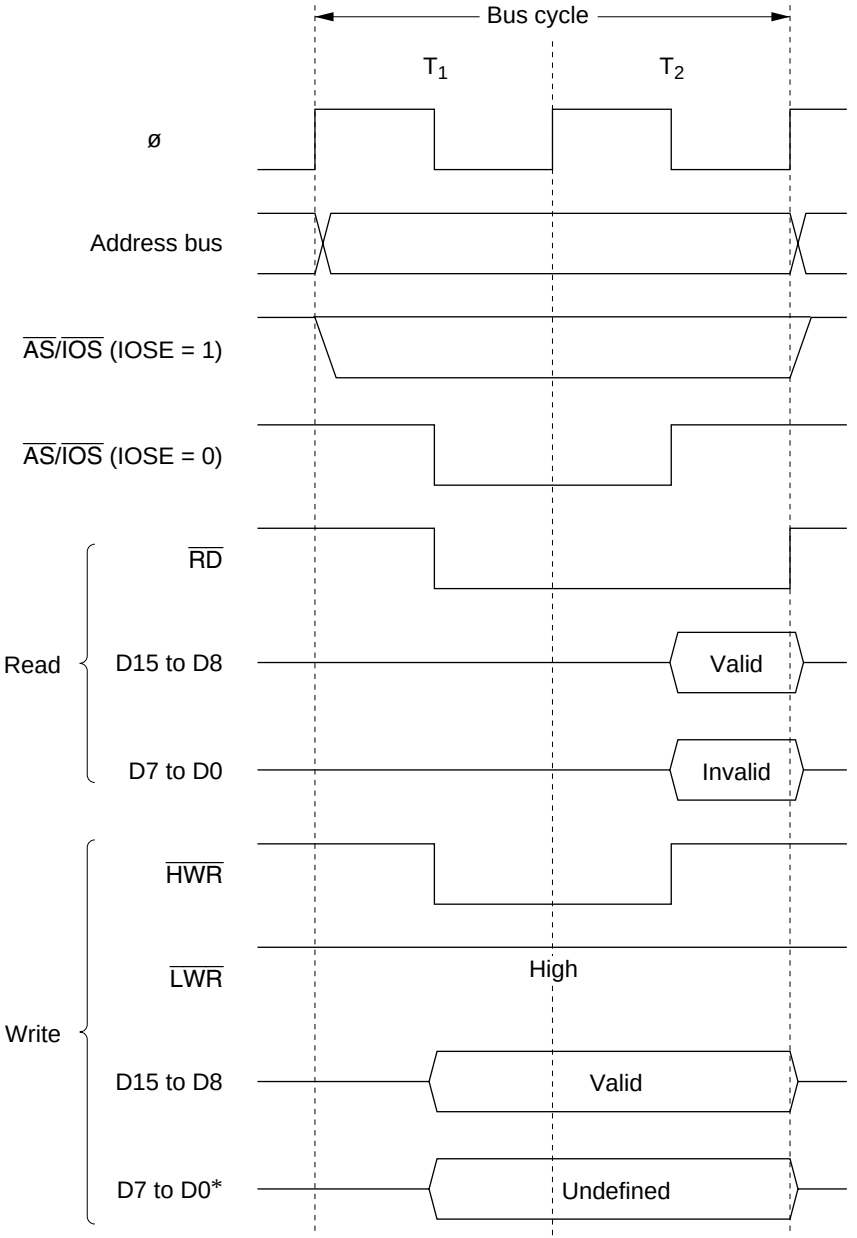
Note: * Cannot be set in the H8S/2138 Series or H8S/2134 Series.

Memory Interfaces: These series' memory interfaces comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on, and a burst ROM interface that allows direct connection of burst ROM.

Basic Bus Interface: 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.

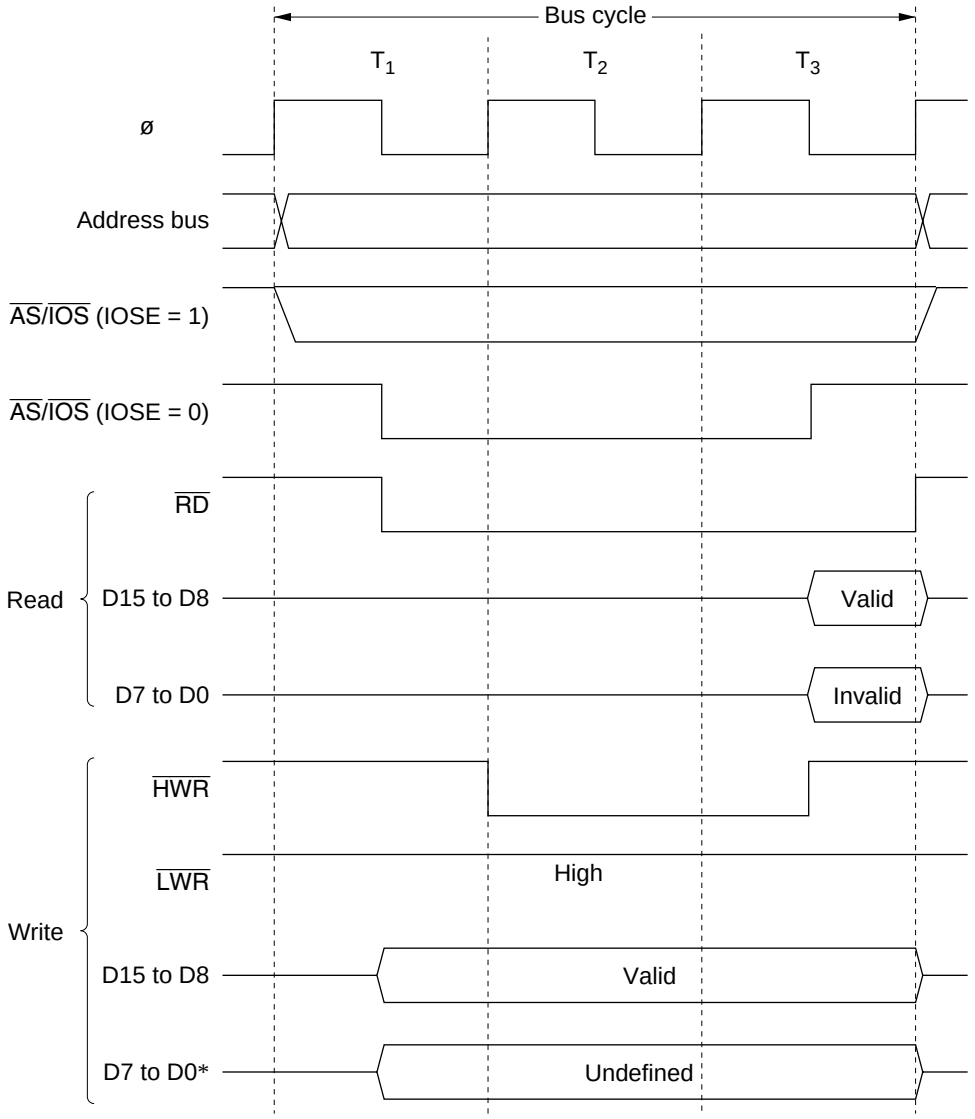
After a reset, all areas are designated as basic bus interface, 8-bit, 3-state access space.

- Bus Timing for Word Access to 8-Bit 2-State Access Space



Note: * The H8S/2138 Series and H8S/2134 Series do not have the lower data bus signals (D7 to D0).

- Bus Timing for Word Access to 8-Bit 3-State Access Space

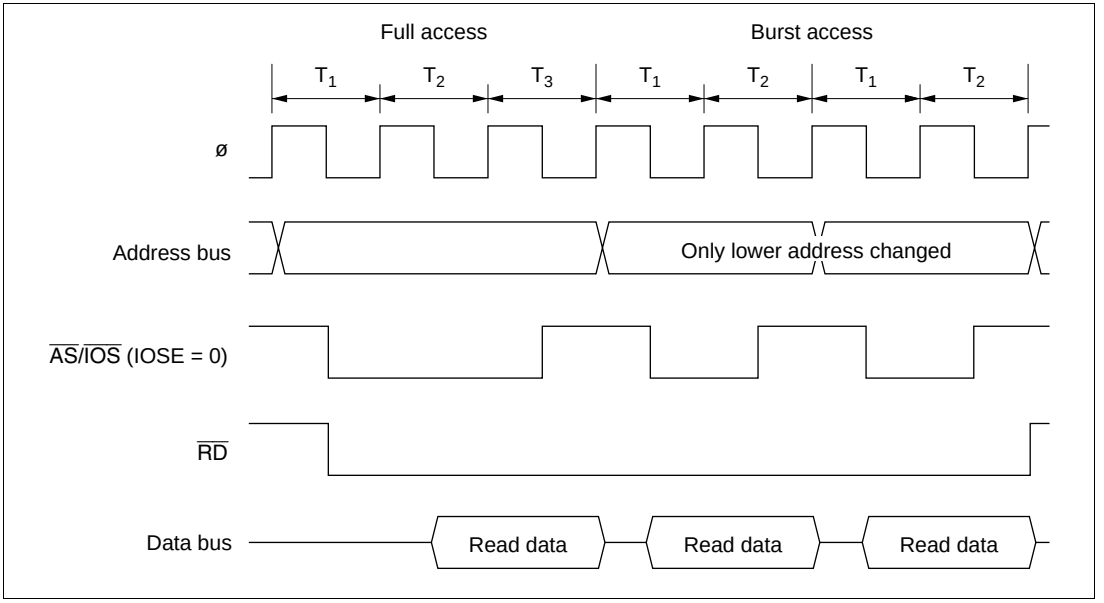


Note: * The H8S/2138 Series and H8S/2134 Series do not have the lower data bus signals (D7 to D0).

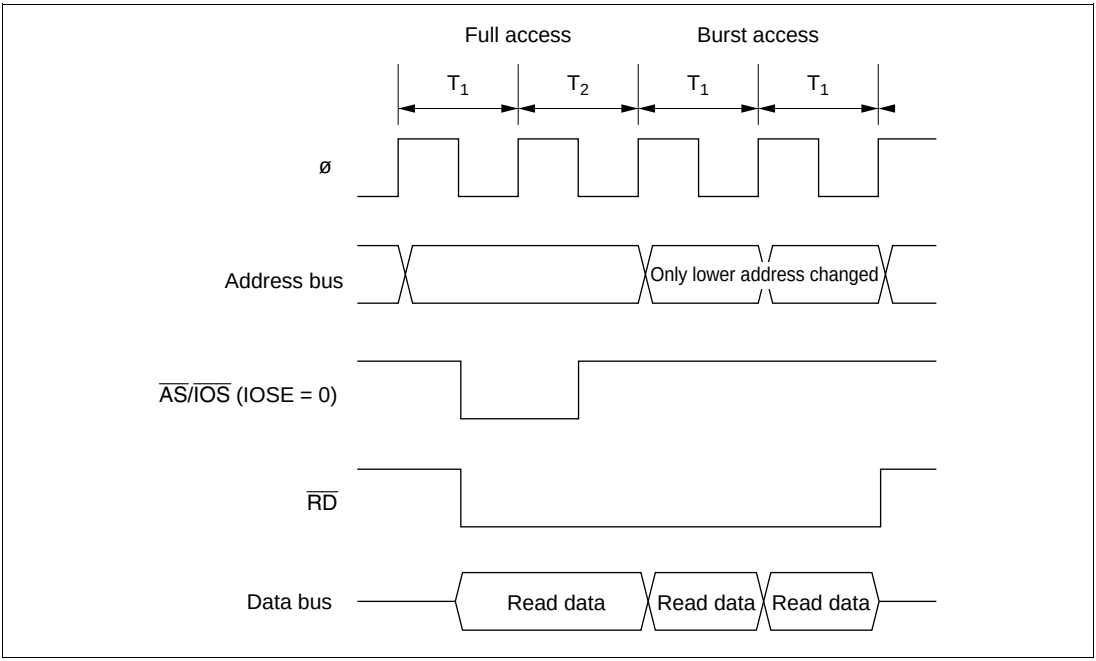
Burst ROM Interface: External space area 0 can be designated as burst ROM space, and burst ROM space interfacing can be performed. The burst ROM space interface enables 16-bit configuration ROM with burst access capability to be accessed at high speed.

Consecutive burst accesses of a maximum or 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 AST = BRSTS1 = 1)



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



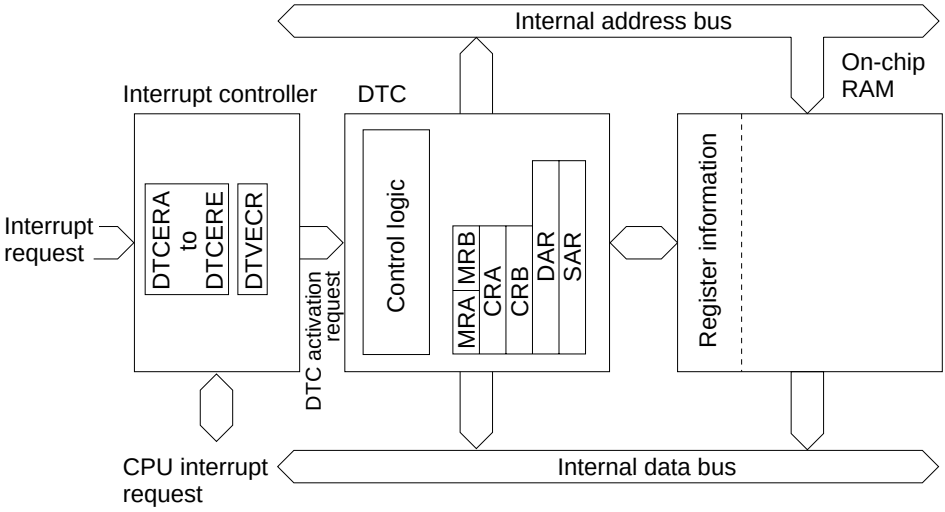
4.2 Data Transfer Controller (DTC) (Provided in the H8S/2148 Series and H8S/2138 Series)

The data transfer controller (DTC) is activated by an interrupt or software, and can transfer data without imposing any load on the CPU.

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 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

Block Diagram

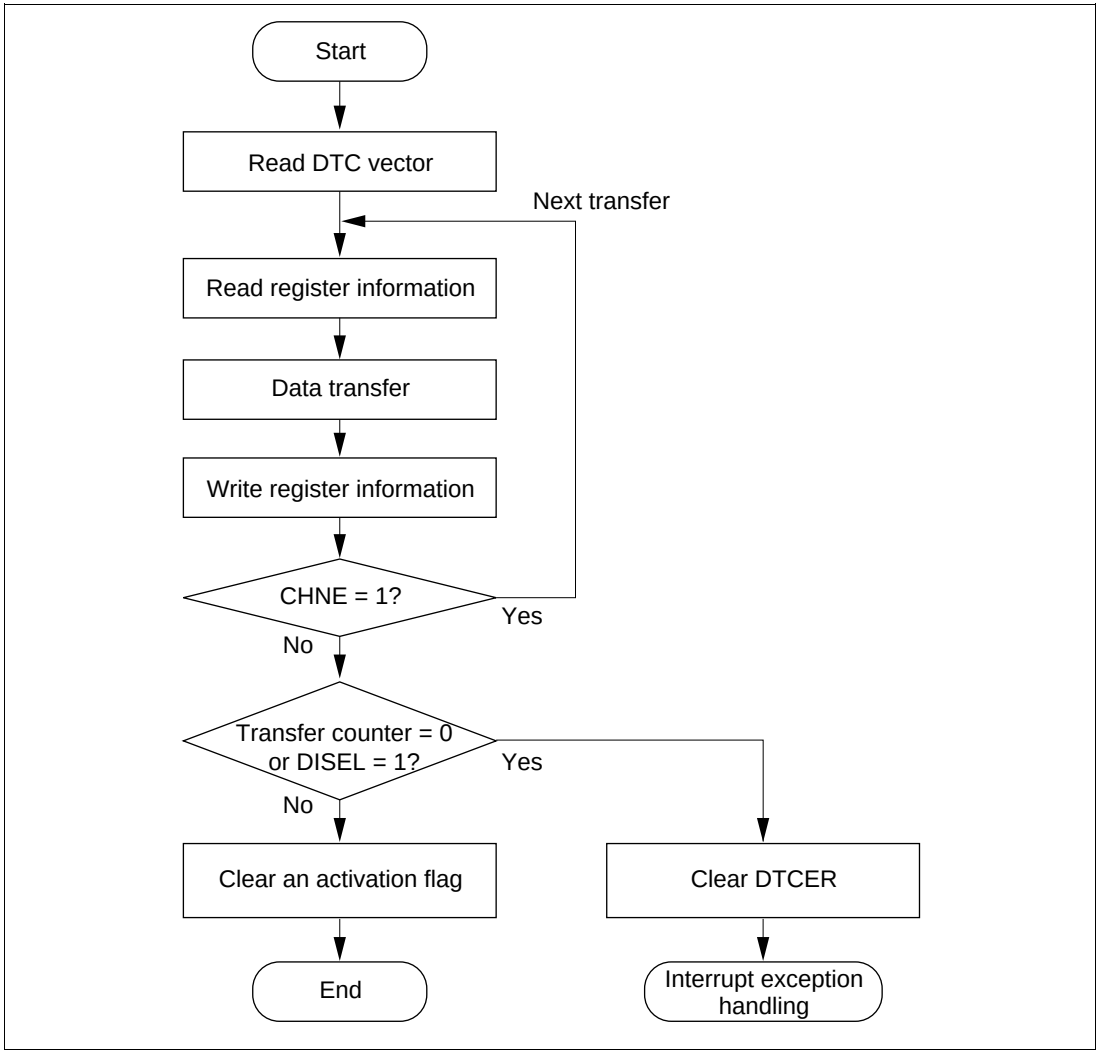


Legend:
MRA, MRB: DTC mode registers A and B
CRA, CRB: DTC transfer count registers A and B
SAR: DTC source address register
DAR: DTC destination address register
DTCERA to DTCERE: DTC enable registers A to E
DTVECR: DTC vector register

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. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation.

- Flowchart of DTC Operation



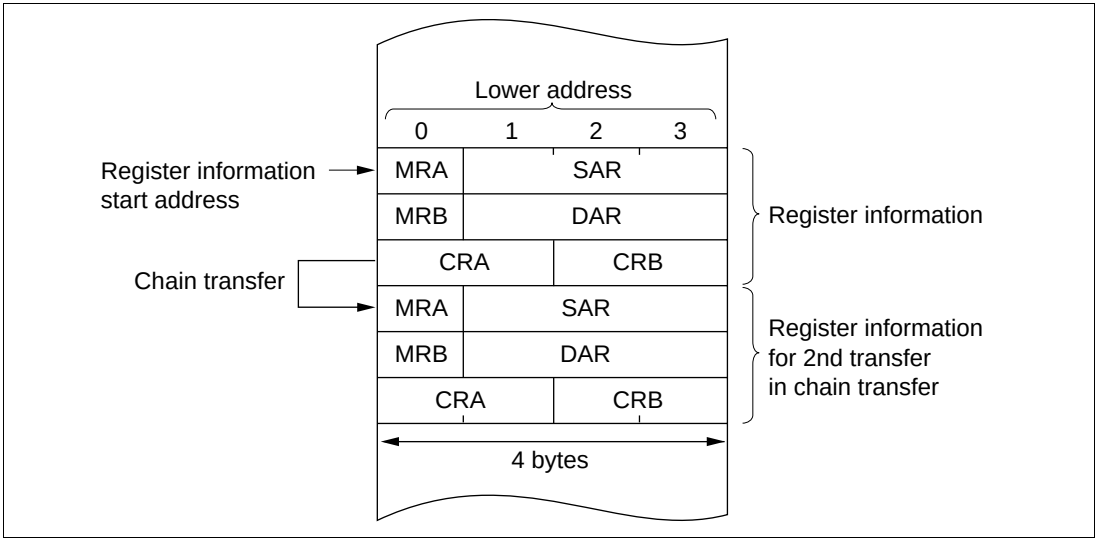
DTC Activation Sources: The DTC is activated by an interrupt or by a vector number write to the DTC vector register (DTVECR) by software. An interrupt request can be designated as a CPU interrupt source or a DTC activation source.

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'FFEC00 to H'FFEFFF).

