

# MIPS® DSP ASE Instruction Set Quick Reference

- Rd, Rs, Rt — DESTINATION (Rd) AND SOURCE (Rs, Rt) REGISTERS
- Ac — ACCUMULATOR REGISTER (Ac0 – Ac3)
- C, Cc — CARRY (BIT 13) AND CONDITION CODE FLAGS (BITS 24-27) IN DSPCONTROL REGISTER
- POS, SIZE — POSITION AND SIZE (SCOUNT) FIELDS IN DSPCONTROL REGISTER
- ± — SIGNED OPERAND/OPERATION OR SIGN EXTENSION
- ∅ — UNSIGNED OPERAND/OPERATION OR ZERO EXTENSION
- × — INTEGER MULTIPLICATION
- ⊗ / • — FRACTIONAL MULTIPLICATION (IMPLIED SHIFT LEFT BY 1 BIT) WITH / WITHOUT ROUNDING
- ⊗ / [ ] — ROUNDING AND SATURATION OPERATIONS
- L / R — LEFT / RIGHT 16-BIT PART OF A RESULT OR A REGISTER
- LL, LR, RL, RR — THE FOUR BYTES IN A 32-BIT REGISTER, FROM LEFT (MSB) TO RIGHT (LSB)
- || — BOUNDARY BETWEEN TWO OR FOUR SIMD ELEMENTS IN A REGISTER
- :: — CONCATENATION OF BIT FIELDS
- R2 — DSP ASE REVISION 2 INSTRUCTION

## ARITHMETIC OPERATIONS: 8-BIT DATA

ABSQ_S.QB <sup>R2</sup>	Rd, Rs	$RD_{XY} = \llbracket RS_{XY}^{\pm} \rrbracket$	$XY \in \{ LL, LR, RL, RR \}$
ADDU.QB	Rd, Rs, Rt	$RD_{XY} = RS_{XY}^{\ominus} + RT_{XY}^{\ominus}$	$XY \in \{ LL, LR, RL, RR \}$
ADDU_S.QB	Rd, Rs, Rt	$RD_{XY} = \llbracket RS_{XY}^{\ominus} + RT_{XY}^{\ominus} \rrbracket$	$XY \in \{ LL, LR, RL, RR \}$
ADDUH.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} + RT_{XY}^{\ominus}) \ggg 1$	$XY \in \{ LL, LR, RL, RR \}$
ADDUH_R.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} + RT_{XY}^{\ominus} + 1^{\ominus}) \ggg 1$	$XY \in \{ LL, LR, RL, RR \}$
RADDU.W.QB	Rd, Rs	$RD = RS_{LL}^{\ominus} + RS_{LR}^{\ominus} + RS_{RL}^{\ominus} + RS_{RR}^{\ominus}$	
REPL.QB	Rd, CONST8	$RD = CONST8 \lll CONST8 \lll CONST8 \lll CONST8$	
REPLV.QB	Rd, Rs	$RD = RS_{7,0} \lll RS_{7,0} \lll RS_{7,0} \lll RS_{7,0}$	
SUBU.QB	Rd, Rs, Rt	$RD_{XY} = RS_{XY}^{\ominus} - RT_{XY}^{\ominus}$	$XY \in \{ LL, LR, RL, RR \}$
SUBU_S.QB	Rd, Rs, Rt	$RD_{XY} = \llbracket RS_{XY}^{\ominus} - RT_{XY}^{\ominus} \rrbracket$	$XY \in \{ LL, LR, RL, RR \}$
SUBUH.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} - RT_{XY}^{\ominus}) \ggg 1$	$XY \in \{ LL, LR, RL, RR \}$
SUBUH_R.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} - RT_{XY}^{\ominus} + 1^{\ominus}) \ggg 1$	$XY \in \{ LL, LR, RL, RR \}$

