Floating-point Instruction Set

Sl. no	Instructions	Short Description	Corresponding Fixed point instr	Ref. pg
		ALU		
1	Fn = Fx + Fy	Adds the floating-point operands in Fx and Fy.	Rn = Rx + Ry	B-24
2	Fn = Fx – Fy	Subtracts the floating-point operand in Fy from that in Fx.	Rn = Rx – Ry	B-25
3	COMP(Fx, Fy)	Compares the floating-point operand in Fx with that in register Fy.	COMP(Rx, Ry)	B-29
4	Fn = –Fx	Complements the sign bit of the floating-point operand in Fx.	Rn = –Rx	B-30
5	Fn = ABS Fx	Returns the absolute value of the floating-point operand in Fx by setting the sign bit to 0.	Rn = ABS Rx	B-31
6	Fn = MIN(Fx, Fy)	Returns the smaller of the floating-point operands in Fx and Fy.	Rn = MIN(Rx, Ry)	B-42
7	Fn = MAX(Fx, Fy)	Returns the larger of the floating-point operands in Fx and Fy.	Rn = MAX(Rx, Ry)	B-43
8	Fn = PASS Fx	Passes the floating-point operand in Fx through the ALU to the floating-point field in register Fn.	Rn = PASS Rx	B-32
9	Fn = (Fx + Fy)/2	Adds the floating-point operands in Fx and Fy and divides the result by 2, by decrementing the exponent of the sum before rounding.	Rn = (Rx + RY)/2	B-28
10	Fn = CLIP Fx BY Fy	Returns the floating-point operand in Fx if the absolute value of the operand in Fx is less than the absolute value of the floating-point operand in Fy. Else, returns | Fy | if Fx is positive, and –| Fy | if Fx is negative.	Rn = CLIP Rx BY Ry	B-44
11	Fn = ABS (Fx + Fy)	Adds the floating-point operands in Fx and Fy, and places the absolute value of the normalized result in register Fn.	-	B-26
12	Fn = ABS (Fx – Fy)	Subtracts the floating-point operand in Fy from that in Fx and places the absolute value of the normalized result in register Fn.	-	B-27
13	Fn = RND Fx	Rounds the floating-point operand in Fx.	-	B-33

14	**Fn = SCALB Fx BY Ry**	**Scales the exponent of the floating-point operand in Fx by adding to it the fixed-point twos-complement integer in Ry.**	**-**	**B-34**
15	**Rn = MANT Fx**	**Extracts the mantissa (fraction bits with the hidden bit, excluding the sign bit) from the floating-point operand in Fx.**	**-**	**B-35**
16	**Rn = LOGB Fx**	**Converts the exponent of the floating-point operand in Fx to an unbiased twos-complement fixed-point integer.**	**-**	**B-36**
17	**Rn = FIX Fx BY Ry**	**Converts the floating-point operand in Fx to a twos-complement 32-bit fixed-point integer result. For the scaling factor (Ry), the fixed-point twos-complement integer in Ry is added to the exponent of the floating-point operand in Fx before the conversion.**	**-**	**B-37**
18	**Rn = FIX Fx**	**Converts the floating-point operand in Fx to a twos-complement 32-bit fixed-point integer result.**	**-**	**B-37**
19	**Rn = TRUNC Fx BY Ry**	**Converts the floating-point operand in Fx to a twos-complement 32-bit fixed-point integer result. The TRUNC operation always truncates toward 0. For the scaling factor (Ry), the fixed-point twos-complement integer in Ry is added to the exponent of the floating-point operand in Fx before the conversion.**	**-**	**B-37**
20	**Rn = TRUNC Fx**	**Converts the floating-point operand in Fx to a twos-complement 32-bit fixed-point integer result. The TRUNC operation always truncates toward 0.**	**-**	**B-37**
21	**Fn = FLOAT Rx BY Ry**	**Converts the fixed-point operand in Rx to a floating-point result. For the scaling factor (Ry), the fixed-point twos-complement integer in Ry is added to the exponent of the floating-point result.**	**-**	**B-38**
22	**Fn = FLOAT Rx**	**Converts the fixed-point operand in Rx to a floating-point result.**	**-**	**B-38**
23	**Fn = RECIPS Fx**	**Creates an 8-bit accurate seed for 1/Fx, the reciprocal of Fx.**	**-**	**B-39**
24	**Fn = RSQRTS Fx**	**Creates a 4-bit accurate seed for 1/√Fx, the reciprocal square root of Fx.**	**-**	**B-40**
25	**Fn = Fx COPYSIGN Fy**	**Copies the sign of the floating-point operand in Fy to the floating-point operand from Fx without changing the exponent or the mantissa.**	**-**	**B-41**

MULTIPLIER				
26	**Fn = Fx * Fy**	**Multiplies the floating-point operands in registers Fx and Fy.**	**Rn = Rx * Ry**	**B-53**
SHIFTER				
27	**Rn = FPACK Fx**	**Converts the IEEE 32-bit floating-point value in Fx to a 16-bit floating-point value stored in Rn.**	**-**	**B-74**
28	**Fn = FUNPACK Rx**	**Converts the 16-bit floating-point value in Rx to an IEEE 32-bit floating-point value stored in Fx.**	**-**	**B-75**

- Floating-point ***Specific*** instructions
- Floating-point instructions in ***correspondence*** with Fixed-point instructions in SHARC (***present*** in MISA 1 instruction set)
- Floating-point instructions in ***correspondence*** with Fixed-point instructions in SHARC (***Not present*** in MISA 1 instruction set)