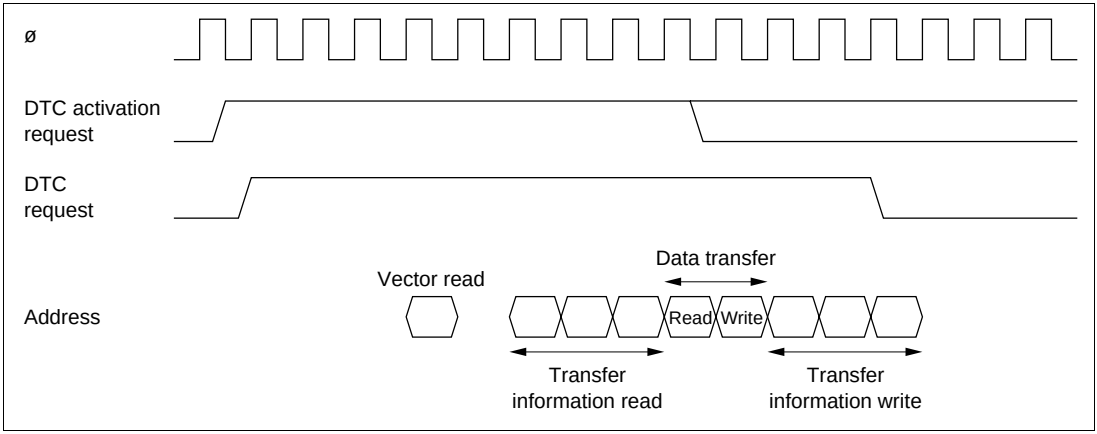
- Location of DTC Register Information in Address Space



- Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE	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	
ADI (A/D conversion end)	A/D	28	H'0438	DTCEA3	
ICIA (FR input capture A)	FRT	48	H'0460	DTCEA2	
ICIB (FRT input capture B)		49	H'0462	DTCEA1	
OCIA (FRT output compare A)		52	H'0468	DTCEA0	
OCIB (FRT output compare B)		54	H'046A	DTCEB7	
CMIA0 (TMR0 compare-match A)	TMR0	64	H'0480	DTCEB2	
CMIB0 (TMR0 compare-match B)		65	H'0482	DTCEB1	
CMIA1 (TMR1 compare-match A)	TMR1	68	H'0488	DTCEB0	
CMIB1 (TMR1 compare-match B)		69	H'048A	DTCEC7	
CMIAY (TMRY compare-match A)	TMRY	72	H'0090	DTCEC6	
CMIBY (TMRY compare-match B)		73	H'0092	DTCEC5	
IBF1 (IDR1 reception completed)	HIF	76	H'0098	DTCEC4	
IBF2 (IDR2 reception completed)		77	H'009A	DTCEC3	
RXI0 (reception completed 0)	SCI channel 0	81	H'04A2	DTCEC2	
TXI0 (transmit data empty 0)		82	H'04A4	DTCEC1	
RXI1 (reception completed 1)	SCI channel 1	85	H'04AA	DTCEC0	
TXI1 (transmit data empty 1)		86	H'04AC	DTCED7	
RXI2 (reception completed 2)	SCI channel 2	89	H'04B2	DTCED6	
TXI2 (transmit data empty 2)		90	H'04B4	DTCED5	
IICI0 (IIC0 1-byte transmission/ reception completed)	IIC0 (option)	92	H'04B8	DTCED4	
IICI1 (IIC1 1-byte transmission/ reception completed)	IIC1 (option)	94	H'04BC	DTCED3	Low

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

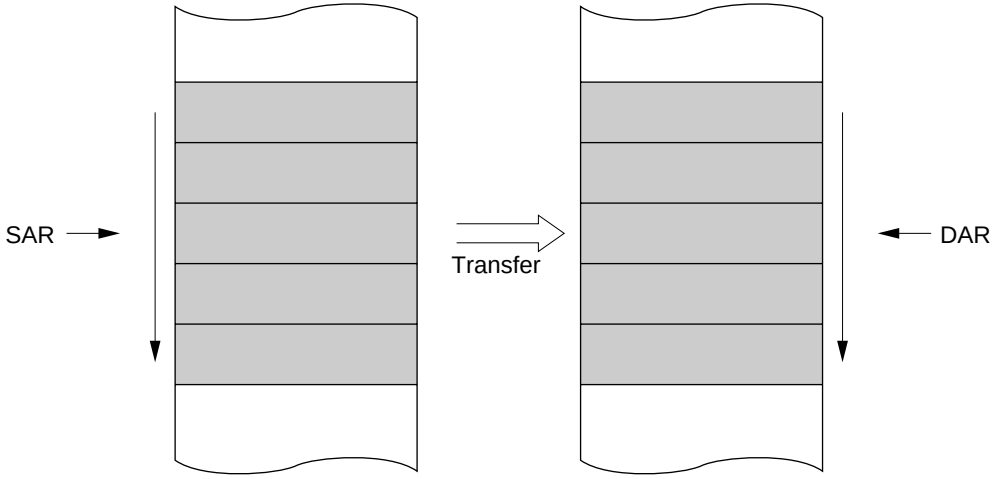
Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8	8	16	16
Access states			1	1	2	2	2	3	2	3
Execution status	Vector read	S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write	S_J	1	—	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	2	3+m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

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

Normal Mode: In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

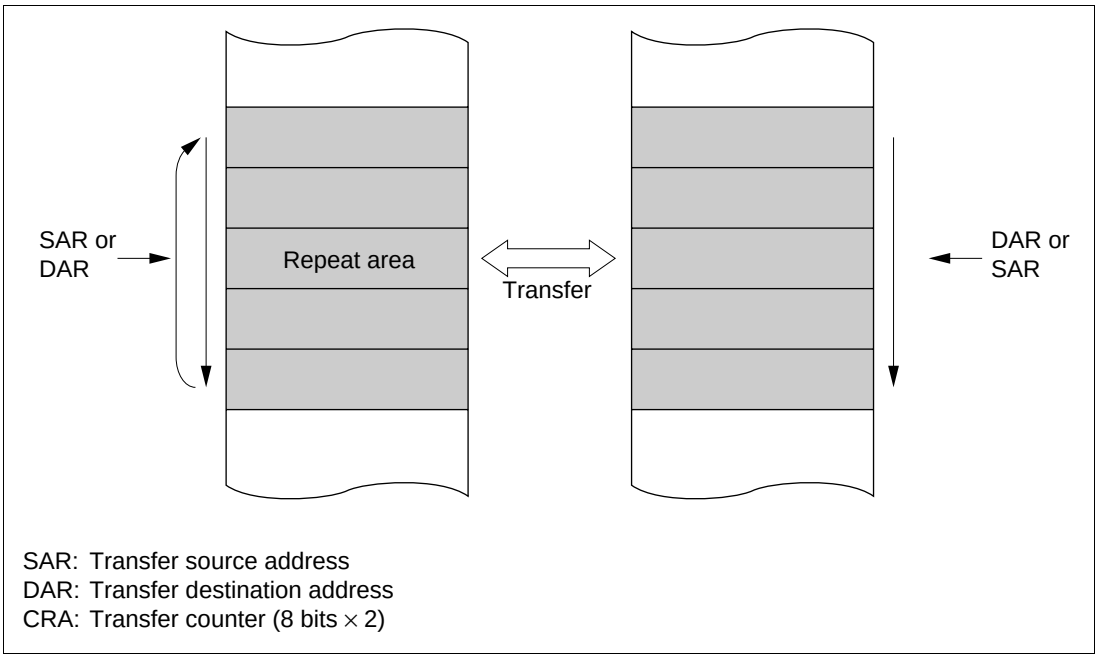


SAR: Transfer source address
DAR: Transfer destination address
CRA: Transfer counter

Memory Mapping in Normal Mode

Repeat Mode: In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state specified by the transfer counter and repeat area resumes and transfer is repeated.

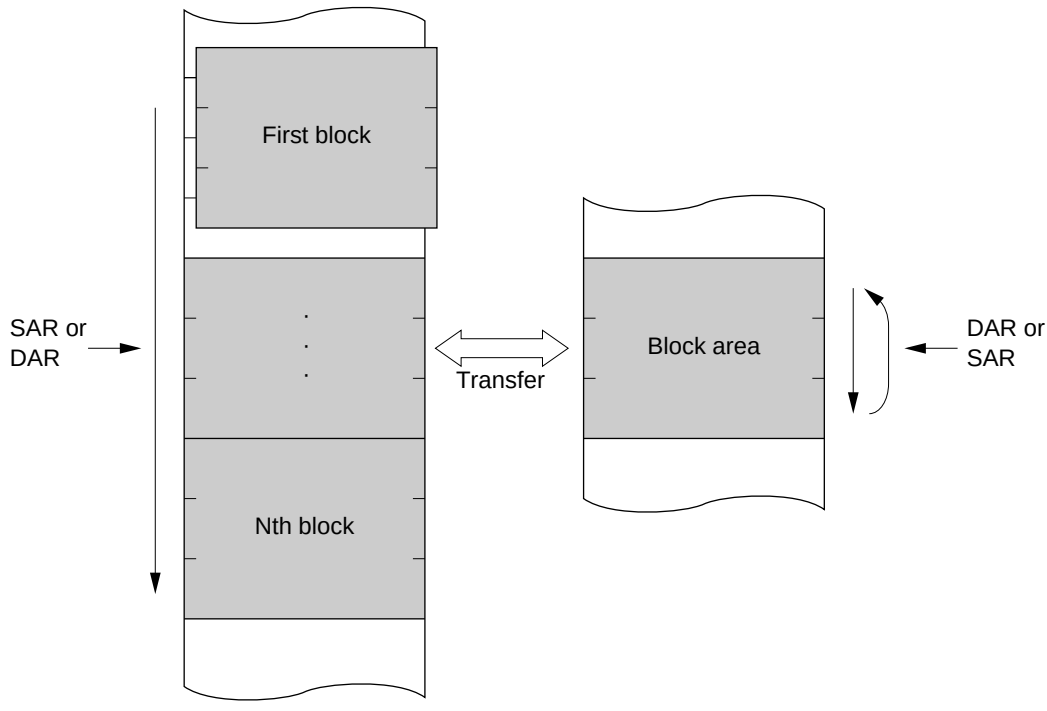


Memory Mapping in Repeat Mode

Block Transfer Mode: In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is specified as a block area.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified in the block area is restored. The other address register is then incremented, decremented, or left fixed.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt is requested.



SAR: Transfer source address
DAR: Transfer destination address
CRA: Block size (8 bits × 2)
CRB: Transfer counter

Memory Mapping in Block Transfer Mode

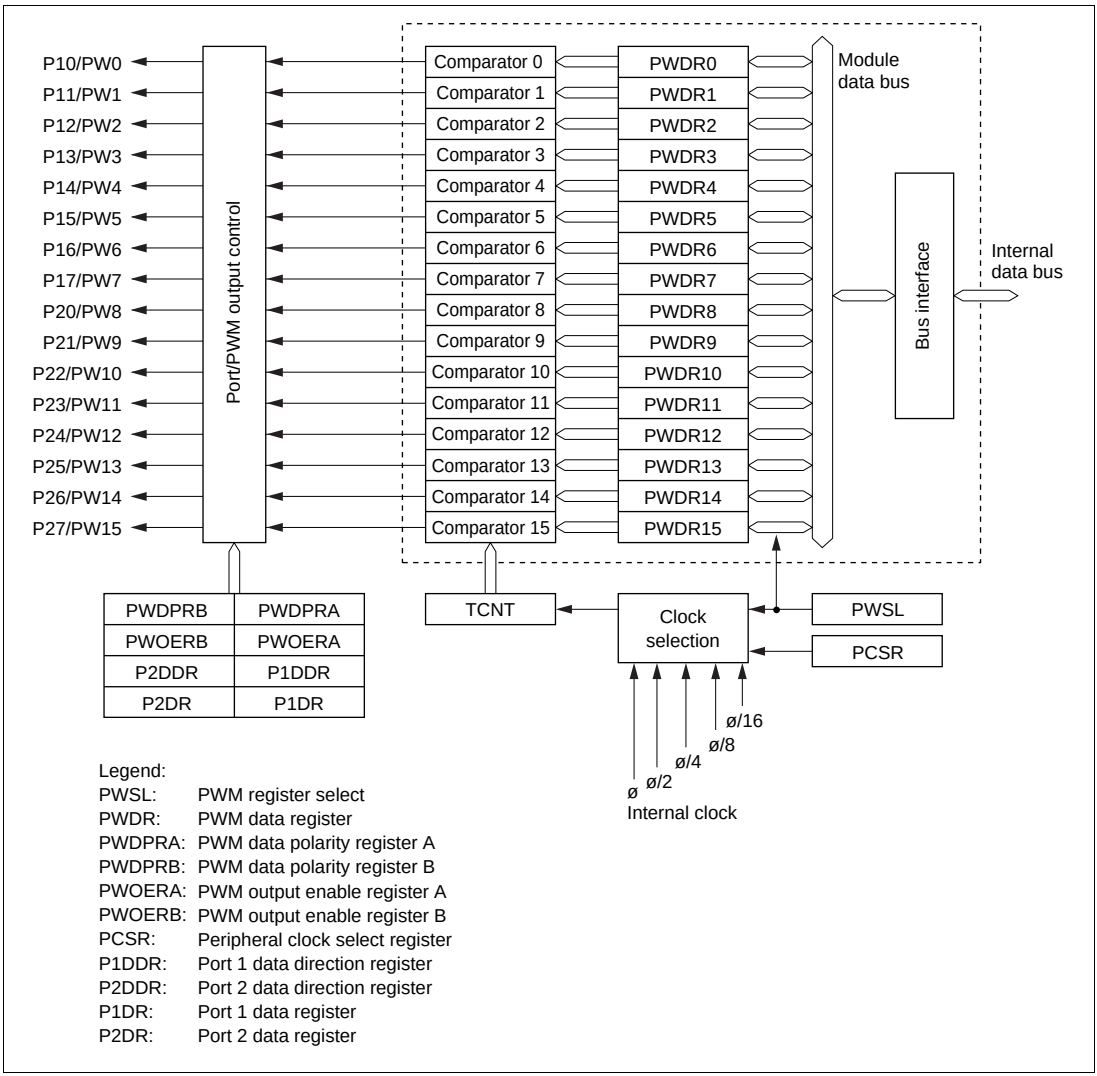
4.3 8-Bit PWM Timers (PWM) (Provided in the H8S/2148 Series and H8S/2138 Series)

These series have an on-chip pulse width modulation (PWM) timer module with sixteen outputs. Sixteen output waveforms are generated from a common time base, enabling PWM output with a high carrier frequency to be produced using pulse division. The PWM timer module has sixteen 8-bit PWM data registers (PWDRs), and an output pulse with a duty cycle of 0 to 100% can be obtained as specified by PWDR and the port data register (P1DR or P2DR).

Features

- Operable at a maximum carrier frequency of 1 MHz using pulse division (at 16 MHz operation)
- Duty cycles from 0 to 100% with 1/256 resolution (100% duty realized by port output)
- Direct or inverted PWM output, and software enable/disable control

Block Diagram



Legend:

PWSL:	PWM register select
PWDR:	PWM data register
PWDPRA:	PWM data polarity register A
PWDPRB:	PWM data polarity register B
PWOERA:	PWM output enable register A
PWOERB:	PWM output enable register B
PCSR:	Peripheral clock select register
P1DDR:	Port 1 data direction register
P2DDR:	Port 2 data direction register
P1DR:	Port 1 data register
P2DR:	Port 2 data register

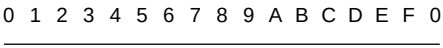
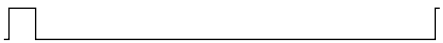
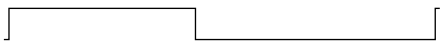
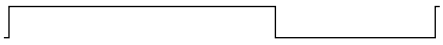
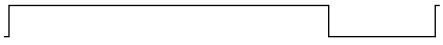
Block Diagram of 8-Bit PWM Timer

Operation

- Correspondence between PWM Data Register Contents and Output Waveform

  The upper 4 bits of PWDR specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16, as shown below.

- Duty Cycle of Basic Pulse

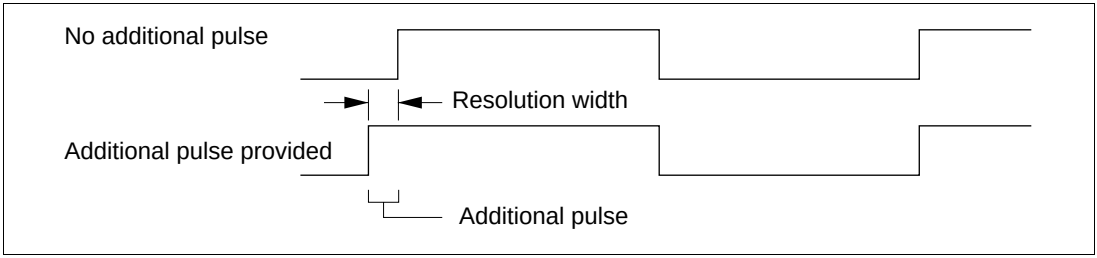
Upper 4 Bits	Basic Pulse Waveform (Internal)
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

- Position of Pulses Added to Basic Pulses

  The lower 4 bits of PWDR specify the position of pulses added to the 16 basic pulses, as shown below. An additional pulse consists of a high period (when OS = 0) with a width equal to the resolution, added before the rising edge of a basic pulse. When the upper 4 bits of PWDR are 0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same.

Lower	Basic Pulse No.															
4 Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0000																
0001																Yes
0010								Yes								Yes
0011								Yes				Yes				Yes
0100				Yes				Yes				Yes				Yes
0101				Yes				Yes				Yes		Yes		Yes
0110				Yes		Yes		Yes				Yes		Yes		Yes
0111				Yes		Yes		Yes		Yes		Yes		Yes		Yes
1000		Yes		Yes		Yes		Yes		Yes		Yes		Yes		Yes
1001		Yes		Yes		Yes		Yes		Yes		Yes		Yes	Yes	Yes
1010		Yes		Yes		Yes	Yes	Yes		Yes		Yes		Yes	Yes	Yes
1011		Yes		Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1100		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1101		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1110		Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1111		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

- Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = 1000)



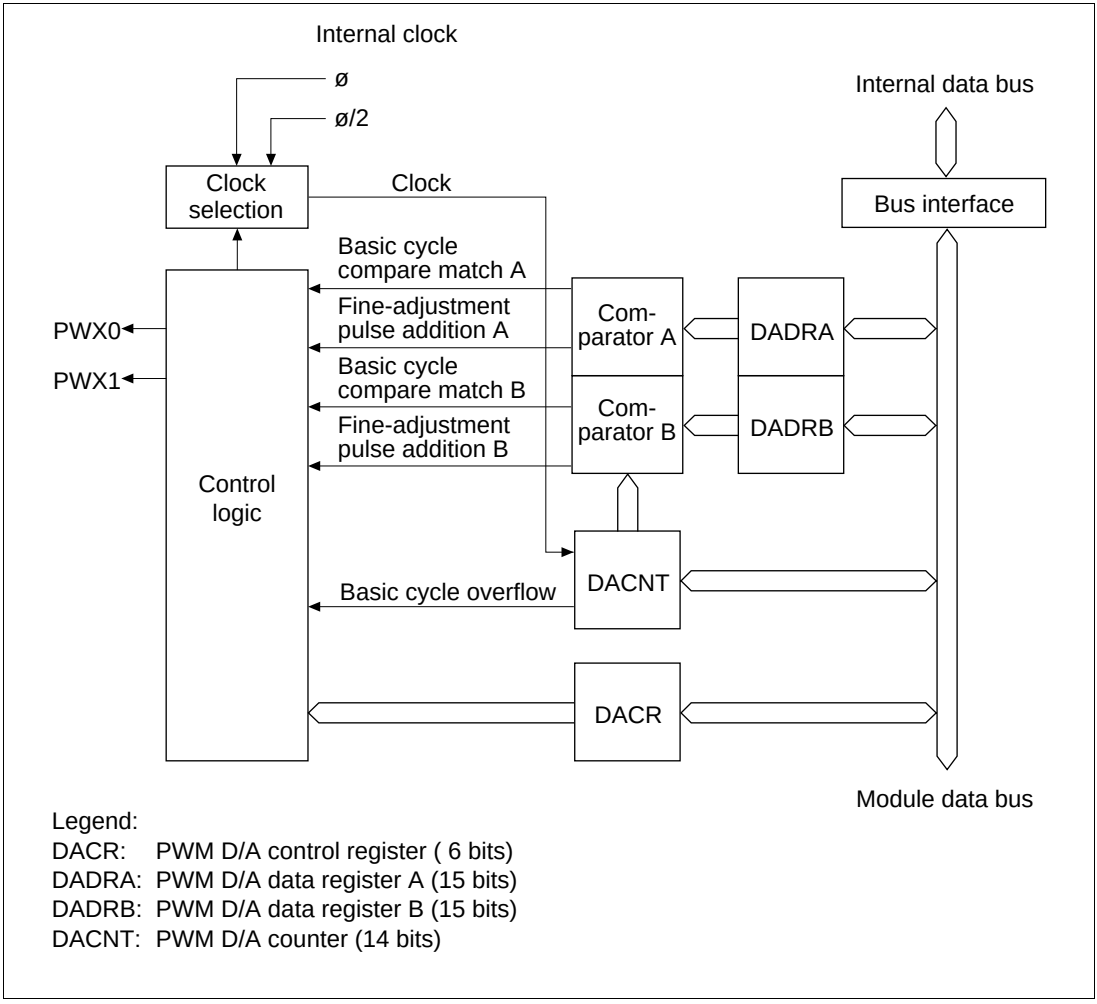
4.4 14-Bit PWM Timer (PWMX)

These series are equipped with a two-channel PWM D/A converter based on a 14-bit counter. Fine adjustment pulses are added to the basic pulses, enabling low-ripple D/A conversion results to be obtained. Connection of an external low-pass filter enables this module to be used as a 14-bit D/A converter. The two PWM channels share the same counter (DACNT) and control register (DACR).

Features

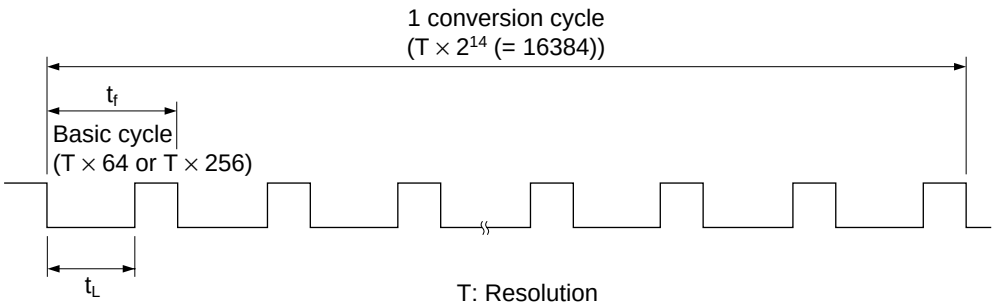
- Selection of two counter input clocks
  - — ϕ or $\phi/2$ internal clock can be selected
- Selection of two basic cycles
  - — Counter input clock cycle × 64 (0 to 255 fine adjustment pulses added to 256 basic pulses)
  - — Counter input clock cycle × 256 (0 to 63 fine adjustment pulses added to 64 basic pulses)
- Selection of four operating speeds
  - — Any of four operating clocks can be selected (two basic cycles × two resolutions)

Block Diagram



Block Diagram of 14-Bit PWM Timer

PWM (D/A) Operation



T: Resolution

$TL = \sum_{n=1}^{m} t_{Ln}$ (when OS = 0)

(When CFS = 0, m = 256; when CFS = 1, m = 64)

4.5 16-Bit Free-Running Timer (FRT)

The 16-bit free-running timer is provided with output compare functions (two output compare registers, two output pins) and input capture functions (four input capture register, four input pins, buffer operation capability).

Features

- Selection of four counter input clocks
  - — Selection can be made from three internal clocks and an external clock (allowing external event counting)
- Output compare functions: two independent converters
  - — Two independent waveforms can be output
- Input capture functions
  - — Rising or falling edge selectable as trigger
  - — Four independent input capture registers, with provision for buffer operation* specification

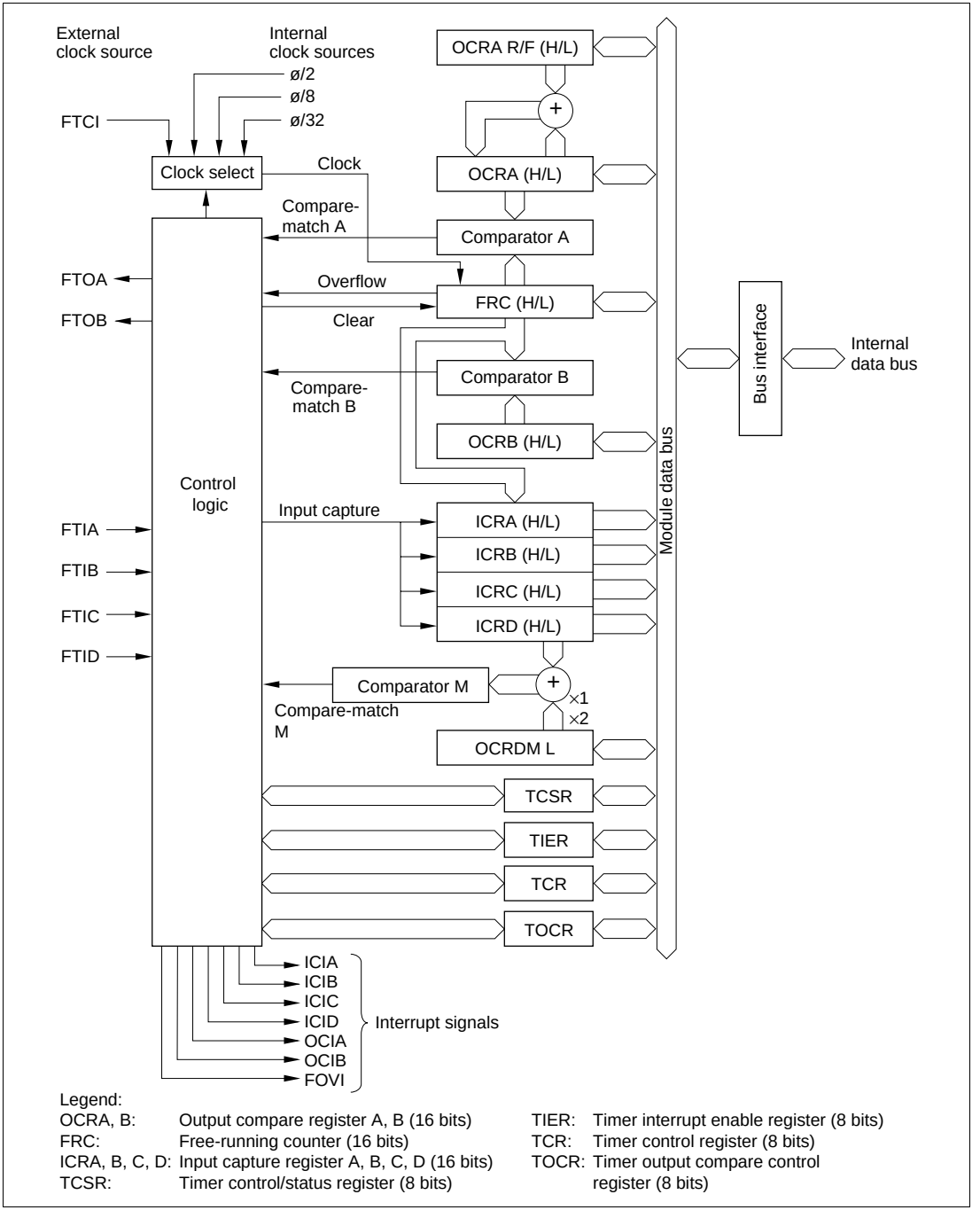
Note: * In buffer operation, the four independent 16-bit input capture registers function as two pairs, with one register in each pair used as a buffer for the other register.