## SHIFT OPERATIONS: 8-BIT DATA

SHLL.QB	Rd, Rs, SHIFT3	$RD_{XY} = RS_{XY} \lll SHIFT3$	$XY \in \{ LL, LR, RL, RR \}$
SHLLV.QB	Rd, Rs, Rt	$RD_{XY} = RS_{XY} \lll RT_{2,0}$	$XY \in \{ LL, LR, RL, RR \}$
SHRA.QB <sup>R2</sup>	Rd, Rs, SHIFT3	$RD_{XY} = RS_{XY}^{\pm} \ggg SHIFT3$	$XY \in \{ LL, LR, RL, RR \}$
SHRA_R.QB <sup>R2</sup>	Rd, Rs, SHIFT3	$RD_{XY} = \textcircled{\text{R}}(RS_{XY}^{\pm} \ggg SHIFT3)$	$XY \in \{ LL, LR, RL, RR \}$
SHRAV.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = RS_{XY}^{\pm} \ggg RT_{2,0}$	$XY \in \{ LL, LR, RL, RR \}$
SHRAV_R.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = \textcircled{\text{R}}(RS_{XY}^{\pm} \ggg RT_{2,0})$	$XY \in \{ LL, LR, RL, RR \}$
SHRL.QB	Rd, Rs, SHIFT3	$RD_{XY} = RS_{XY}^{\ominus} \ggg SHIFT3$	$XY \in \{ LL, LR, RL, RR \}$
SHRLV.QB	Rd, Rs, Rt	$RD_{XY} = RS_{XY}^{\ominus} \ggg RT_{2,0}$	$XY \in \{ LL, LR, RL, RR \}$

## UNSIGNED INTEGER MULTIPLY OPERATIONS: 8-BIT/16-BIT DATA » GPR

MULEU_S.PH.QBL	Rd, Rs, Rt	$RD = \llbracket RS_{LL}^{\ominus} \times RT_{LL}^{\ominus} \rrbracket \lll \llbracket RS_{LR}^{\ominus} \times RT_{LR}^{\ominus} \rrbracket$	
MULEU_S.PH.QBR	Rd, Rs, Rt	$RD = \llbracket RS_{RL}^{\ominus} \times RT_{RL}^{\ominus} \rrbracket \lll \llbracket RS_{RR}^{\ominus} \times RT_{RR}^{\ominus} \rrbracket$	

## UNSIGNED INTEGER MULTIPLY OPERATIONS: 8-BIT DATA » ACCUMULATOR

DPAU.H.QBL	Ac, Rs, Rt	$AC += (RS_{LL}^{\ominus} \times RT_{LL}^{\ominus}) + (RS_{LR}^{\ominus} \times RT_{LR}^{\ominus})$	
DPAU.H.QBR	Ac, Rs, Rt	$AC += (RS_{RL}^{\ominus} \times RT_{RL}^{\ominus}) + (RS_{RR}^{\ominus} \times RT_{RR}^{\ominus})$	
DPSU.H.QBL	Ac, Rs, Rt	$AC -= (RS_{LL}^{\ominus} \times RT_{LL}^{\ominus}) + (RS_{LR}^{\ominus} \times RT_{LR}^{\ominus})$	
DPSU.H.QBR	Ac, Rs, Rt	$AC -= (RS_{RL}^{\ominus} \times RT_{RL}^{\ominus}) + (RS_{RR}^{\ominus} \times RT_{RR}^{\ominus})$	

## PRECISION EXPANSION (DATA UNPACKING) OPERATIONS: 8-BIT DATA

PRECEQU.PH.QBL	Rd, Rs	$RD = (RS_{LL} \lll 7) \lll (RS_{LR} \lll 7)$	
PRECEQU.PH.QBLA	Rd, Rs	$RD = (RS_{LL} \lll 7) \lll (RS_{RL} \lll 7)$	
PRECEQU.PH.QBR	Rd, Rs	$RD = (RS_{RL} \lll 7) \lll (RS_{RR} \lll 7)$	
PRECEQU.PH.QBRA	Rd, Rs	$RD = (RS_{LR} \lll 7) \lll (RS_{RR} \lll 7)$	
PRECEU.PH.QBL	Rd, Rs	$RD = 0^8 \lll RS_{LL} \lll 0^8 \lll RS_{LR}$	
PRECEU.PH.QBLA	Rd, Rs	$RD = 0^8 \lll RS_{LL} \lll 0^8 \lll RS_{RL}$	
PRECEU.PH.QBR	Rd, Rs	$RD = 0^8 \lll RS_{RL} \lll 0^8 \lll RS_{RR}$	
PRECEU.PH.QBRA	Rd, Rs	$RD = 0^8 \lll RS_{LR} \lll 0^8 \lll RS_{RR}$	

