2.6.3 Table of Instructions Classified by Function

Table 2-3 summarizes the instructions in each functional category. The notation used in table 2-3 is defined below.

Operation Notation

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

HITACHI

Table 2-3   Instructions Classified by Function

Type	Instruction	Size*[1]	Function
Data transfer	MOV	B/W/L	(EAs) → Rd, Rs → (Ead) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
	MOVFPE	B	Cannot be used in the H8S/2633 Series.
	MOVTPE	B	Cannot be used in the H8S/2633 Series.
	POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
	PUSH	W/L	Rn → @−SP Pushes a register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @−SP. PUSH.L ERn is identical to MOV.L ERn, @−SP.
	LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
	STM	L	Rn (register list) → @−SP Pushes two or more general registers onto the stack.
Arithmetic operations	ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
	ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or on immediate data and data in a general register.
	INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
	ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
	DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.

HITACHI

Type	Instruction	Size*1	Function
Arithmetic operations	MULXU	B/W	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
	DIVXS	B/W	$Rd \div Rs \rightarrow Rd$ Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
	CMP	B/W/L	$Rd - Rs$, $Rd - \#IMM$ Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
	NEG	B/W/L	$0 - Rd \rightarrow Rd$ Takes the two's complement (arithmetic complement) of data in a general register.
	EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
	EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
	TAS*2	B	@ERd – 0, 1 → (<bit 7> of @Erd) Tests memory contents, and sets the most significant bit (bit 7) to 1.
	MAC	—	$(EAs) \times (EAd) + MAC \rightarrow MAC$ Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating

HITACHI

Type	Instruction	Size*[1]	Function
Arithmetic operations	CLRMAC	—	$0 \rightarrow$ MAC Clears the multiply-accumulate register to zero.
	LDMAC STMAC	L	Rs $\rightarrow$ MAC, MAC $\rightarrow$ Rd Transfers data between a general register and a multiply-accumulate register.
Logic operations	AND	B/W/L	Rd $\wedge$ Rs $\rightarrow$ Rd,　Rd $\wedge$ #IMM $\rightarrow$ Rd Performs a logical AND operation on a general register and another general register or immediate data.
	OR	B/W/L	Rd $\vee$ Rs $\rightarrow$ Rd,　Rd $\vee$ #IMM $\rightarrow$ Rd Performs a logical OR operation on a general register and another general register or immediate data.
	XOR	B/W/L	Rd $\oplus$ Rs $\rightarrow$ Rd,　Rd $\oplus$ #IMM $\rightarrow$ Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
	NOT	B/W/L	$\neg$ (Rd) $\rightarrow$ (Rd) Takes the one's complement of general register contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) $\rightarrow$ Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) $\rightarrow$ Rd Performs a logical shift on general register contents. 1-bit or 2-bit shift is possible.
	ROTL ROTR	B/W/L	Rd (rotate) $\rightarrow$ Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
	ROTXL ROTXR	B/W/L	Rd (rotate) $\rightarrow$ Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotation is possible.
Bit-manipulation instructions	BSET	B	$1 \rightarrow$ (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BCLR	B	$0 \rightarrow$ (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BNOT	B	$\neg$ (<bit-No.> of <EAd>) $\rightarrow$ (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.

HITACHI

Type	Instruction	Size*[1]	Function
Bit-manipulation instructions	BTST	B	$\neg$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BAND	B	C $\wedge$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIAND	B	C $\wedge \neg$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BOR	B	C $\vee$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIOR	B	C $\vee \neg$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BXOR	B	C $\oplus$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIXOR	B	C $\oplus \neg$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BLD	B	(\<bit-No.\> of \<EAd\>) $\rightarrow$ C Transfers a specified bit in a general register or memory operand to the carry flag.
	BILD	B	$\neg$ (\<bit-No.\> of \<EAd\>) $\rightarrow$ C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.

HITACHI

Type	Instruction	Size*[1]	Function
Bit-manipulation instructions	BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
	BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.
Branch instructions	Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA(BT)	Always (true)	Always
BRN(BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
BCC(BHS)	Carry clear (high or same)	$C = 0$
BCS(BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

Type	Instruction	Size	Function
	JMP	—	Branches unconditionally to a specified address.
	BSR	—	Branches to a subroutine at a specified address.
	JSR	—	Branches to a subroutine at a specified address.
	RTS	—	Returns from a subroutine
System control instructions	TRAPA	—	Starts trap-instruction exception handling.
	RTE	—	Returns from an exception-handling routine.
	SLEEP	—	Causes a transition to a power-down state.

HITACHI

Type	Instruction	Size[*1]	Function
System control instructions	LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
	ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
	XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
	NOP	—	PC + 2 → PC Only increments the program counter.

Block data transfer instruction	EEPMOV.B	—	if R4L ≠ 0 then
			    Repeat @ER5+ → @ER6+
			      R4L–1 → R4L
			    Until R4L = 0
			else next;

	EEPMOV.W	—	if R4 ≠ 0 then
			    Repeat @ER5+ → @ER6+
			      R4–1 → R4
			    Until R4 = 0
			else next;
			
			Transfers a data block according to parameters set in general registers R4L or R4, ER5, and ER6.
			
			R4L or R4: size of block (bytes)
			ER5: starting source address
			ER6: starting destination address
			
			Execution of the next instruction begins as soon as the transfer is completed.

Notes: *1 Size refers to the operand size.

    B: Byte

    W: Word

    L: Longword

  *2 When using the TAS instruction, use register ER0, ER1, ER4, or ER5.

HITACHI