- Counter can be cleared under program control

  The free-running counters can be cleared on compare-match A.
- Seven independent interrupts

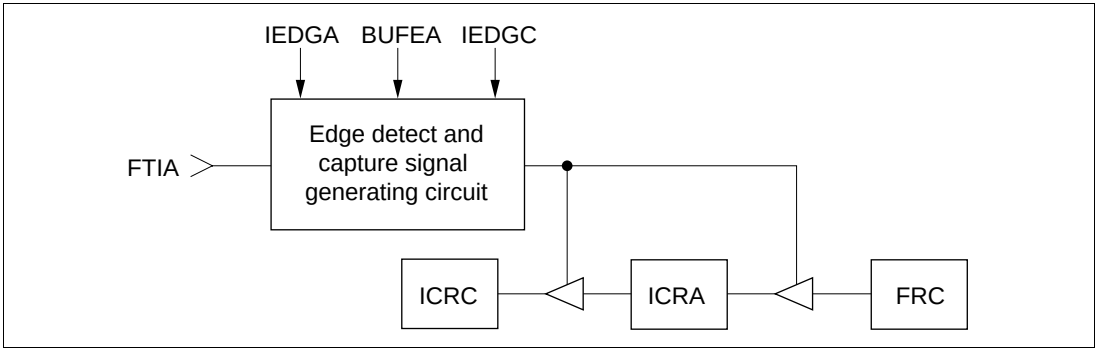
  Two compare-match interrupts, four input capture interrupts, and one overflow interrupt are requested independently.

Block Diagram



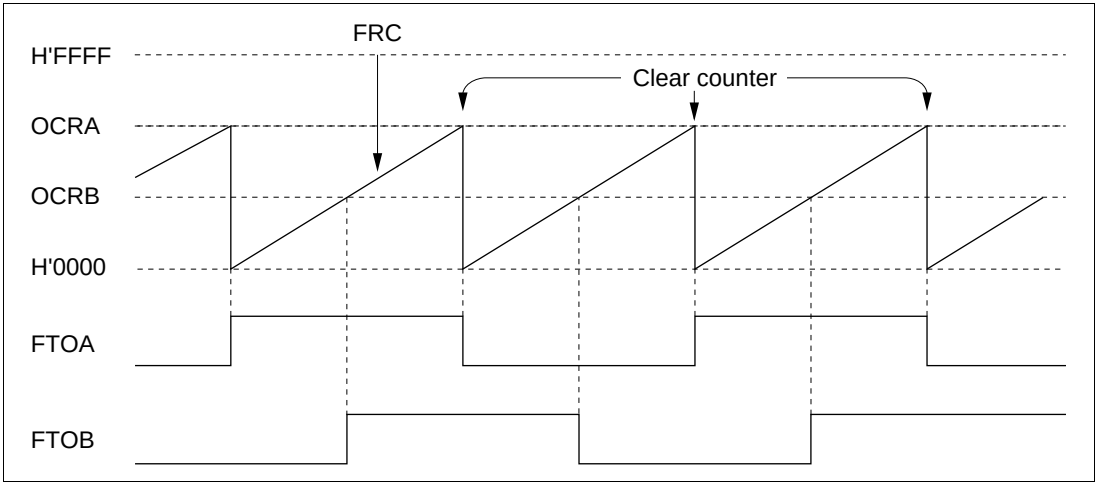
Block Diagram of 16-Bit Free-Running Timer

Input Capture Buffering



Input Capture Buffering (Example)

Pulse Output: In the example below, the free-running timer is used to generate pulse outputs with a 50% duty cycle and arbitrary phase relationship.



Pulse Output (Example)

4.6 8-Bit Timers (TMR)

These series include an 8-bit timer module with two channels (TMR0 and TMR1). The 8-bit timer module can be used as a multifunction timer in a variety of applications, such as generation of a rectangular-wave output with an arbitrary duty cycle. Also provided are two similar 8-bit timer channels (TMRX and TMRY), which can be used in a connected configuration using the timer connection function.

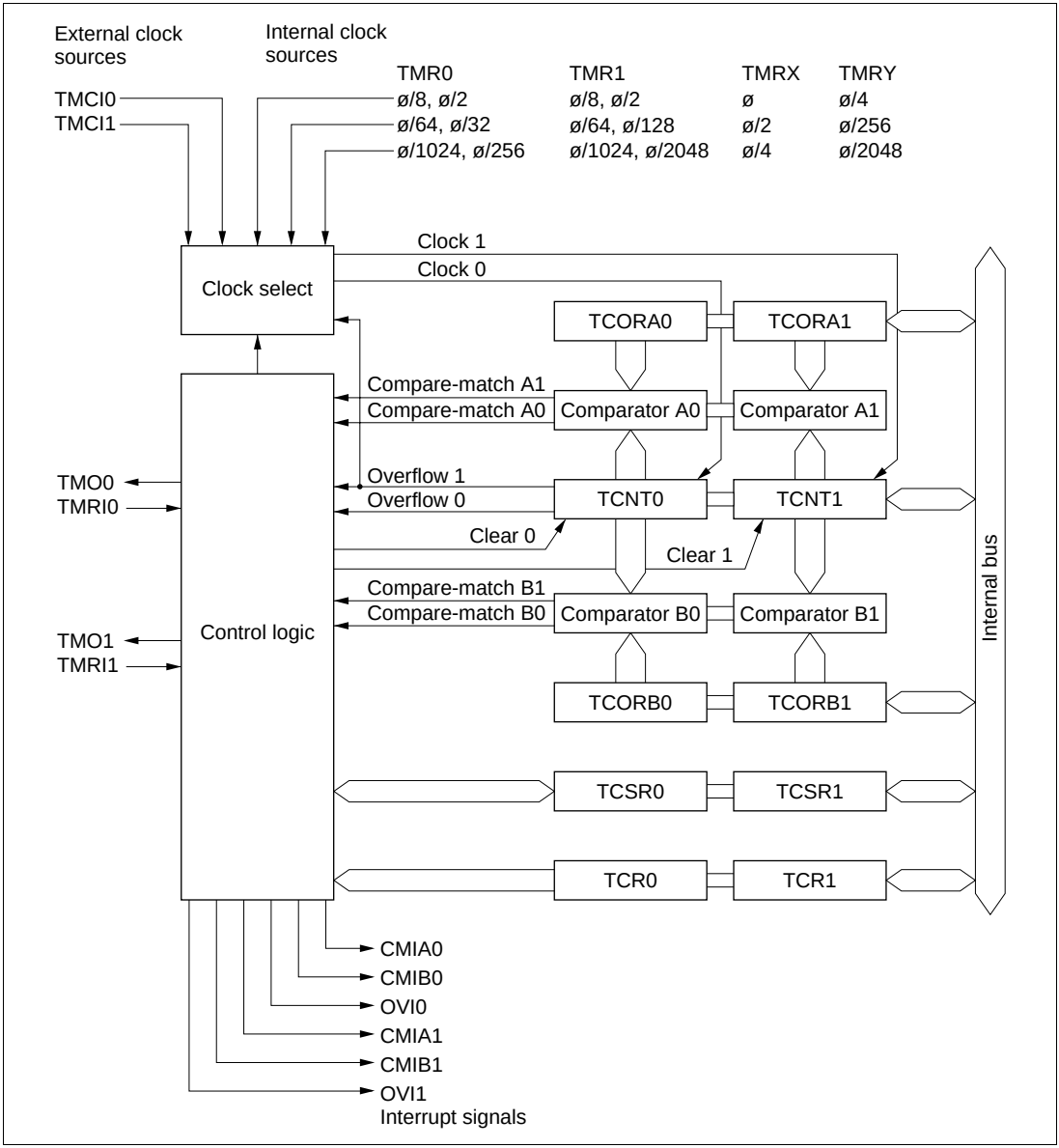
Features

- Selection of clock sources
  - — TMR0, TMR1: The counter input clock can be selected from six internal clocks and an external clock (enabling use as an external event counter).
  - — TMRX, TMRY: The counter input clock can be selected from three internal clocks and an external clock (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 controlled by two compare-match signals
  - — The timer output signal in each channel is controlled by two independent compare-match signals, enabling the timer to be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle.

    (Note: TMRY does not have a timer output pin.)
- Cascading of the two channels (TMR0, TMR1)
  - — Operation as a 16-bit timer can be performed using channel 0 as the upper half and channel 1 as the lower half (16-bit counter mode).
  - — Channel 1 can be used to count channel 0 compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel
  - — TMR0, TMR1, TMRY: Two compare-match interrupts and one overflow interrupt can be requested independently.
  - — TMRX: One input capture source is available.

Note: TMRX is not provided in the H8S/2144 Series or H8S/2134 Series.

Block Diagram



Block Diagram of 8-Bit Timer

Interrupt Sources and Data Transfer Controller (DTC) Activation

- 8-Bit Timer TMR0 and TMR1 Interrupt Sources

Interrupt Source	Description	DTC Activation	Priority
CMIA	Interrupt initiated by CMFA	Possible	High
CMIB	Interrupt initiated by CMFB	Possible	↑
OVI	Interrupt initiated by OVF	Not possible	Low

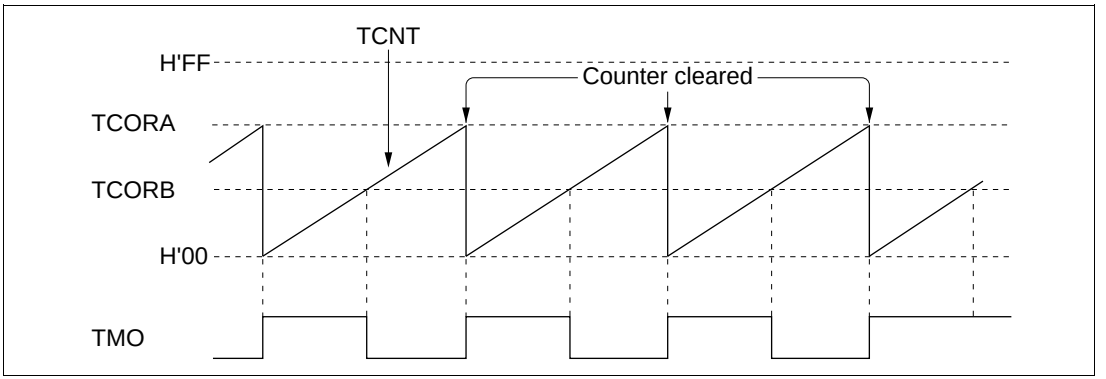
- 8-Bit Timer TMRX Interrupt Source

Interrupt Source	Description	DTC Activation
ICIX	Interrupt initiated by ICF	Not possible

- 8-Bit Timer TMRY Interrupt Sources

Interrupt Source	Description	DTC Activation	Priority
CMIA	Interrupt initiated by CMFA	Possible	High
CMIB	Interrupt initiated by CMFB	Possible	↑
OVI	Interrupt initiated by OVF	Not possible	Low

Example of Pulse Output: A TCR setting is made so that counter clearing is performed by a TCORA compare match. The cycle is set in TCORA, and the duty in TCORB. This enables the pulses shown below to be output continuously without software intervention.



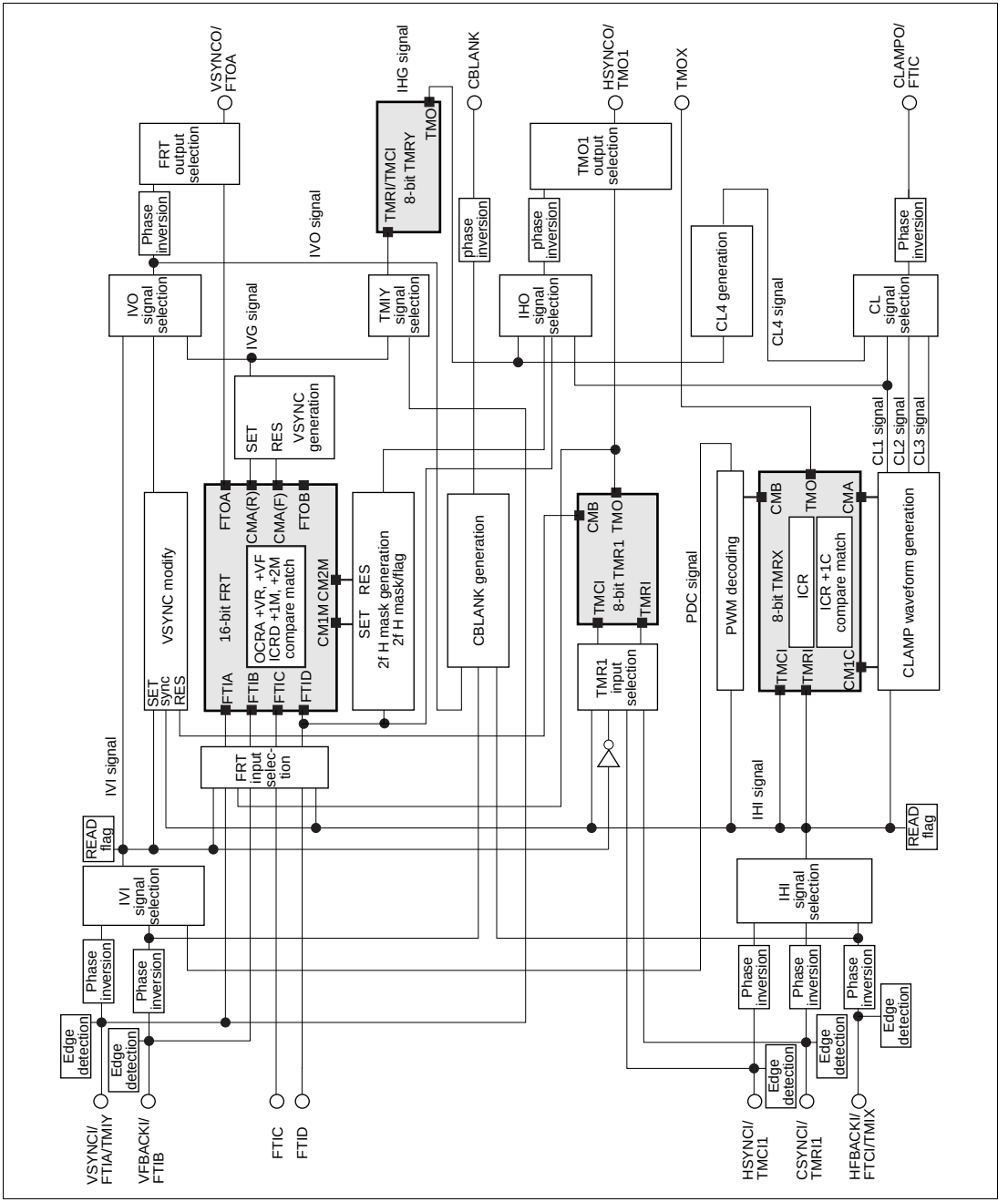
Example of Pulse Output

4.7 Timer Connection (Provided in the H8S/2148 Series and H8S/2138 Series)

The H8S/2148 Series and H8S/2138 Series allow interconnection between the input/output of the single free-running timer (FRT) channel and three 8-bit timer channels (TMR1, TMRX, and TMRY). This capability can be used to implement complex functions such as PWM decoding and clamp waveform output. All the timers are initially set for independent operation.

Features

- Five input pins and four output pins, all of which can be designated for phase inversion.
  - Positive logic is assumed for all signals used within the timer connection facility.
- An edge-detection circuit is connected to the input pins, simplifying signal input detection.
- TMRX can be used for PWM input signal decoding and clamp waveform generation.
- An external clock signal divided by TMR1 can be used as the FRT capture input signal.
- An internal synchronization signal can be generated using FRT and TMRY.
- A signal generated/modified using an input signal and timer connection can be selected and output.



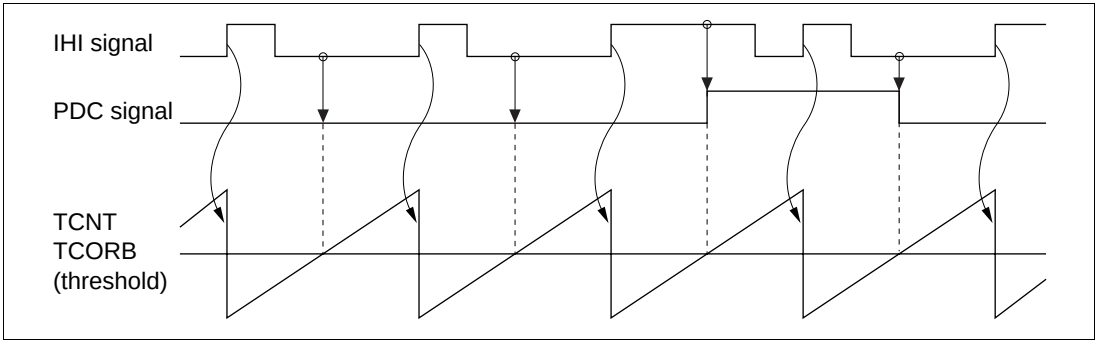
Block Diagram of Timer Connection

Operation

- PWM Decoding (PDC Signal Generation)

  Timer connection TMRX can be used to decode a PWM signal in which 0 and 1 are represented by the pulse width.

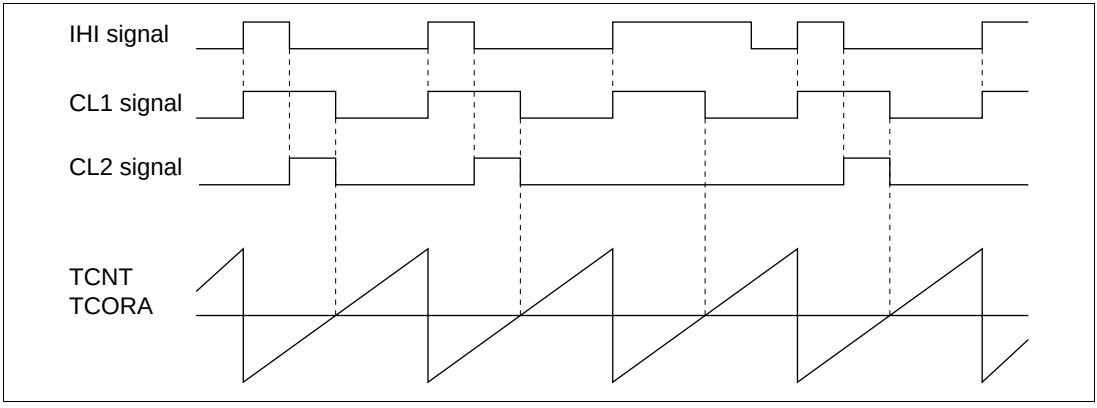
- Timing Chart for PWM Decoding



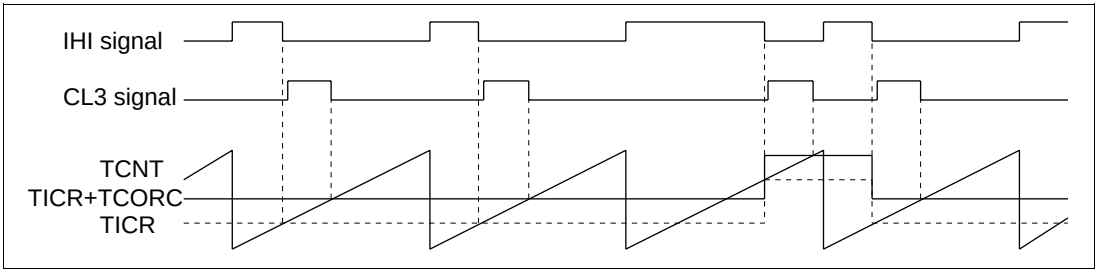
- Clamp Waveform Generation (CL1/CL2/CL3 Signal Generation)

  Timer connection TMRX can be used to generate signals with different duty cycles and rising/falling edges (clamp waveforms) in synchronization with the input signal (IHI signal). Three clamp waveforms can be generated: the CL1, CL2, and CL3 signals.

- Timing Chart for Clamp Waveform Generation (CL1 and CL2 Signals)



- Timing Chart for Clamp Waveform Generation (CL3 Signal)

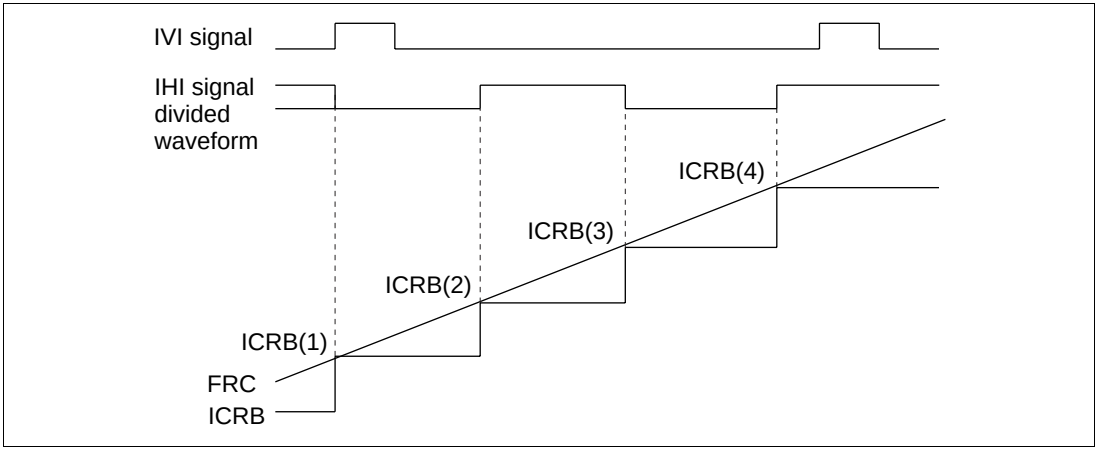


- Measurement of 8-Bit Timer Divided Waveform Period

  Timer connection TMR1 and the free-running timer (FRT) can be used to measure the period of an IHI signal divided waveform. Since TMR1 can be cleared by a rising edge of the IVI signal, the rise and fall of the IHI signal divided waveform can be virtually synchronized with the IVI signal. This enables period measurement to be carried out efficiently.

  The period of the IHI signal divided waveform by (ICRD(3) – ICRD(2)) × the resolution.

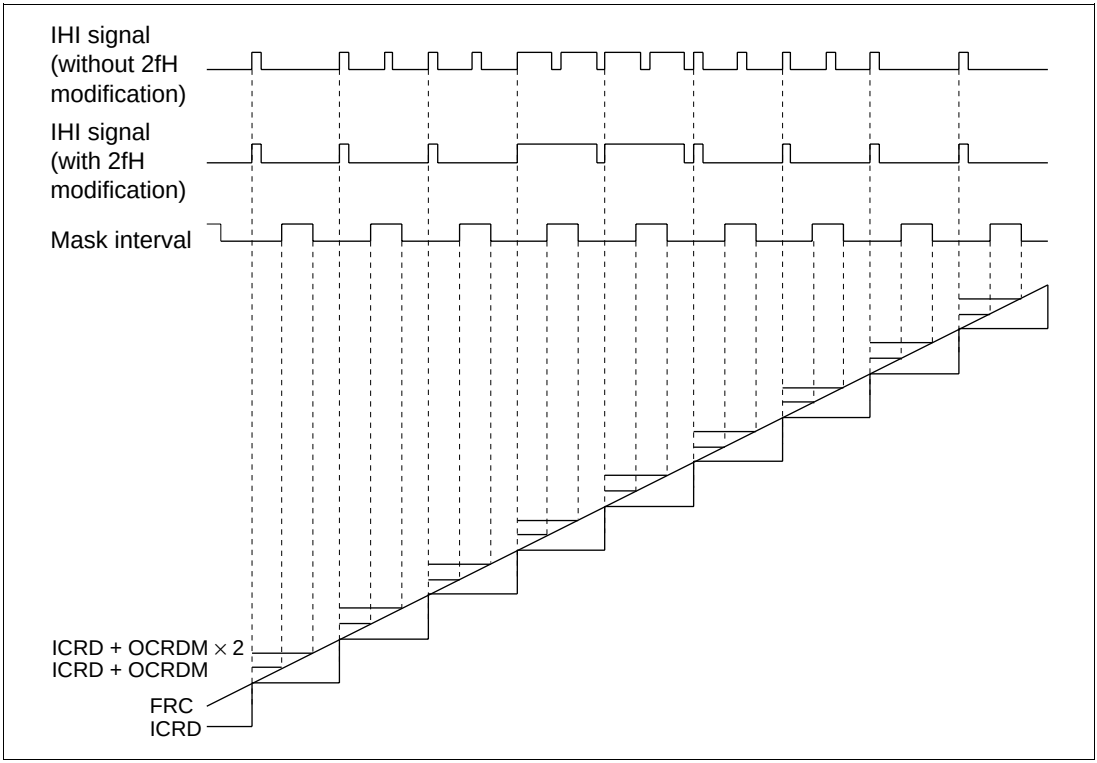
- Timing Chart for Measurement of IVI Signal and IHI Signal Divided Waveform Periods



- IHI Signal and 2fH Modification

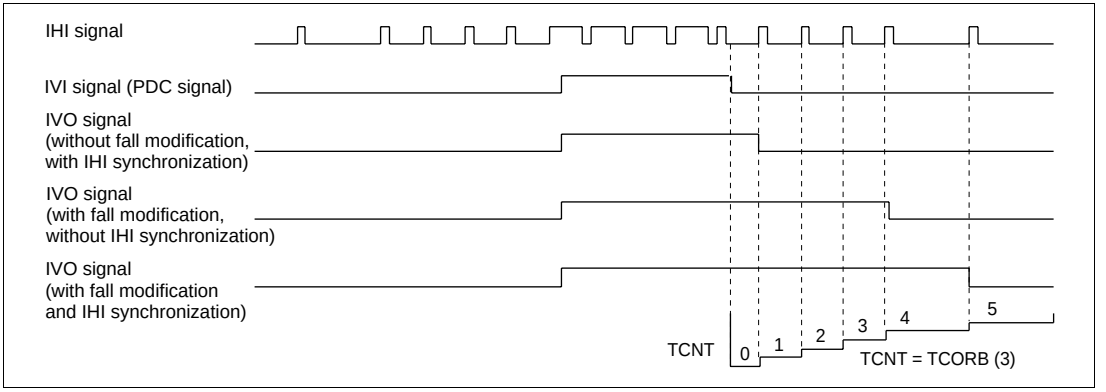
  By using the timer connection FRT, even if there is a part of the IHI signal with twice the frequency, this can be eliminated.

- 2fH Modification Timing Chart



- IVI Signal Fall Modification and IHI Synchronization

  By using the timer connection TMR1, the fall of the IVI signal can be shifted backward by the specified number of IHI signal waveforms. Also, the fall of the IVI signal can be synchronized with the rise of the IHI signal.

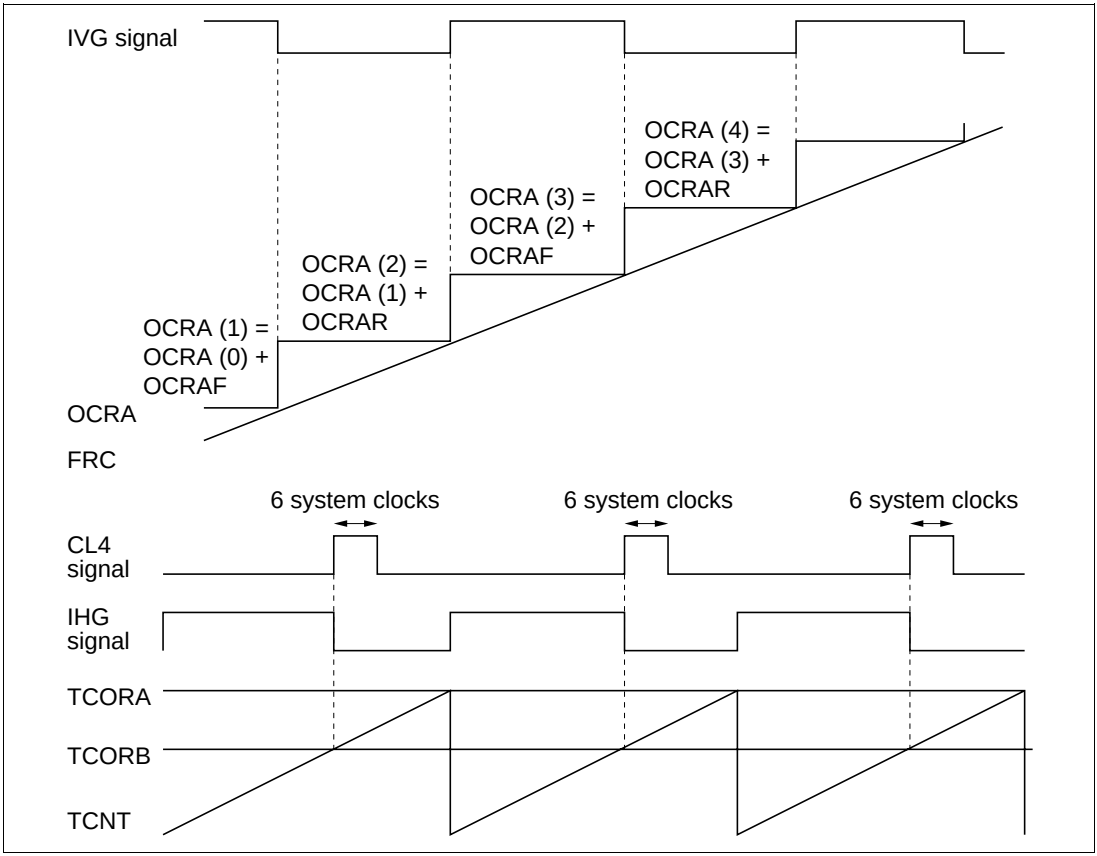


Fall Modification/IHI Synchronization Timing Chart

- Internal Synchronization Signal Generation (IHG, IVG, CL4 Signal Generation)

  By using the timer connection FRT and TMRY, it is possible to automatically generate internal signals (IHG and IVG signals) corresponding to the IHI and IVI signals. CL4 signal can also be generated in synchronization with the IHG signal.

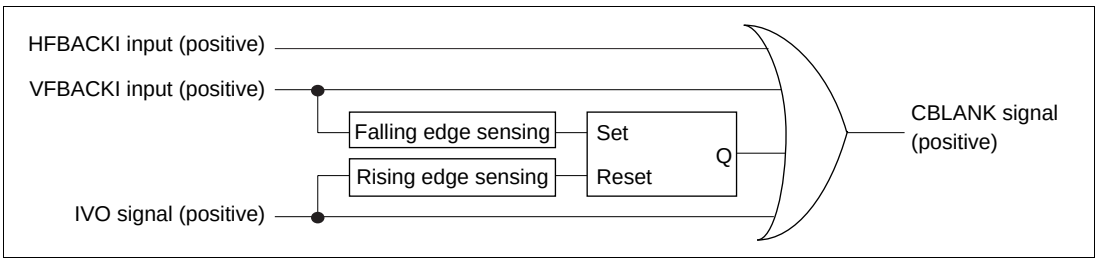
- IVG Signal, IHG Signal, and CL4 Signal Timing Chart



- CBLANK Output

  Using the signals generated/selected with timer connection, it is possible to generate a waveform based on the composite synchronization signal (blanking waveform).

- CBLANK Output Waveform Generation



4.8 Watchdog Timer (WDT)

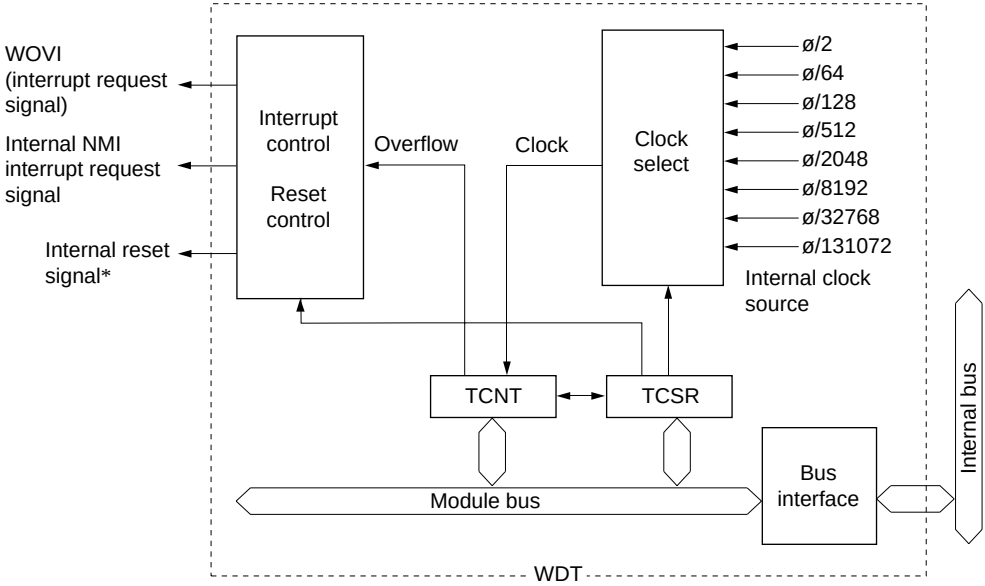
These series have an on-chip watchdog timer with two channels (WDT0, WDT1) for monitoring system operation.

When this watchdog function is not needed, the WDT can be used as an interval timer.

Features

- Choice of 8 (WDT0) or 16 (WDT1) counter input clocks
  - — Maximum WDT interval: system clock period × 131072 × 256
  - — Subclock can be selected for the WDT1 input counter
    Maximum interval when the subclock is selected: subclock period × 256 × 256
- Can be used as an interval timer
- Internal reset or internal interrupt generated when the timer counter overflows
  - — Choice of internal reset or NMI interrupt generation in watchdog timer mode
- Interrupt generation in interval timer mode
  - — When the counter overflows, the WDT generates an interval timer interrupt.

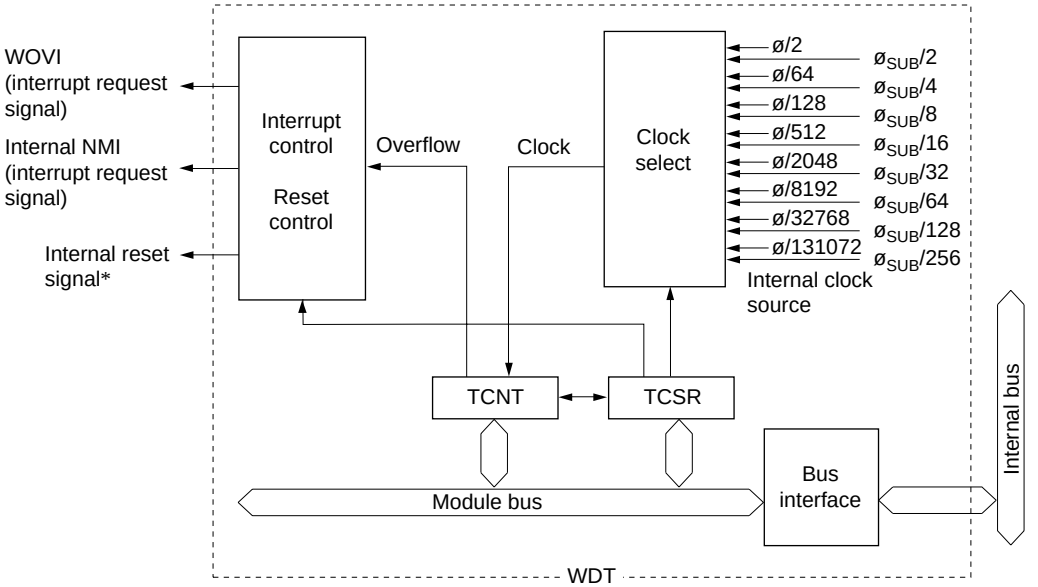
Block Diagram



Legend:
TCSR: Timer control/status register
TCNT: Timer counter

Note: * The internal reset signal can be generated by means of a register setting.

Block Diagram of WDT0

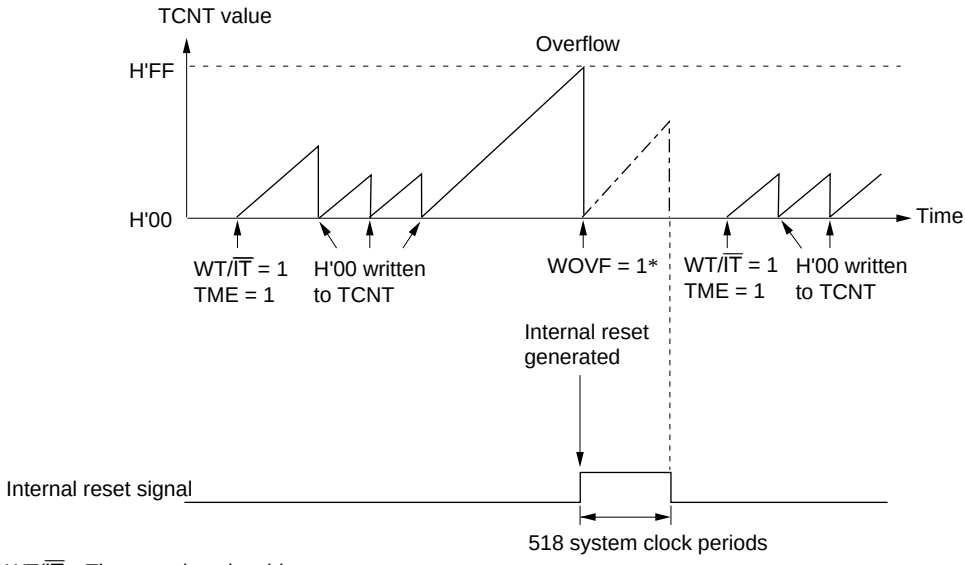


Legend:
TCSR: Timer control/status register
TCNT: Timer counter

Note: * The internal reset signal can be gnerated by means of a register setting.

Block Diagram of WDT1

Watchdog Timer Operation: The example below shows this module used as a watchdog timer. The timer counter (TCNT) starts counting up using the specified clock.

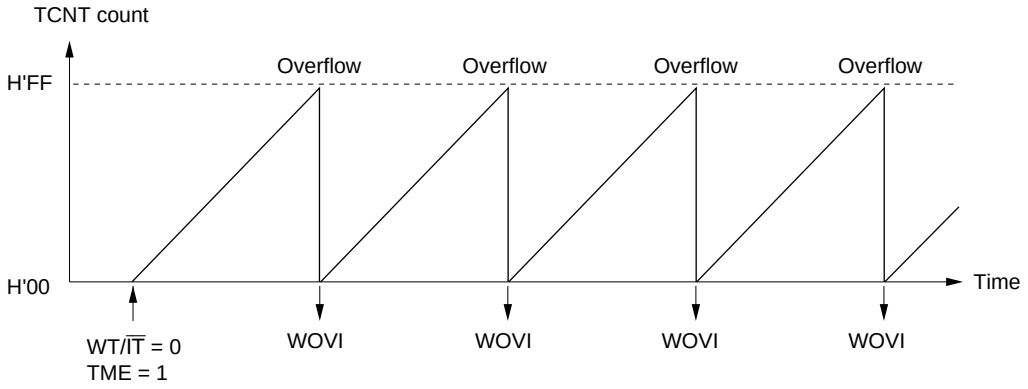


WT/$\overline{IT}$: Timer mode select bit
TME: Timer enable bit

Note: * Cleared to 0 by an internal reset when WOVF is set to 1. XRST is cleared to 0.

Operation in Watchdog Timer Mode

Interval Timer Operation: To use the WDT as an interval timer, clear the WT/$\overline{IT}$ bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown below. This function can be used to generate interrupt requests at regular intervals.



Legend:
WOVI: Interval timer interrupt request generation

Interval Timer Operation

4.9 Serial Communication Interface (SCI, IrDA)

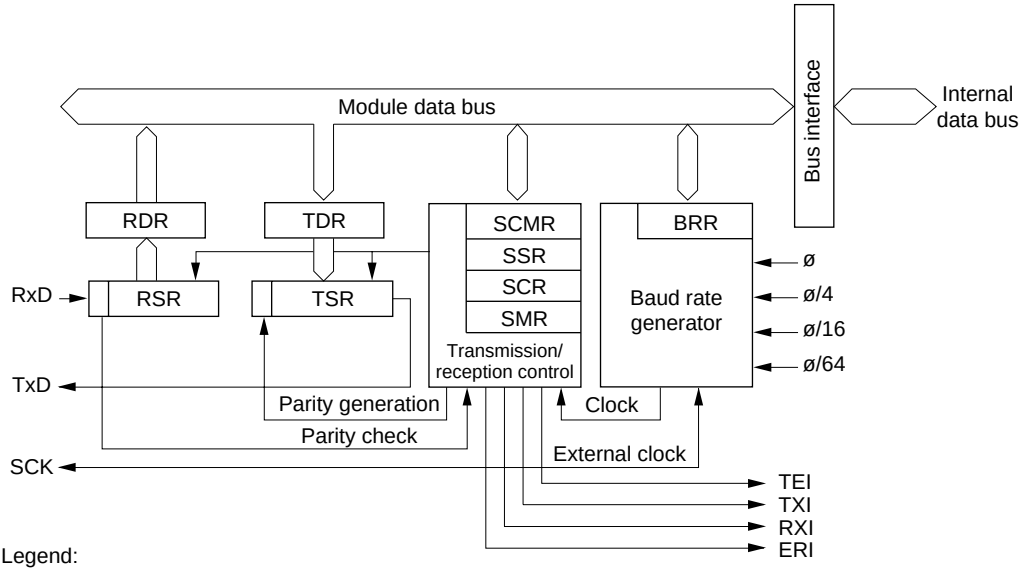
These series are equipped with a 3-channel serial communication interface (SCI). The SCI can handle both asynchronous and clocked synchronous serial communication (SCI0, SCI1, SCI2). A function is also provided for serial communication between processors (multiprocessor communication function).

SCI2 can transmit and receive IrDA communication waveforms based on IrDA specification version 1.0.

Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability
- Data register double-buffering enables continuous transmission/reception
- On-chip dedicated baud rate generator allows any bit rate to be selected
- Selection of internal clock from baud rate generator or external clock input (SCK pin) as serial clock source
- Detection of three receive errors
  - Overrun errors, framing errors, and parity errors can be detected
- Break detection
- 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 data transfer controller (DTC) to execute data transfer
- Built-in multiprocessor communication function
- 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)

Block Diagram



Legend:

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
SCMR:	Serial interface mode register
BRR:	Bit rate register
KBCOMP:	Keyboard comparator control register (for SCI2 IrDA communication)

Block Diagram of SCI

SCI Interrupt Sources and Data Transfer Controller (DTC) Activation

Channel	Interrupt Source	Description	DTC Activation	Priority*
0	ERI0	Interrupt due to receive error (ORER, FER, or PER)	Not possible	High
	RXI0	Interrupt due to receive data full state (RDRF)	Possible	
	TXI0	Interrupt due to transmit data empty state (TDRE)	Possible	
	TEI0	Interrupt due to transmission end (TEND)	Not possible	
1	ERI1	Interrupt due to receive error (ORER, FER, or PER)	Not possible	
	RXI1	Interrupt due to receive data full state (RDRF)	Possible	
	TXI1	Interrupt due to transmit data empty state (TDRE)	Possible	
	TEI1	Interrupt due to transmission end (TEND)	Not possible	
2	ERI2	Interrupt due to receive error (ORER, FER, or PER)	Not possible	
	RXI2	Interrupt due to receive data full state (RDRF)	Possible	
	TXI2	Interrupt due to transmit data empty state (TDRE)	Possible	
	TEI2	Interrupt due to transmission end (TEND)	Not possible	Low

Note: * The table shows the initial state immediately after a reset. Relative channel priorities can be changed by means of the interrupt controller.

SCI Asynchronous Communication: Asynchronous mode is a serial communication mode in which synchronization is achieved on a character by character basis, using a start bit and one or two stop bits.

- Twelve serial data transfer formats
  - — Data length: 7 or 8 bits
  - — Stop bit length: 1 or 2 bits
  - — Parity: Even/odd/none
  - — Multiprocessor bit: 1 or 0
- Selection of on-chip baud rate generator or external clock from SCK pin as clock source
- Transmit/receive clock can be output from SCK pin

- Break detection
  — A break can be detected by reading the RxD pin level directly in case of a framing error
- Multiprocessor communication capability

Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	—	1	0	S	8-bit data								MPB	STOP	
0	—	1	1	S	8-bit data								MPB	STOP	STOP
1	—	1	0	S	7-bit data							MPB	STOP		
1	—	1	1	S	7-bit data							MPB	STOP	STOP	

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Multiprocessor Communication Function: A multiprocessor format, in which a multiprocessor bit is added to the transfer data, can be used for serial communication, enabling data transfer to be performed among a number of processors.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

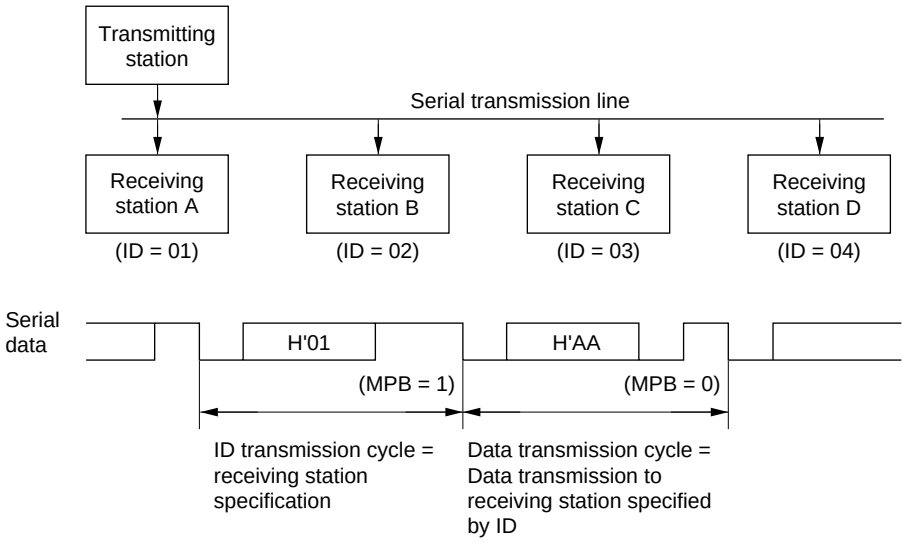
The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received.

SCI Synchronous Communication: In synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

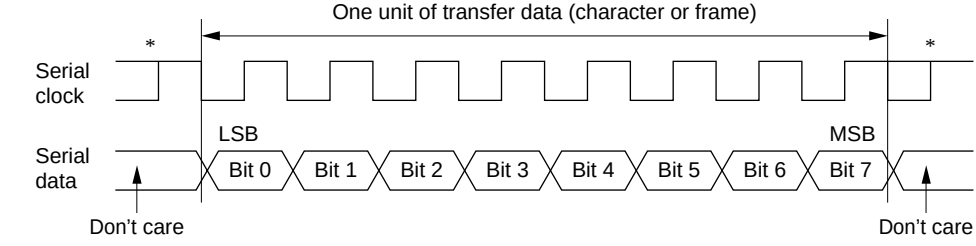
- Data length: 8 bits per character
- Overrun error detection
- Selection of on-chip baud rate generator or external clock from SCK pin as transmit/receive clock source
- Selection of LSB-first or MSB-first transfer
- Communication is possible with chips provided with a synchronous mode, such as the H8 Series, HD64180, and HD6301

Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)



Legend:
MPB: Multiprocessor bit

Data Format in Synchronous Communication



Note: * High except in continuous transfer

BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate	Operating Frequency ø (MHz)											
	ø = 2 MHz		ø = 4 MHz		ø = 8 MHz		ø = 10 MHz		ø = 16 MHz		ø = 20 MHz	
(bit/s)	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—								
250	2	124	2	249	3	124	—	—	3	249		
500	1	249	2	124	2	249	—	—	3	124	—	—
1 k	1	124	1	249	2	124	—	—	2	249	—	—
2.5 k	0	199	1	99	1	199	1	249	2	99	2	124
5 k	0	99	0	199	1	99	1	124	1	199	1	249
10 k	0	49	0	99	0	199	0	249	1	99	1	124
25 k	0	19	0	39	0	79	0	99	0	159	0	199
50 k	0	9	0	19	0	39	0	49	0	79	0	99
100 k	0	4	0	9	0	19	0	24	0	39	0	49
250 k	0	1	0	3	0	7	0	9	0	15	0	19
500 k	0	0*	0	1	0	3	0	4	0	7	0	9
1 M			0	0*	0	1			0	3	0	4
2.5 M							0	0*			0	1
5 M											0	0*

Note: As far as possible, the setting should be made so that the error is no more than 1%.

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

The BRR setting is found from the following formulas.

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

n	Clock
0	ø
1	ø/4
2	ø/16
3	ø/64

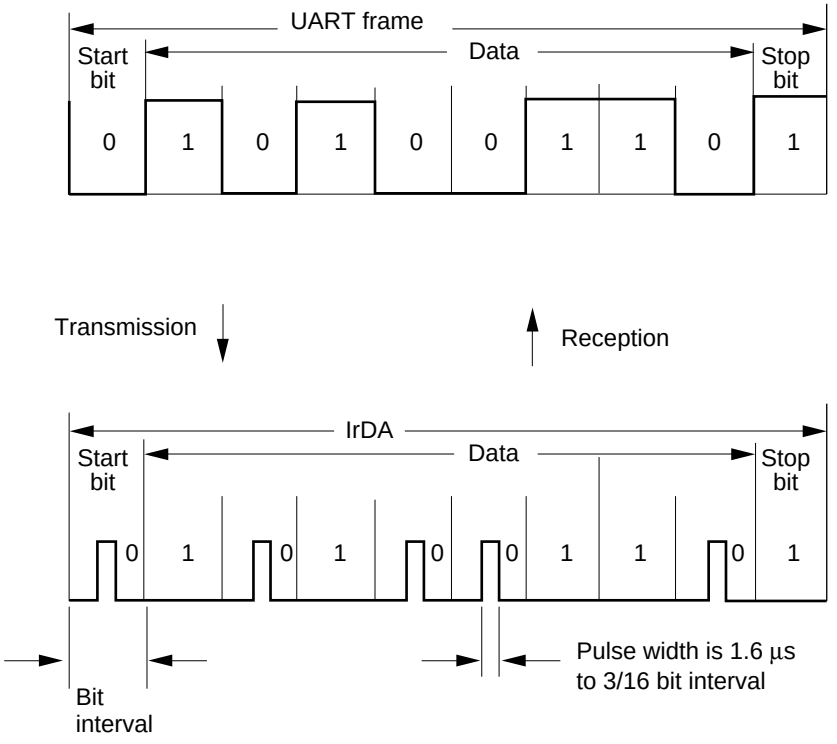
IrDA Operation: When the IrDA function is enabled with bit IrE in KBCOMP, the SCI channel 2 TxD2 and RxD2 signals are subjected to waveform encoding/decoding conforming to IrDA specification version 1.0 (IrTxD and IrRxD pins). By connecting these pins to an infrared transceiver/receiver, it is possible to implement infrared transmission/reception conforming to the IrDA specification version 1.0 system.

- Transmission

  In transmission, the output signal (UART frame) from the SCI is converted to an IR frame by the IrDA interface.
- Reception

  In reception, IR frame data is converted to a UART frame by the IrDA interface, and input to the SCI.

- IrDA Transmit/Receive Operations



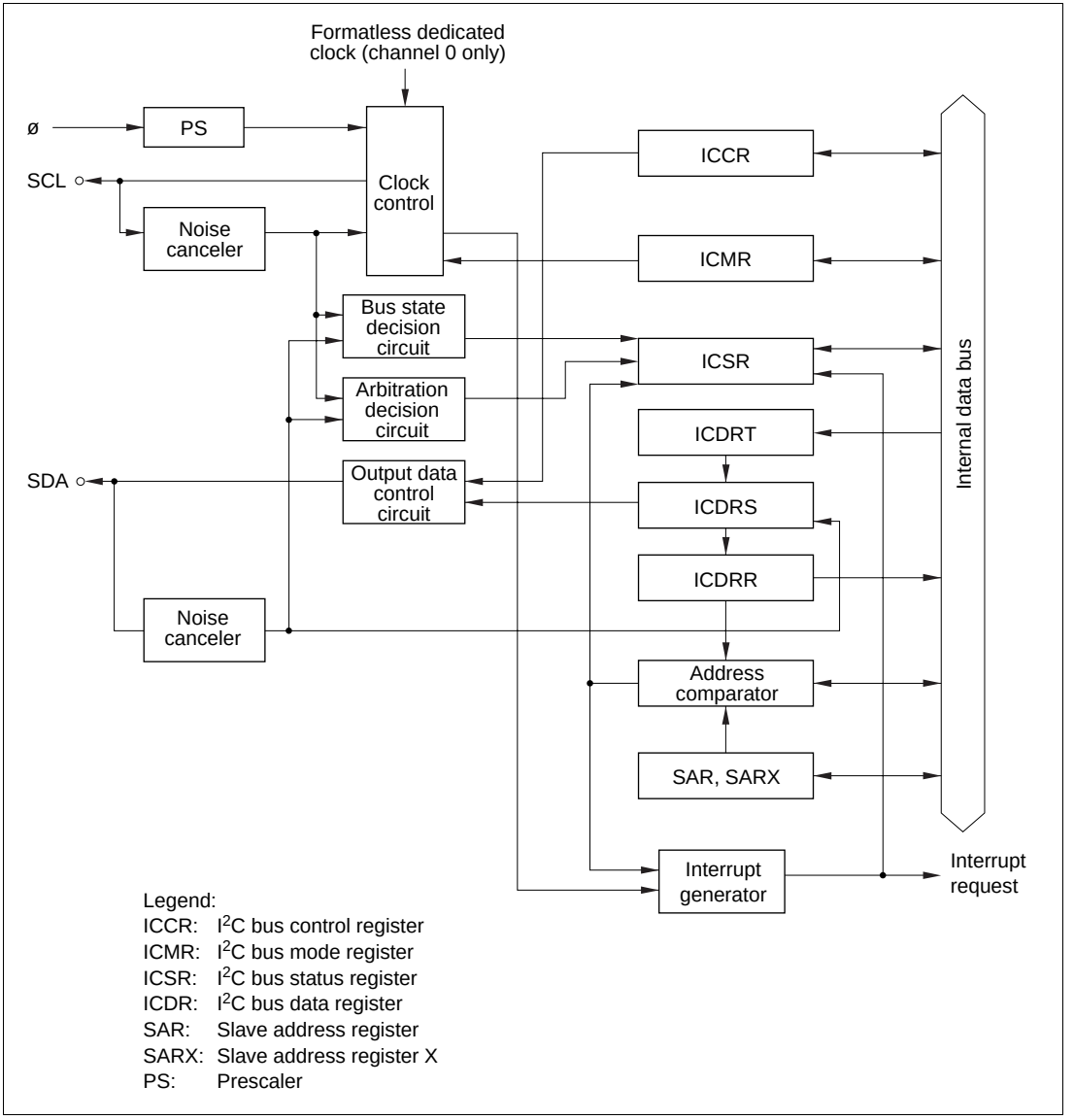
4.10 I^2C Bus Interface [Option] (Available for the H8S/2148 Series and H8S/2138 Series)

These series can be equipped with an optional two-channel serial interface conforming to the I^2C bus (Inter IC Bus) interface proposed by Philips. Data transfer using the I^2C bus interface requires only one data line (SDA) and one clock line (SCL), providing an economical solution in terms of connectors and printed circuit board area.

Features

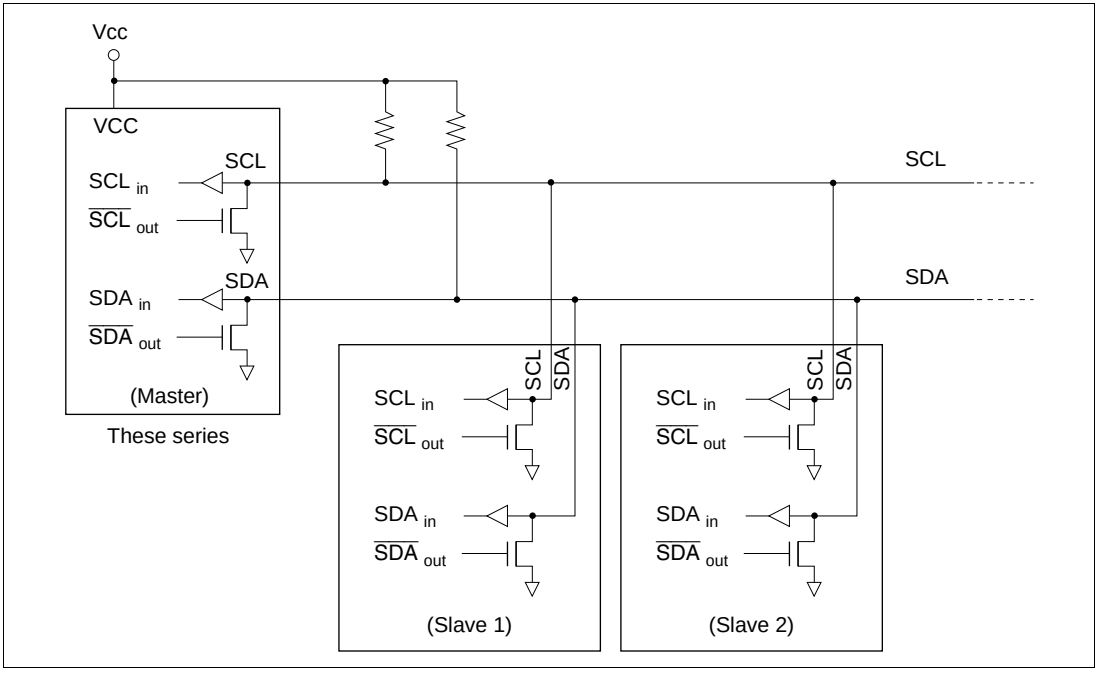
- Conforms to Philips I^2C bus interface
- Selection of addressing format or non-addressing format
- Two ways of setting slave address
- Start and stop conditions generated automatically
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Selection of 8 internal clocks (in master mode)
- Selection of acknowledgment mode or serial mode with no acknowledge bit
- Wait function
  - A wait can be inserted by driving the SCL pin low after data transfer excluding acknowledgment in acknowledge mode.
- Three interrupt sources
  - Data transfer end
  - Address match: when any slave address matches or the general call address is received in slave receive mode with the I^2C bus format
  - Stop condition detection
- Direct bus drive capability

Block Diagram



Block Diagram of I²C Bus Interface

I²C Bus Interface Connections (Example)



4.11 Keyboard Buffer Controller (Provided in the H8S/2148 Series)

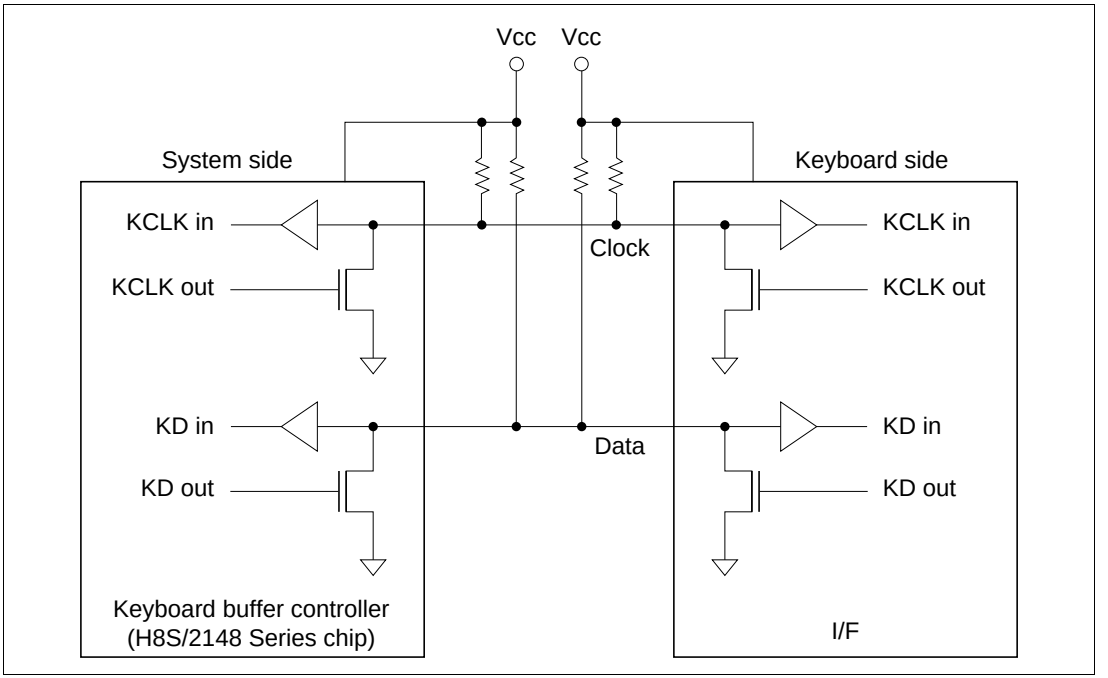
The H8S/2148 Series has three on-chip keyboard buffer controller channels, designated 0, 1, and 2. The keyboard buffer controller is provided with functions conforming to PS/2 interface specifications.

Data transfer using the keyboard buffer controller employs a data line (KD) and a clock line, providing economical use of connectors, board surface area, etc.

Features

- Conforms to PS/2 interface specifications
- Direct bus drive (via the KCLK and KD pins)
- Interrupt source: on completion of data reception
- Error detection: parity error and stop bit monitoring

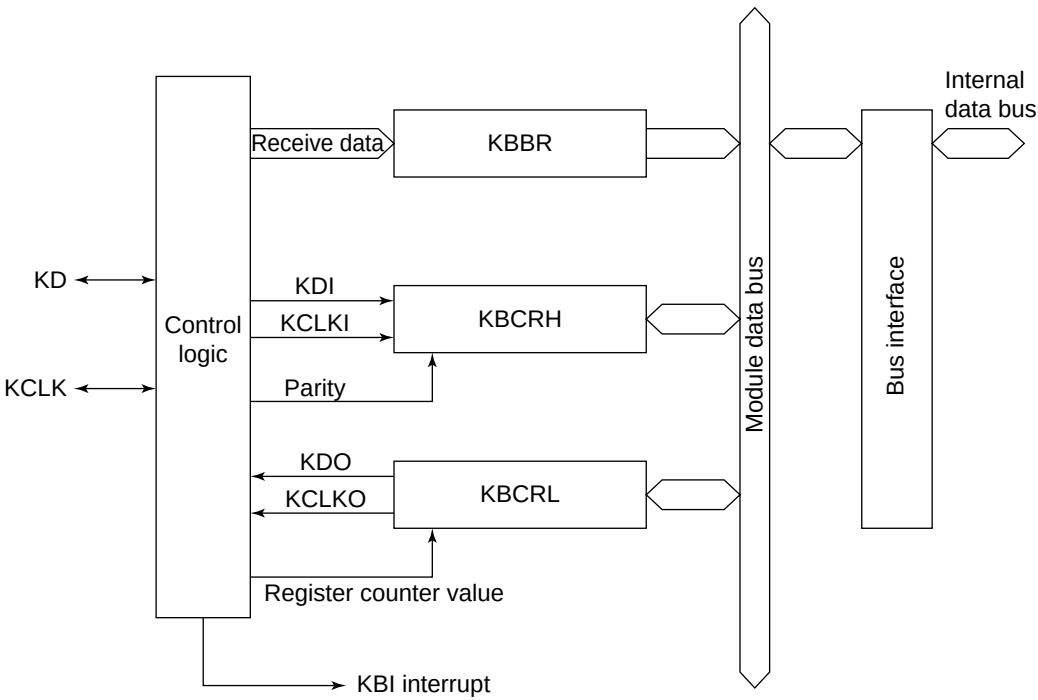
Keyboard Buffer Controller Connection



Pins Used for Each Function

Function	Pins
PS2 clock (KCLK)	PS2AC, PS2BC, PS2CC
PS2 data (KD)	PS2AD, PS2BD, PS2CD

Block Diagram



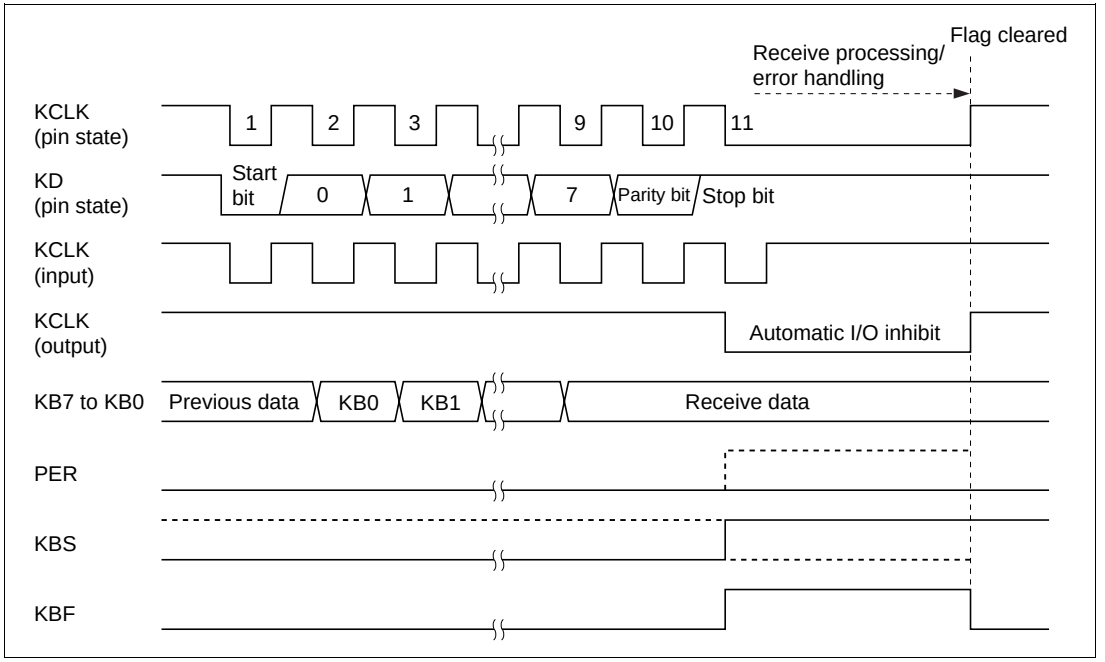
Legend:
KD: KBC data I/O pin
KCLK: KBC clock I/O pin
KBBR: Keyboard data buffer register
KBCRH: Keyboard control register H
KBCRL: Keyboard control register L

Operation

- Receive Operation

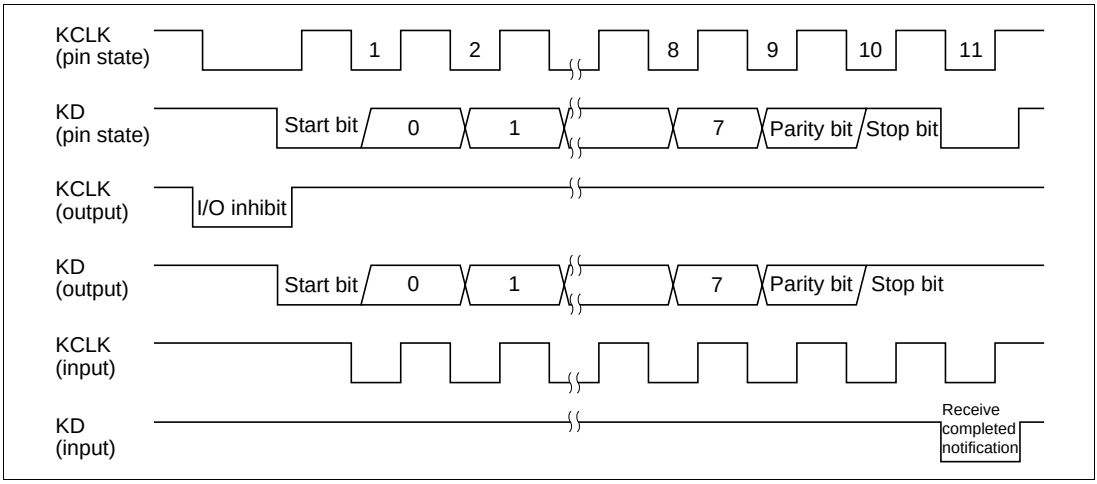
  In a receive operation, both KCLK (clock) and KD (data) are outputs on the keyboard side and inputs on the H8S/2148 Series chip (system) side. KD receives a start bit, 8 data bits (LSB-first), an odd parity bit, and a stop bit, in that order. The KD value is valid when KCLK is low.

- Receive Timing



- Transmit Operation

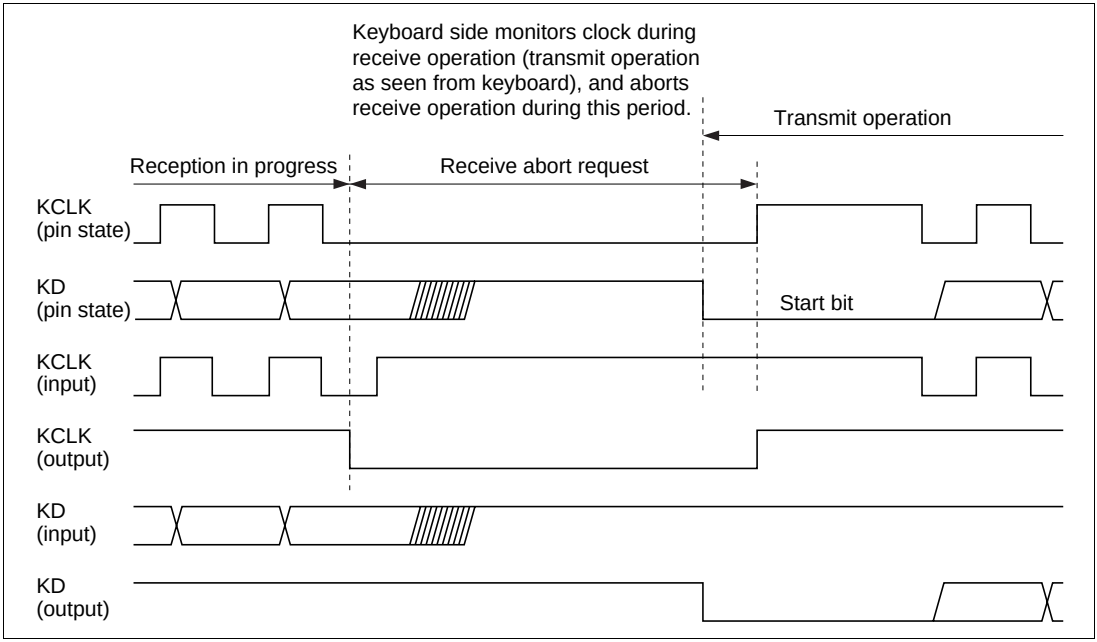
  In a transmit operation, KCLK (clock) is an output on the keyboard side, and KD (data) is an output on the H8S/2148 Series chip side. KD outputs a start bit, 8 data bits (LSB-first), an odd parity bit, and a stop bit, in that order. The KD value is valid when KCLK is high.



Transmit Timing

- Receive Abort

  The H8S/2148 Series device (system side) can forcibly abort transmission from the device connected to it (keyboard side) in the event of a protocol error, etc. In this case, the system holds the clock low.



Receive Abort and Transmit Start (Transmission/Reception Switchover) Timing

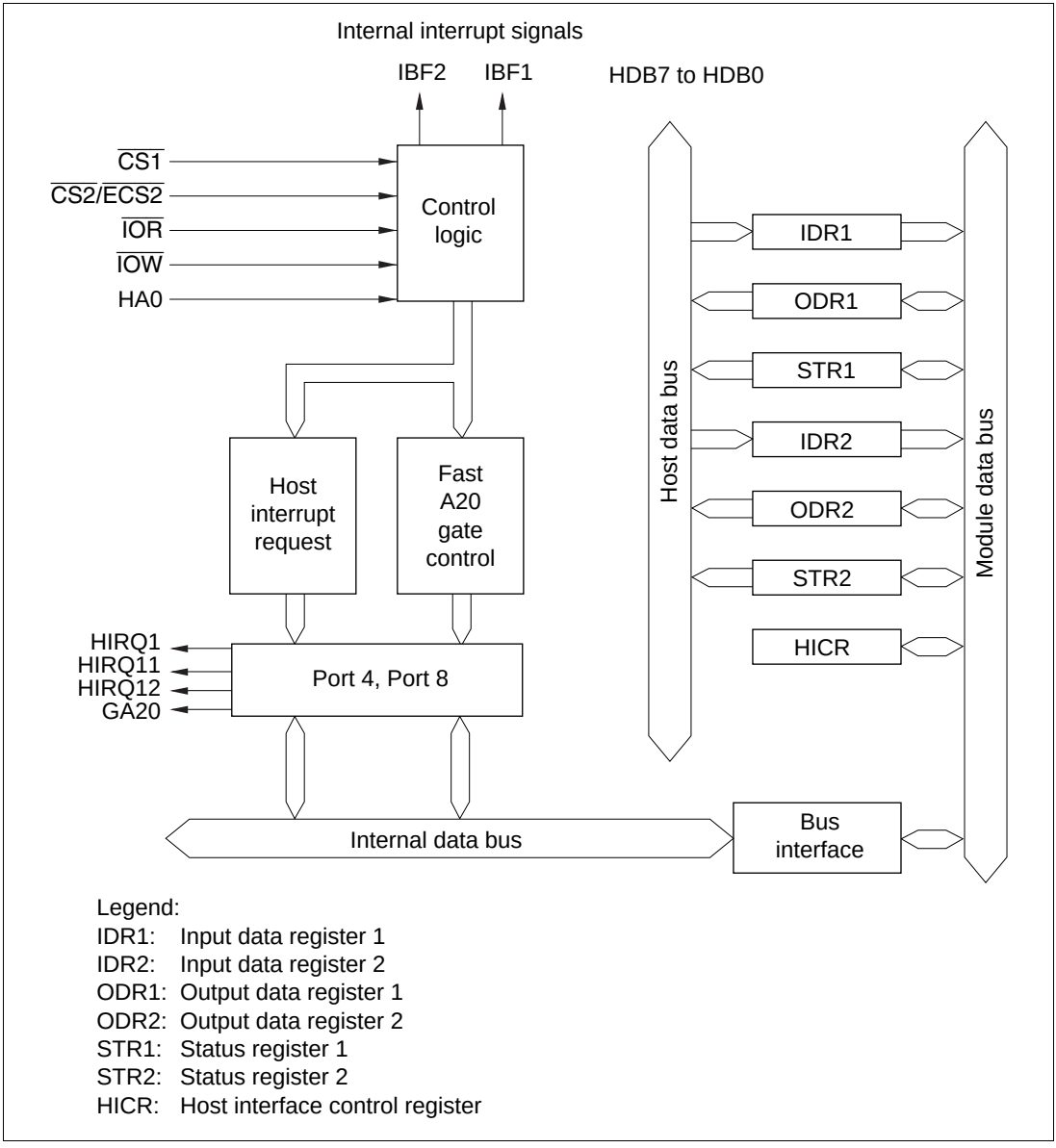
4.12 Host Interface (HIF) (Provided in the H8S/2148 Series and H8S/2138 Series)

These series are equipped with a host interface (HIF) that performs two-channel parallel communication with the master CPU. The HIF includes a 4-byte data register, 2-byte status register, and 1-byte control register, plus high-speed gate A20 logic and a host interrupt source circuit. The HIF registers can be accessed as registers in memory space by the internal CPU and master.

Features

- Use of two chip select signal inputs allows independent operation of the two host interface channels
- An interrupt request to the internal CPU can be generated by a host write to the input data register
- An interrupt request to the master can be output by an internal CPU write to the output data register (also using software)
- A high-speed GATE A20 signal can be output (turn-on/turn-off sequence) by a host write of a specific data sequence to the input data register

Block Diagram



Legend:
IDR1: Input data register 1
IDR2: Input data register 2
ODR1: Output data register 1
ODR2: Output data register 2
STR1: Status register 1
STR2: Status register 2
HICR: Host interface control register

Block Diagram of Host Interface

4.13 D/A Converter

These series have an on-chip D/A converter with 8-bit precision. Analog signals can be output on up to two channels by the program.

Features

- Eight-bit resolution
- Two-channel output
- Maximum conversion time: 10 μs (with 20-pF load capacitance)
- Output voltage: 0 V to AV_{ref} (H8S/2148 Series and H8S/2144 Series)
  0 V to AV_{CC} (H8S/2138 Series and H8S/2134 Series)
- D/A output retention in software standby mode.

Operation

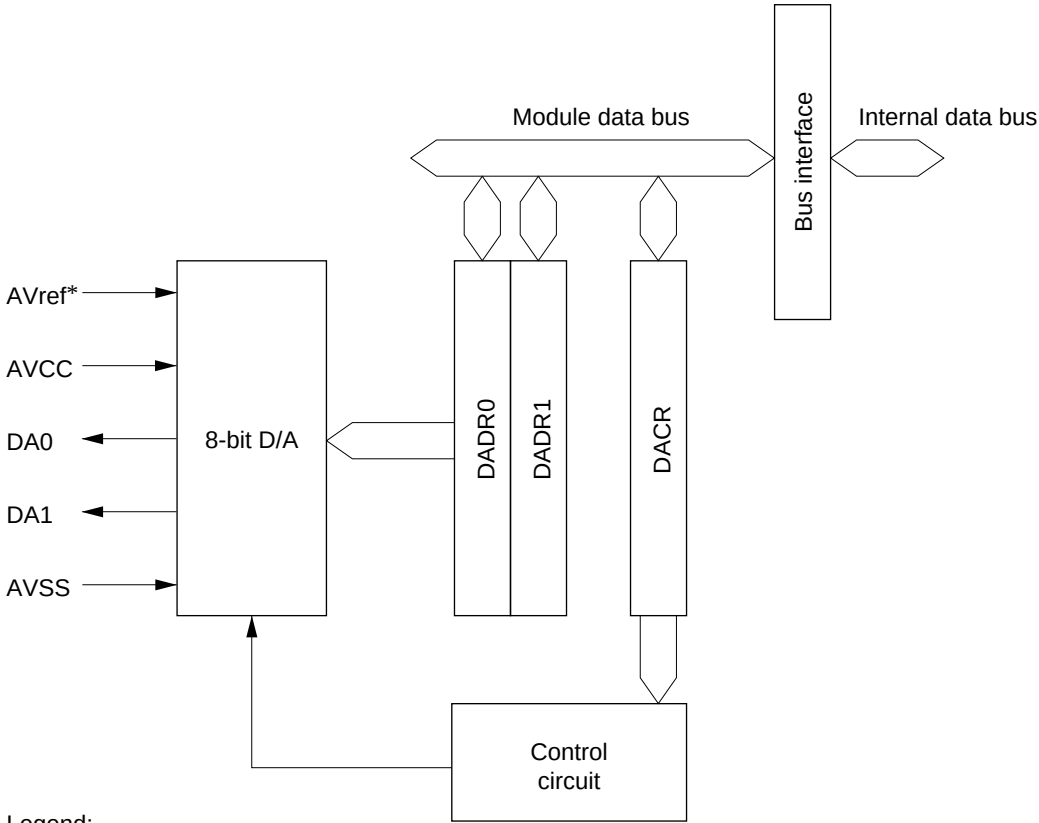
D/A converter operation is enabled by setting the D/A output enable bit to 1. While this bit is set to 1, DADR contents are constantly converted and output to the corresponding pin.

The output value is:

$$\frac{\text{DADR contents}}{256} \times \text{AVref} \qquad \text{(H8S/2148 Series and H8S/2144 Series)}$$

$$\frac{\text{DADR contents}}{256} \times \text{AVcc} \qquad \text{(H8S/2138 Series and H8S/2134 Series)}$$

Block Diagram



Legend:
DACR: D/A control register
DADR0: D/A data register 0
DADR1: D/A data register 1

Note: * The H8S/2138 Series and H8S/2134 Series do not have AVref pin.

Block Diagram of D/A Converter

4.14 A/D Converter

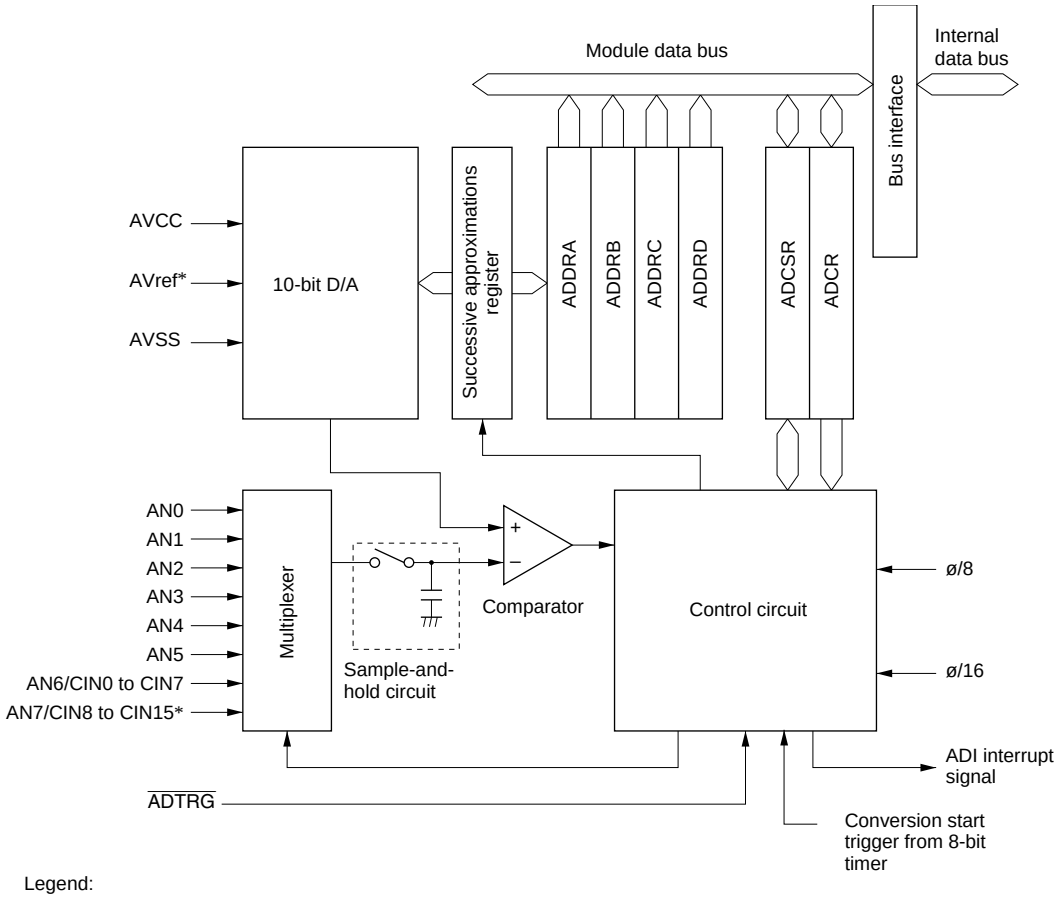
These series incorporate a successive approximation type 10-bit A/D converter that allows up to eight analog input channels to be selected.

In addition to the eight analog input channels, up to 16 channels of digital input can be selected for A/D conversion. Since the conversion precision falls to the equivalent of 6-bit resolution when digital input is selected, digital input is ideal for use by a comparator identifying multi-valued inputs, for example.

Features

- 10-bit resolution
  Input channels: 8 (analog input)
  16 (digital input: H8S/2148 Series and H8S/2144 Series)
  8 (digital input: H8S/2138 Series and H8S/2134 Series)
- Settable analog conversion voltage range
  — Conversion of analog voltages from 0 V to AV_{ref}, with the reference voltage pin (AVref) as the analog reference voltage (H8S/2148 Series and H8S/2144 Series only)
- High-speed conversion
  — 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
- Three kinds of conversion start
  — Selection of software or timer conversion start trigger (8-bit timer), or $\overline{\text{ADTRG}}$ pin
- Four data registers
  — Conversion results held in a data register for each channel
- Sample and hold function
- A/D conversion end interrupt
  — An A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion

Block Diagram



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

Note: * The H8S/2138 Series and H8S/2134 Series do not have signals CIN8 to CIN15 and AVref pin.

Block Diagram of A/D Converter

Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode.

- Single Mode

  Single mode is selected when A/D conversion is to be performed on a single channel only.

  A/D conversion is started when the ADST bit is set to 1, according to the specified conversion start condition.

  On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.

- Scan Mode

  Scan mode is selected when A/D conversion is to be performed repeatedly on a number of channels.

  Once the ADST bit is set to 1 according to the specified conversion start condition, A/D conversion is performed repeatedly on the selected channels until the ADST bit is cleared to 0 by software.

  An ADI interrupt request can be generated on completion of the first conversion operation for all the selected channels.

4.15 I/O Ports

The H8S/2148 Series and H8S/2144 Series have ten I/O ports (ports 1 to 6, 8, 9, A, and B), and one input-only port (port 7).

The H8S/2138 Series and H8S/2134 Series have eight I/O ports (ports 1 to 6, 8, 9) and one input-only port (port 7).

Each port includes a data direction register (DDR) that controls input/output and data registers (DR, ODR) that store output data.

Ports 1 to 3, 6, A*, and B* have a built-in MOS input pull-up function. For ports A and B, the on/off status of the MOS input pull-up is controlled by DDR and ODR. Ports 1 to 3 and 6 have a MOS input pull-up control register (PCR), in addition to DDR and DR, to control the on/off status of the MOS input pull-up.

Note: * Ports A and B apply to the H8S/2148 Series and H8S/2144 Series

H8S/2148 Series Port Functions in Each Operating Mode

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 1	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P17 to P10/ A7 to A0/ PW7 to PW0	Lower address output (A7 to A0)	When DDR = 0 (after reset): input port When DDR = 1: lower address output (A7 to A0) or PWM timer output (PW7 to PW0)	I/O port also functioning as PWM timer output (PW7 to PW0)
Port 2	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P27/A15/PW15/ CBLANK P26/A14/PW14 P25/A13/PW13 P24/A12/PW12 P23/A11/PW11 P22/A10/PW10 P21/A9/PW9 P20/A8/PW8	Upper address output (A15 to A8)	When DDR = 0 (after reset): input port or timer connection output (CBLANK) When DDR = 1: upper address output (A15 to A8), PWM timer output (PW15 to PW8), timer connection output (CBLANK), or output port (P27 to P24)	I/O port also functioning as PWM timer output (PW15 to PW8) and timer connection output (CBLANK)

H8S/2148 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 3	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P37 to P30/ HDB7 to HDB0/ D15 to D8	Data bus input/output (D15 to D8)		I/O port also functioning as host interface data bus input/output (HDB7 to HDB0)
Port 4	• 8-bit I/O port	P47/PWX1 P46/PWX0 P45/TMRI1/ HIRQ12/CSYNCI P44/TMO1/ HIRQ1/HSYNCO P43/TMCI1/ HIRQ11/HSYNCI P42/TMRI0/ SCK2/SDA1 P41/TMO0/ RxD2/IrRxD P40/TMCI0/ TxD2/IrTxD	I/O port also functioning as 14-bit PWM timer output (PWX1, PWX0), 8-bit timer 0 and 1 input/output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), timer connection input/output (HSYNCO, CSYNCI, HSYNCI), SCI2 input/output (TxD2, RxD2, SCK2), IrDA interface input/output (IrTxD, IrRxD), and I^2C bus interface 1 (option) input/output (SDA1)		I/O port also functioning as 14-bit PWM timer output (PWX1, PWX0), 8-bit timer 0 and 1 input/ output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), timer connection input/output (HSYNCO, CSYNCI, HSYNCI), host interface host CPU interrupt request output (HIRQ12, HIRQ1, HIRQ11), SCI2 input/ output (TxD2, RxD2, SCK2), IrDA interface input/output (IrTxD, IrRxD), and I^2C bus interface 1 (option) input/output (SDA1)
Port 5	• 3-bit I/O port	P52/SCK0/SCL0 P51/RxD0 P50/TxD0	I/O port also functioning as SCI0 input/output (TxD0, RxD0, SCK0) and I^2C bus interface 0 (option) input/output (SCL0)		

H8S/2148 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 6	• 8-bit I/O port	P67/$\overline{\text{IRQ7}}$/TMOX/$\overline{\text{KIN7}}$/CIN7 P66/$\overline{\text{IRQ6}}$/FTOB/$\overline{\text{KIN6}}$/CIN6 P65/FTID/$\overline{\text{KIN5}}$/CIN5 P64/FTIC/$\overline{\text{KIN4}}$/CIN4/CLAMPO P63/FTIB/$\overline{\text{KIN3}}$/CIN3/VFBACKI P62/FTIA/TMIY/$\overline{\text{KIN2}}$/CIN2/VSYNCI P61/FTOA/$\overline{\text{KIN1}}$/CIN1/VSYNCO P60/FTCI/TMIX/$\overline{\text{KIN0}}$/CIN0/HFBACKI	I/O port also functioning as external interrupt input ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$), FRT input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer X and Y input (TMIX, TMIY), timer connection input/output (CLAMPO, VFBACKI, VSYNCI, VSYNCO, HFBACKI), key-sense interrupt input ($\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$), and comparator input (CIN7 to CIN0)		
Port 7	• 8-bit input port	P77/AN7/DA1 P76/AN6/DA0 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Input port also functioning as A/D converter analog input (AN7 to AN0) and D/A converter analog output (DA1, DA0)		

H8S/2148 Series Port Functions in Each Operating Mode (cont)

Port	Description	Pins	Expanded Modes: Mode 1	Expanded Modes: Mode 2, Mode 3 (EXPE = 1)	Single-Chip Mode: Mode 2, Mode 3 (EXPE = 0)
Port 8	• 7-bit I/O port	P86/$\overline{IRQ5}$/SCK1/SCL1 P85/$\overline{IRQ4}$/RxD1 P84/$\overline{IRQ3}$/TxD1 P83 P82 P81/$\overline{CS2}$/GA20 P80/HA0	I/O port also functioning as external interrupt input ($\overline{IRQ5}$, $\overline{IRQ4}$, $\overline{IRQ3}$), SCI1 input/output (TxD1, RxD1, SCK1), and I^2C bus interface 1 (option) input/output (SCL1)		I/O port also functioning as external interrupt input ($\overline{IRQ5}$, $\overline{IRQ4}$, $\overline{IRQ3}$), SCI1 input/output (TxD1, RxD1, SCK1), host interface control input/output ($\overline{CS2}$, GA20, HA0), and I^2C bus interface 1 (option) input/output (SCL1)
Port 9	• 8-bit I/O port	P97/$\overline{WAIT}$/SDA0	I/O port also functioning as expanded data bus control input ($\overline{WAIT}$) and I^2C bus interface 0 (option) input/output (SDA0)		I/O port also functioning as I^2C bus interface 0 (option) input/output (SDA0)
		P96/ø/EXCL	When DDR = 0: input port or EXCL input When DDR = 1 (after reset): ø output	When DDR = 0 (after reset) input port or EXCL input When DDR = 1: ø output	
		P95/$\overline{AS}$/$\overline{IOS}$/$\overline{CS1}$ P94/$\overline{HWR}$/$\overline{IOW}$ P93/$\overline{RD}$/$\overline{IOR}$	Expanded data bus control output ($\overline{AS}$/$\overline{IOS}$, $\overline{HWR}$, $\overline{RD}$)		I/O port also functioning as host interface control input ($\overline{CS1}$, $\overline{IOW}$, $\overline{IOR}$)
		P92/$\overline{IRQ0}$ P91/$\overline{IRQ1}$	I/O port also functioning as external interrupt input ($\overline{IRQ0}$, $\overline{IRQ1}$)		
		P90/$\overline{LWR}$/$\overline{IRQ2}$/$\overline{ADTRG}$/$\overline{ECS2}$	I/O port also functioning as expanded data bus control output ($\overline{LWR}$), external interrupt input ($\overline{IRQ2}$), and A/D converter external trigger input ($\overline{ADTRG}$)		I/O port also functioning as external interrupt input ($\overline{IRQ2}$), A/D converter external trigger input ($\overline{ADTRG}$), and host interface control input ($\overline{ECS2}$)

H8S/2148 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port A	• 8-bit I/O port	PA7/A23/$\overline{\text{KIN15}}$/CIN15/PS2CD PA6/A22/$\overline{\text{KIN14}}$/CIN14/PS2CC PA5/A21/$\overline{\text{KIN13}}$/CIN13/PS2BD PA4/A20/$\overline{\text{KIN12}}$/CIN12/PS2BC PA3/A19/$\overline{\text{KIN11}}$/CIN11/PS2AD PA2/A18/$\overline{\text{KIN10}}$/CIN10/PS2AC PA1/A17/$\overline{\text{KIN9}}$/CIN9 PA0/A16/$\overline{\text{KIN8}}$/CIN8	I/O port also functioning as key-sense interrupt input ($\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$), comparator input (CIN15 to CIN8), and keyboard buffer controller input/output (PS2CD, PS2CC, PS2BD, PS2BC, PS2AD, PS2AC)	I/O port also functioning as address output (A23 to A16), key-sense interrupt input ($\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$), comparator input (CIN15 to CIN8), and keyboard buffer controller input/output (PS2CD, PS2CC, PS2BD, PS2BC, PS2AD, PS2AC)	I/O port also functioning as key-sense interrupt input ($\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$), comparator input (CIN15 to CIN8), and keyboard buffer controller input/output (PS2CD, PS2CC, PS2BD, PS2BC, PS2AD, PS2AC)
Port B	• 8-bit I/O port	PB7 to PB0/D7 to D0	In 8-bit bus mode (ABW = 1): I/O port In 16-bit bus mode (ABW = 0): Data bus input/output (D7 to D0)		I/O port

H8S/2144 Series Port Functions in Each Operating Mode

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 1	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P17 to P10/ A7 to A0	Lower address output (A7 to A0)	When DDR = 0 (after reset): input port When DDR = 1: lower address output (A7 to A0)	I/O port
Port 2	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P27 to P20/ A15 to A8	Upper address output (A15 to A8)	When DDR = 0 (after reset): input port When DDR = 1: upper address output (A15 to A8) or output port (P27 to P24)	I/O port
Port 3	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P37 to P30/ D15 to D8	Data bus input/output (D15 to D8)		I/O port
Port 4	• 8-bit I/O port	P47/PWX1 P46/PWX0 P45/TMRI1 P44/TMO1 P43/TMCI1 P42/TMRI0/SCK2 P41/TMO0/RxD2/ IrRxD P40/TMCI0/ TxD2/IrTxD	I/O port also functioning as 14-bit PWM timer output (PWX1, PWX0), 8-bit timer 0 and 1 input/output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), SCI2 input/output (TxD2, RxD2, SCK2), and IrDA interface input/output (IrTxD, IrRxD)		
Port 5	• 3-bit I/O port	P52/SCK0 P51/RxD0 P50/TxD0	I/O port also functioning as SCI0 input/output (TxD0, RxD0, SCK0)		

H8S/2144 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 6	• 8-bit I/O port	P67/$\overline{\text{IRQ7}}$/$\overline{\text{KIN7}}$/CIN7 P66/$\overline{\text{IRQ6}}$/FTOB/$\overline{\text{KIN6}}$/CIN6 P65/FTID/$\overline{\text{KIN5}}$/CIN5 P64/FTIC/$\overline{\text{KIN4}}$/CIN4 P63/FTIB/$\overline{\text{KIN3}}$/CIN3 P62/FTIA/TMIY/$\overline{\text{KIN2}}$/CIN2 P61/FTOA/$\overline{\text{KIN1}}$/CIN1 P60/FTCI/$\overline{\text{KIN0}}$/CIN0	I/O port also functioning as external interrupt input ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$), FRT input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer Y input (TMIY), key-sense interrupt input ($\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$), and comparator input (CIN7 to CIN0)		
Port 7	• 8-bit input port	P77/AN7/DA1 P76/AN6/DA0 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Input port also functioning as A/D converter analog input (AN7 to AN0) and D/A converter analog output (DA1, DA0)		
Port 8	• 7-bit I/O port	P86/$\overline{\text{IRQ5}}$/SCK1 P85/$\overline{\text{IRQ4}}$/RxD1 P84/$\overline{\text{IRQ3}}$/TxD1 P83 P82 P81 P80	I/O port also functioning as external interrupt input ($\overline{\text{IRQ5}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$) and SCI1 input/ output (TxD1, RxD1, SCK1)		

H8S/2144 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 9	• 8-bit I/O port	P97/$\overline{\text{WAIT}}$	I/O port also functioning as expanded data bus control input ($\overline{\text{WAIT}}$)		I/O port
		P96/ø/EXCL	When DDR = 0: input port or EXCL input When DDR = 1 (after reset): ø output	When DDR = 0 (after reset) input port or EXCL input When DDR = 1: ø output	
		P95/$\overline{\text{AS}}$/$\overline{\text{IOS}}$ P94/$\overline{\text{HWR}}$ P93/$\overline{\text{RD}}$	Expanded data bus control output($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$)		I/O port
		P92/$\overline{\text{IRQ0}}$ P91/$\overline{\text{IRQ1}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$)		
		P90/$\overline{\text{LWR}}$/ $\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	I/O port also functioning as expanded data bus control output($\overline{\text{LWR}}$), external interrupt input ($\overline{\text{IRQ2}}$), and A/D converter external trigger input ($\overline{\text{ADTRG}}$)		I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$) and A/D converter external trigger input ($\overline{\text{ADTRG}}$)
Port A	• 8-bit I/O port	PA7 to PA0/ A23 to A16/ $\overline{\text{KIN15}}$ to $\overline{\text{KIN5}}$/ CIN15 to CIN8	I/O port also functioning as key-sense interrupt input ($\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$), and comparator input (CIN15 to CIN8)	I/O port also functioning as address output (A23 to A16), key-sense interrupt input ($\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$), and comparator input (CIN15 to CIN8)	I/O port also functioning as key-sense interrupt input ($\overline{\text{KIN15}}$ to $\overline{\text{KIN8}}$) and comparator input (CIN15 to CIN8)
Port B	• 8-bit I/O port	PB7 to PB0/ D7 to D0	In 8-bit bus mode (ABW = 1): I/O port In 16-bit bus mode (ABW = 0): Data bus input/output (D7 to D0)		I/O port

H8S/2138 Series Port Functions in Each Operating Mode

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 1	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P17 to P10/ A7 to A0/ PW7 to PW0	Lower address output (A7 to A0)	When DDR = 0 (after reset): input port When DDR = 1: lower address output (A7 to A0) or PWM timer output (PW7 to PW0)	I/O port also functioning as PWM timer output (PW7 to PW0)
Port 2	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P27/A15/PW15/ CBLANK P26/A14/PW14 P25/A13/PW13 P24/A12/PW12 P23/A11/PW11 P22/A10/PW10 P21/A9/PW9 P20/A8/PW8	Upper address output (A15 to A8)	When DDR = 0 (after reset): input port or timer connection output (CBLANK) When DDR = 1: upper address output (A15 to A8), PWM timer output (PW15 to PW8), timer connection output (CBLANK), or output port (P27 to P24)	I/O port also functioning as PWM timer output (PW15 to PW8) and timer connection output (CBLANK)

H8S/2138 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 3	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P37 to P30/ HDB7 to HDB0/ D15 to D8	Data bus input/output (D15 to D8)		I/O port also functioning as host interface data bus input/output (HDB7 to HDB0)
Port 4	• 8-bit I/O port	P47/PWX1 P46/PWX0 P45/TMRI1/ HIRQ12/CSYNCI P44/TMO1/ HIRQ1/HSYNCO P43/TMCI1/ HIRQ11/HSYNCI P42/TMRI0/ SCK2/SDA1 P41/TMO0/ RxD2/IrRxD P40/TMCI0/ TxD2/IrTxD	I/O port also functioning as 14-bit PWM timer output (PWX1, PWX0), 8-bit timer 0 and 1 input/output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), timer connection input/output (HSYNCO, CSYNCI, HSYNCI), SCI2 input/output (TxD2, RxD2, SCK2), IrDA interface input/output (IrTxD, IrRxD), and I^2C bus interface 1 (option) input/output (SDA1)		I/O port also functioning as 14-bit PWM timer output (PWX1, PWX0), 8-bit timer 0 and 1 input/ output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), timer connection input/output (HSYNCO, CSYNCI, HSYNCI), host interface host CPU interrupt request output (HIRQ12, HIRQ1, HIRQ11), SCI2 input/ output (TxD2, RxD2, SCK2), IrDA interface input/output (IrTxD, IrRxD), and I^2C bus interface 1 (option) input/output (SDA1)
Port 5	• 3-bit I/O port	P52/SCK0/SCL0 P51/RxD0 P50/TxD0	I/O port also functioning as SCI0 input/output (TxD0, RxD0, SCK0) and I^2C bus interface 0 (option) input/output (SCL0)		

H8S/2138 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 6	• 8-bit I/O port	P67/$\overline{\text{IRQ7}}$/TMOX/$\overline{\text{KIN7}}$/CIN7 P66/$\overline{\text{IRQ6}}$/FTOB/$\overline{\text{KIN6}}$/CIN6 P65/FTID/$\overline{\text{KIN5}}$/CIN5 P64/FTIC/$\overline{\text{KIN4}}$/CIN4/CLAMPO P63/FTIB/$\overline{\text{KIN3}}$/CIN3/VFBACKI P62/FTIA/TMIY/$\overline{\text{KIN2}}$/CIN2/VSYNCI P61/FTOA/$\overline{\text{KIN1}}$/CIN1/VSYNCO P60/FTCI/TMIX/$\overline{\text{KIN0}}$/CIN0/HFBACKI	I/O port also functioning as external interrupt input ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$), FRT input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer X and Y input (TMIX, TMIY), timer connection input/output (CLAMPO, VFBACKI, VSYNCI, VSYNCO, HFBACKI), key-sense interrupt input ($\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$), and comparator input (CIN7 to CIN0)		
Port 7	• 8-bit input port	P77/AN7/DA1 P76/AN6/DA0 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Input port also functioning as A/D converter analog input (AN7 to AN0) and D/A converter analog output (DA1, DA0)		

H8S/2138 Series Port Functions in Each Operating Mode (cont)

Port	Description	Pins	Expanded Modes: Mode 1	Expanded Modes: Mode 2, Mode 3 (EXPE = 1)	Single-Chip Mode: Mode 2, Mode 3 (EXPE = 0)
Port 8	• 7-bit I/O port	P86/$\overline{\text{IRQ5}}$/SCK1/SCL1 P85/$\overline{\text{IRQ4}}$/RxD1 P84/$\overline{\text{IRQ3}}$/TxD1 P83 P82 P81/$\overline{\text{CS2}}$/GA20 P80/HA0	I/O port also functioning as external interrupt input ($\overline{\text{IRQ5}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$), SCI1 input/output (TxD1, RxD1, SCK1), and I^2C bus interface 1 (option) input/output (SCL1)		I/O port also functioning as external interrupt input ($\overline{\text{IRQ5}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$), SCI1 input/output (TxD1, RxD1, SCK1), host interface control input/output ($\overline{\text{CS2}}$, GA20, HA0), and I^2C bus interface 1 (option) input/output (SCL1)
Port 9	• 8-bit I/O port	P97/$\overline{\text{WAIT}}$/SDA0	I/O port also functioning as expanded data bus control input ($\overline{\text{WAIT}}$) and I^2C bus interface 0 (option) input/output (SDA0)		I/O port also functioning as I^2C bus interface 0 (option) input/output (SDA0)
		P96/ø/EXCL	When DDR = 0: input port or EXCL input When DDR = 1 (after reset): ø output	When DDR = 0 (after reset) input port or EXCL input When DDR = 1: ø output	
		P95/$\overline{\text{AS}}$/$\overline{\text{IOS}}$/$\overline{\text{CS1}}$ P94/$\overline{\text{HWR}}$/$\overline{\text{IOW}}$ P93/$\overline{\text{RD}}$/$\overline{\text{IOR}}$	Expanded data bus control output ($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$)		I/O port also functioning as host interface control input ($\overline{\text{CS1}}$, $\overline{\text{IOW}}$, $\overline{\text{IOR}}$)
		P92/$\overline{\text{IRQ0}}$ P91/$\overline{\text{IRQ1}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$)		
		P90/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$/$\overline{\text{ECS2}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$), and A/D converter external trigger input ($\overline{\text{ADTRG}}$)		I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$), A/D converter external trigger input ($\overline{\text{ADTRG}}$), and host interface control input ($\overline{\text{ECS2}}$)

H8S/2134 Series Port Functions in Each Operating Mode

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 1	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P17 to P10/ A7 to A0	Lower address output (A7 to A0)	When DDR = 0 (after reset): input port When DDR = 1: lower address output (A7 to A0)	I/O port
Port 2	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P27 to P20/ A15 to A8	Upper address output (A15 to A8)	When DDR = 0 (after reset): input port When DDR = 1: upper address output (A15 to A8) or output port (P27 to P24)	I/O port
Port 3	• 8-bit I/O port • Built-in MOS input pull-up • LED drive capability	P37 to P30/ D15 to D8	Data bus input/output (D15 to D8)		I/O port
Port 4	• 8-bit I/O port	P47/PWX1 P46/PWX0 P45/TMRI1 P44/TMO1 P43/TMCI1 P42/TMRI0/SCK2 P41/TMO0/RxD2/ IrRxD P40/TMCI0/ TxD2/IrTxD	I/O port also functioning as 14-bit PWM timer output (PWX1, PWX0), 8-bit timer 0 and 1 input/output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), SCI2 input/output (TxD2, RxD2, SCK2), and IrDA interface input/output (IrTxD, IrRxD)		
Port 5	• 3-bit I/O port	P52/SCK0 P51/RxD0 P50/TxD0	I/O port also functioning as SCI0 input/output (TxD0, RxD0, SCK0)		

H8S/2134 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 6	• 8-bit I/O port	P67/$\overline{\text{IRQ7}}$/$\overline{\text{KIN7}}$/CIN7 P66/$\overline{\text{IRQ6}}$/FTOB/$\overline{\text{KIN6}}$/CIN6 P65/FTID/$\overline{\text{KIN5}}$/CIN5 P64/FTIC/$\overline{\text{KIN4}}$/CIN4 P63/FTIB/$\overline{\text{KIN3}}$/CIN3 P62/FTIA/TMIY/$\overline{\text{KIN2}}$/CIN2 P61/FTOA/$\overline{\text{KIN1}}$/CIN1 P60/FTCI/$\overline{\text{KIN0}}$/CIN0	I/O port also functioning as external interrupt input ($\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$), FRT input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer Y input (TMIY), key-sense interrupt input ($\overline{\text{KIN7}}$ to $\overline{\text{KIN0}}$), and comparator input (CIN7 to CIN0)		
Port 7	• 8-bit input port	P77/AN7/DA1 P76/AN6/DA0 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Input port also functioning as A/D converter analog input (AN7 to AN0) and D/A converter analog output (DA1, DA0)		
Port 8	• 7-bit I/O port	P86/$\overline{\text{IRQ5}}$/SCK1 P85/$\overline{\text{IRQ4}}$/RxD1 P84/$\overline{\text{IRQ3}}$/TxD1 P83 P82 P81 P80	I/O port also functioning as external interrupt input ($\overline{\text{IRQ5}}$, $\overline{\text{IRQ4}}$, $\overline{\text{IRQ3}}$) and SCI1 input/ output (TxD1, RxD1, SCK1)		

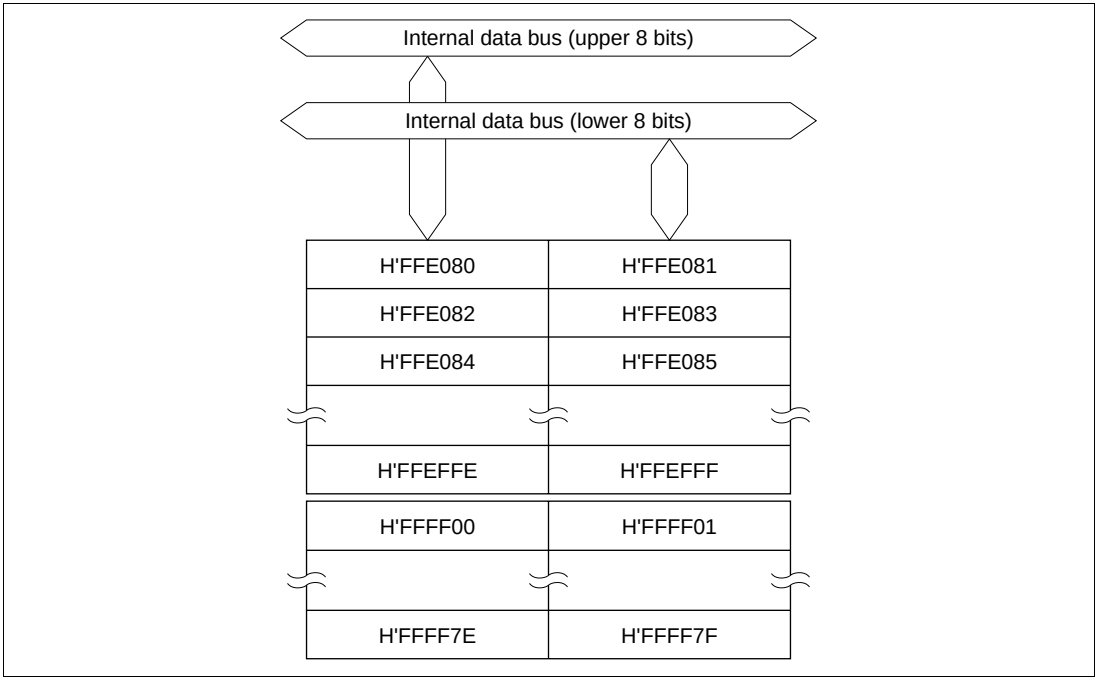
H8S/2134 Series Port Functions in Each Operating Mode (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 9	• 8-bit I/O port	P97/$\overline{\text{WAIT}}$	I/O port also functioning as expanded data bus control input ($\overline{\text{WAIT}}$)		I/O port
		P96/ø/EXCL	When DDR = 0: input port or EXCL input When DDR = 1 (after reset): ø output	When DDR = 0 (after reset) input port or EXCL input When DDR = 1: ø output	
		P95/$\overline{\text{AS}}$/$\overline{\text{IOS}}$ P94/$\overline{\text{HWR}}$ P93/$\overline{\text{RD}}$	Expanded data bus control output($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{HWR}}$, $\overline{\text{RD}}$)		I/O port
		P92/$\overline{\text{IRQ0}}$ P91/$\overline{\text{IRQ1}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$)		
		P90/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$), and A/D converter external trigger input ($\overline{\text{ADTRG}}$)		

4.16 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).



Block Diagram of RAM
(Example of H8S/2148, H8S/2144, H8S/2143, H8S/2138, H8S/2134, H8S/2133)

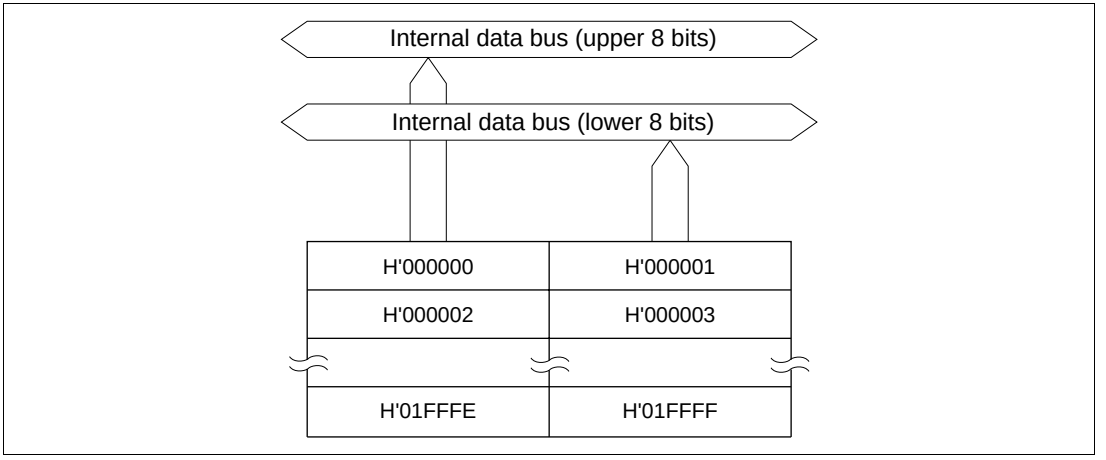
- RAM Size

	Product Name			
RAM (Bytes)	**H8S/2148 Series**	**H8S/2144 Series**	**H8S/2138 Series**	**H8S/2134 Series**
4 k	H8S/2148	H8S/2144, H8S/2143	H8S/2138	H8S/2134, H8S/2133
2 k	H8S/2147	H8S/2142	H8S/2137	H8S/2132, H8S/2130

4.17 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.



ROM Block Diagram (Example of H8S/2148, H8S/2144, H8S/2138, H8S/2134)

On-chip ROM can be enabled or disabled by means of the mode pins (MD1, MD0) and the EXPE bit in MDCR.

- Flash Memory or Mask ROM Size

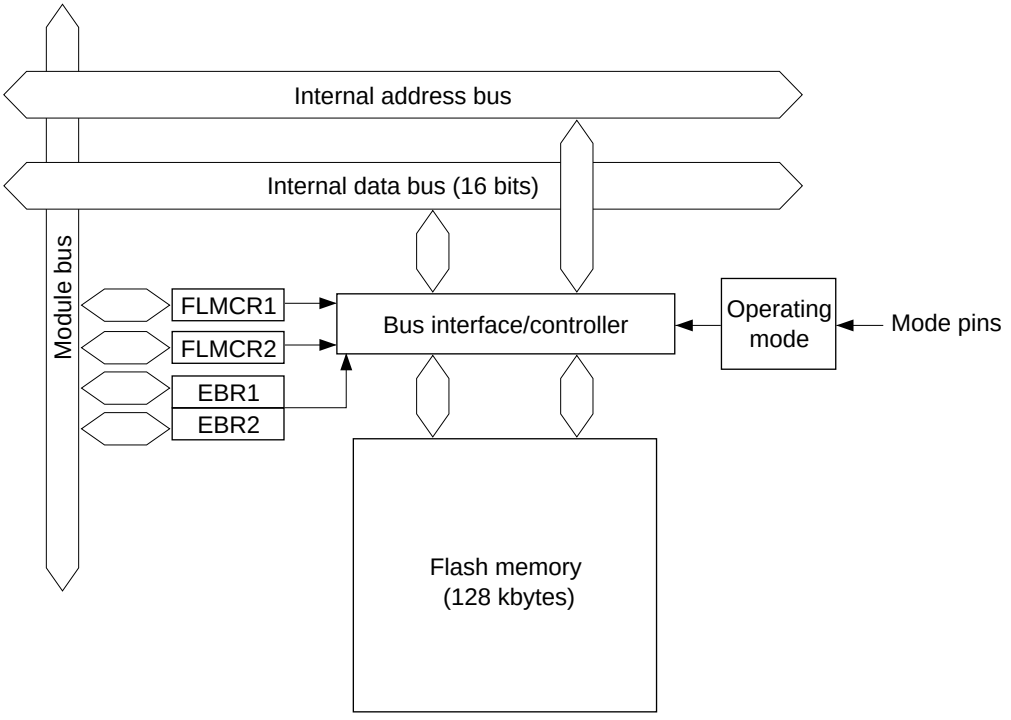
	Product Name			
RAM (Bytes)	**H8S/2148 Series**	**H8S/2144 Series**	**H8S/2138 Series**	**H8S/2134 Series**
128 k	H8S/2148	H8S/2144	H8S/2138	H8S/2134
96 k	—	H8S/2143	—	H8S/2133
64 k	H8S/2147	H8S/2142	H8S/2137	H8S/2132
32 k	—	—	—	H8S/2130

Overview of Flash Memory

Features

- 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.
- 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
- Programmer mode
  - — Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

Block Diagram



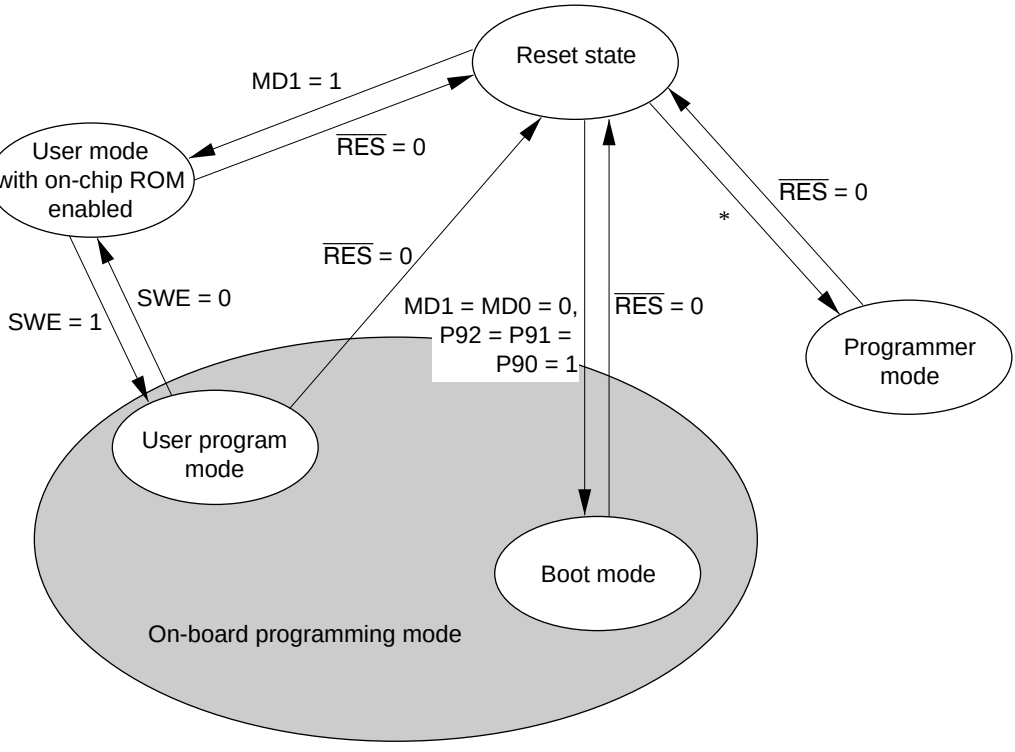
Legend:
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2

Block Diagram of Flash Memory (Example of H8S/2148, H8/2144, H8S/2138, H8S/2134)

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 programmer mode.



Note: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.
* MD0 = MD1 = 0, P92 = 0, P91 = P90 = 1

Mode Transitions

Section 5 Power-Down Modes

In addition to the normal program execution state, the these 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.

These series operating modes are as follows:

1. High-speed mode
2. Medium-speed mode
3. Subactive mode
4. Sleep mode
5. Subsleep mode
6. Watch mode
7. Module stop mode
8. Software standby mode
9. Hardware standby mode

Of these, 2 to 9 are power-down modes. Sleep mode and subsleep mode are CPU modes, medium-speed mode is a CPU and bus master mode, subactive mode is a CPU, bus master, and on-chip supporting module 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 standby control register (SBYCR) are set to 1 in high-speed mode, medium-speed mode is entered as soon as the current bus cycle ends. In medium-speed mode, the bus masters—the CPU and DTC—operate on the operating clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) specified by bits SCK2 to SCK0. However, on-chip supporting modules other than the bus masters operate on the high-speed clock (ϕ).

- Subactive Mode

  If a SLEEP instruction is executed in high-speed mode when the SSBY bit in SBYCR is set to 1, the DTON bit and LSON bit in the low-power control register (LPWRCR) are both set to 1, and the PSS bit in the timer control/status register (TCSR (WDT1)) is set to 1, the CPU enters subactive mode. A transition to subactive mode is also caused by an interrupt generated in watch mode when the LSON bit in LPWRCR is set to 1, and by an interrupt generated in subsleep mode.

  In subactive mode, the CPU performs low-speed sequential program execution using the subclock, and supporting modules other than TMR0, TMR1, WDT0, and WDT1 are halted.

- Sleep Mode

  If a SLEEP instruction is executed when the SSBY bit in 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.

- Subsleep Mode

  If a SLEEP instruction is executed in subactive mode when the SSBY bit in SBYCR is cleared to 0, the LSON bit in LPWRCR is set to 1, and the PSS bit in TCSR (WDT1) is set to 1, the CPU enters subsleep mode.

  In subsleep mode, CPU operation stops, together with the operation of supporting modules other than TMR0, TMR1, WDT0, and WDT1. As long as the specified voltage is supplied, the contents of CPU registers, some on-chip peripheral registers, and on-chip RAM are retained, and I/O ports retain their states prior to the transition.

- Watch Mode

  If a SLEEP instruction is executed in high-speed mode or subactive mode when the SSBY bit in SBYCR is set to 1, the DTON bit in LPWRCR is cleared to 0, and the PSS bit in TCSR (WDT1) is set to 1, the CPU enters watch mode.

  In watch mode, CPU operation stops, together with the operation of supporting modules other than WDT1. As long as the specified voltage is supplied, the contents of CPU registers, some on-chip peripheral registers, and on-chip RAM are retained, and I/O ports retain their states prior to the transition.

- Module Stop Mode

  Module stop mode can be set for individual on-chip supporting modules.

  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.

- Software Standby Mode

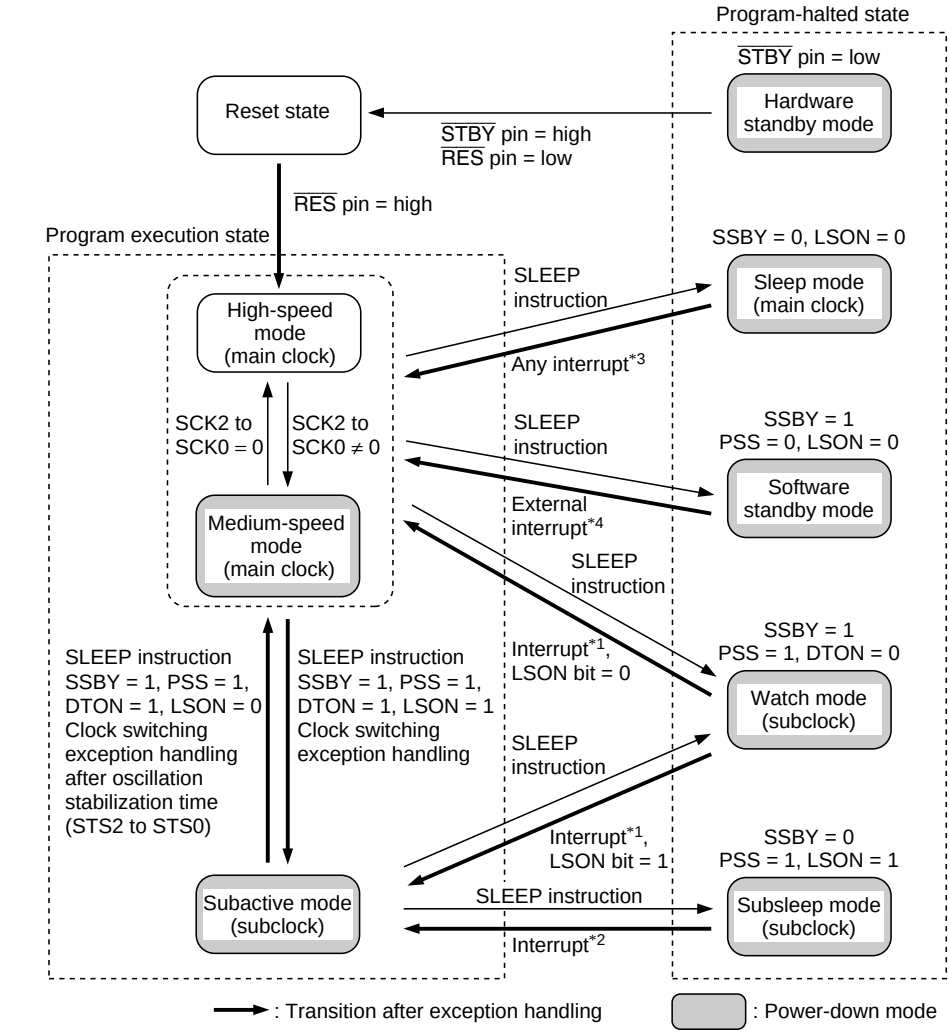
  If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, the LSON bit in LPWRCR is cleared to 0, and the PSS bit in TCSR (WDT1) is cleared to 0, software standby mode is entered. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, on-chip RAM data, the states of on-chip supporting modules other than the SCI, PWM, PWMX, HIF, PS2, D/A, and A/D, and the states of I/O ports, are retained.

- 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.

Mode Transitions



Notes: 1. When a transition is made between modes by means of an interrupt, transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
2. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low.
3. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
4. When making a transition to watch mode, high-speed mode or subactive mode must be set. When making a transition to subactive mode, high-speed mode must be set.

*1 NMI, IRQ0 to IRQ2, IRQ6, IRQ7 and WDT1 interrupts
*2 NMI, IRQ0 to IRQ2, IRQ6, IRQ7 and WDT0 interrupts, WDT1 interrupt, TMR0 interrupt, TMR1 interrupt
*3 All interrupts
*4 NMI, IRQ0 to IRQ2, IRQ6, IRQ7

Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby
System clock oscillator		Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted	Halted	Halted	Halted
Subclock input		Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted
CPU operation	Instruc-tions	Function-ing	Medium-speed operation	Halted	Function-ing	Halted	Subclock operation	Halted	Halted	Halted
	Registers			Retained		Retained		Retained	Retained	Undefined
External interrupts	NMI	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
	IRQ0									
	IRQ1									
	IRQ2									
Peripheral function operation	DTC	Function-ing	Medium-speed operation	Function-ing	Function-ing/halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	WDT1	Function-ing	Function-ing	Function-ing	Function-ing	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT0					Halted (retained)				
	TMR0, 1				Function-ing/halted (retained)					
	FRT						Halted (retained)	Halted (retained)		
	TMRX, Y									
	Timer connec-tion									
	IIC0									
	IIC1									
	SCI0				Function-ing/halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	
	SCI1									
	SCI2									
	PWM									
	PWMX									
	HIF, PS2									
	D/A									
	A/D									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Retained	Retained	High impedance

Note: “Halted (retained)” means that internal register values are retained. The internal state is “operation interrupted.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

Power-Down Mode Transition Conditions

State before Transition	Control Bit States at Time of Transition				State after Transition by SLEEP Instruction	State after Return by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ medium-speed	0	*	0	*	Sleep	High-speed/ medium-speed
	0	*	1	*	—	—
	1	0	0	*	Software standby	High-speed/ medium-speed
	1	0	1	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Subactive
	1	1	0	1	—	—
	1	1	1	1	Subactive	—
Subactive	0	0	*	*	—	—
	0	1	0	*	—	—
	0	1	1	*	Subsleep	Subactive
	1	0	*	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Subactive
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

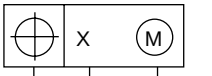
*: Don't care

—: Do not set.

Packages

Package Dimension Diagrams (Unit: mm)

Indication according to geometrical tolerances

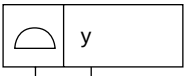


Tolerance method based on maximum solid state

Tolerance value

Kind of geometrical tolerance
(in this case, positional tolerance)

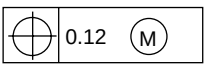
Pin precision y indication



Allowable value

Pin precision

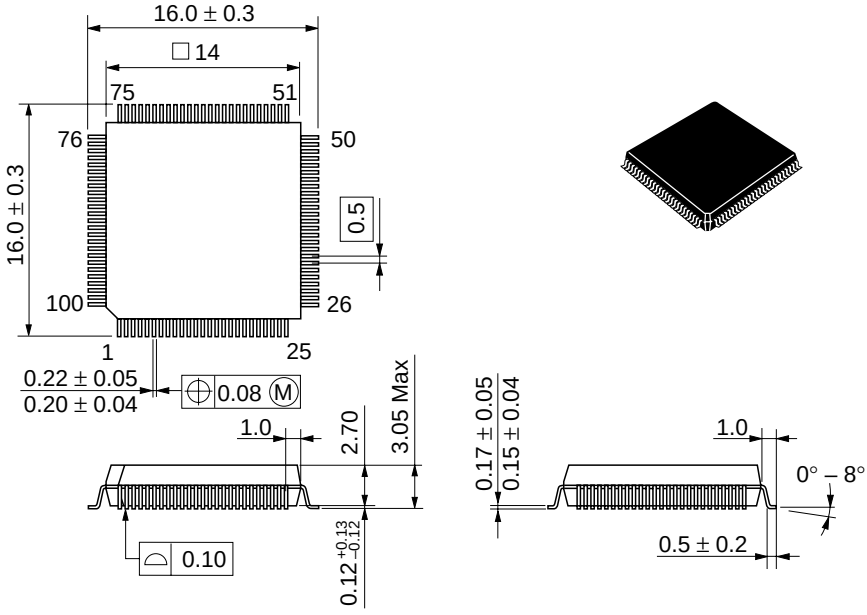
Example:

b 

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.

FP-100B

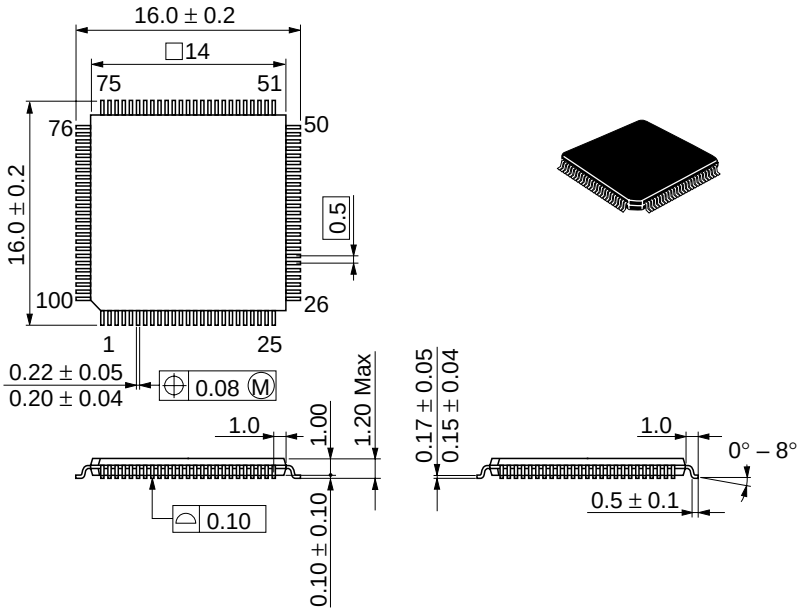
Unit: mm



Dimension including the plating thickness
Base material dimension

TFP-100B

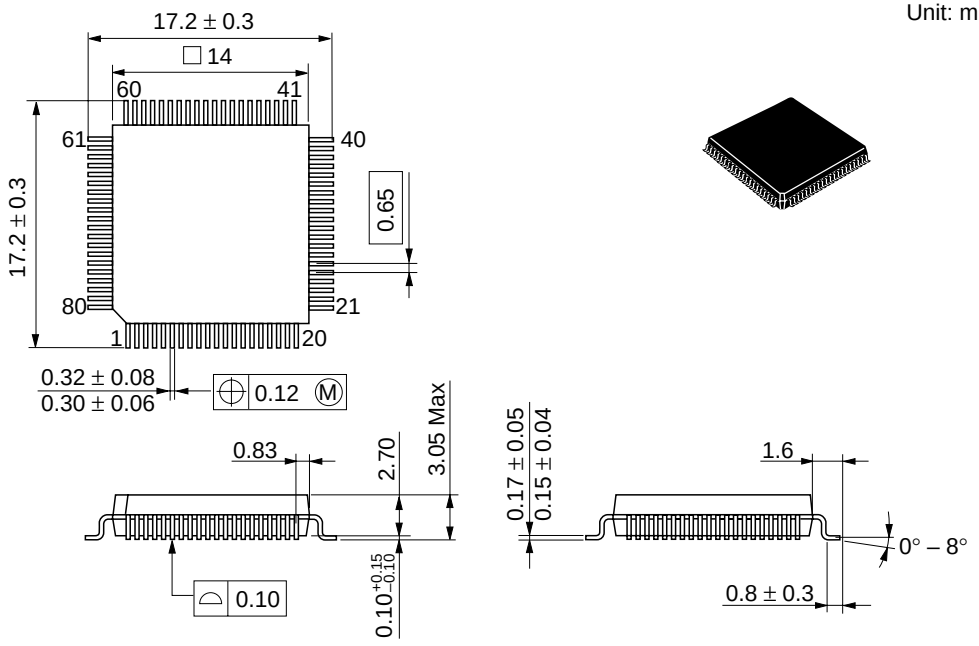
Unit: mm



Dimension including the plating thickness
Base material dimension

FP-80A

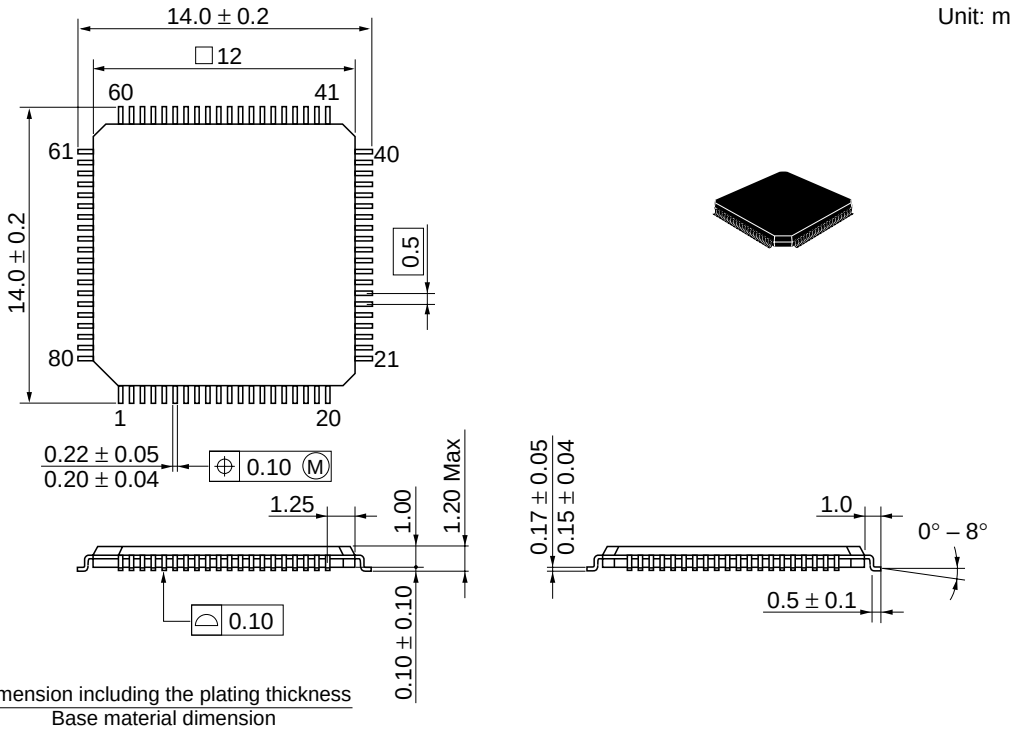
Unit: mm



Dimension including the plating thickness
Base material dimension

TFP-80C

Unit: mm



Dimension including the plating thickness

Base material dimension

H8S/2148 Series, H8S/2144 Series, H8S/2138 Series, H8S/2134 Series Overview

Publication Date: 1st Edition, July 1997
Published by: Semiconductor and IC Div.
Hitachi, Ltd.
Edited by: Technical Documentation Center
Hitachi Microcomputer System Ltd.