## COMPARE AND PICK OPERATIONS: 8-BIT DATA



CMPGDU.EQ.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = CC_{XY} = (RS_{XY}^{\ominus} = RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGDU.LT.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = CC_{XY} = (RS_{XY}^{\ominus} < RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGDU.LE.QB <sup>R2</sup>	Rd, Rs, Rt	$RD_{XY} = CC_{XY} = (RS_{XY}^{\ominus} \leq RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGU.EQ.QB	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} = RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGU.LT.QB	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} < RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGU.LE.QB	Rd, Rs, Rt	$RD_{XY} = (RS_{XY}^{\ominus} \leq RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPU.EQ.QB	Rd, Rs, Rt	$CC_{XY} = (RS_{XY}^{\ominus} = RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPU.LT.QB	Rd, Rs, Rt	$CC_{XY} = (RS_{XY}^{\ominus} < RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPU.LE.QB	Rd, Rs, Rt	$CC_{XY} = (RS_{XY}^{\ominus} \leq RT_{XY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
PICK.QB	Rd, Rs, Rt	$RD_{XY} = CC_{XY} ? RS_{XY} : RT_{XY}$	$XY \in \{ LL, LR, RL, RR \}$



## COMPARE AND PICK OPERATIONS: 16-BIT DATA

CMP.EQ.PH	Rd, Rs, Rt	$CC = ((RS_{L}^{\pm} = RT_{L}^{\pm}) ? 1 : 0) \lll ((RS_{R}^{\pm} = RT_{R}^{\pm}) ? 1 : 0)$
CMP.LT.PH	Rd, Rs, Rt	$CC = ((RS_{L}^{\pm} < RT_{L}^{\pm}) ? 1 : 0) \lll ((RS_{R}^{\pm} < RT_{R}^{\pm}) ? 1 : 0)$
CMP.LE.PH	Rd, Rs, Rt	$CC = ((RS_{L}^{\pm} \leq RT_{L}^{\pm}) ? 1 : 0) \lll ((RS_{R}^{\pm} \leq RT_{R}^{\pm}) ? 1 : 0)$
PICK.PH	Rd, Rs, Rt	$RD = (CC_L ? RS_L : RT_L) \lll (CC_R ? RS_R : RT_R)$

ARITHMETIC OPERATIONS: 16-BIT DATA		
ABSQ_S.PH	R <sub>D</sub> , R <sub>S</sub>	$R_D = [R_{S_L}^{\pm}] \parallel [R_{S_R}^{\pm}]$
ADDQ.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\pm} + R_{T_L}^{\pm}) \parallel (R_{S_R}^{\pm} + R_{T_R}^{\pm})$
ADDQ_S.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\pm} + R_{T_L}^{\pm}] \parallel [R_{S_R}^{\pm} + R_{T_R}^{\pm}]$
ADDQH.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = ((R_{S_L}^{\pm} + R_{T_L}^{\pm}) \ggg 1) \parallel ((R_{S_R}^{\pm} + R_{T_R}^{\pm}) \ggg 1)$
ADDQH_R.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = @((R_{S_L}^{\pm} + R_{T_L}^{\pm}) \ggg 1) \parallel @((R_{S_R}^{\pm} + R_{T_R}^{\pm}) \ggg 1)$
ADDU.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\circ} + R_{T_L}^{\circ}) \parallel (R_{S_R}^{\circ} + R_{T_R}^{\circ})$
ADDU_S.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\circ} + R_{T_L}^{\circ}] \parallel [R_{S_R}^{\circ} + R_{T_R}^{\circ}]$
REPL.PH	R <sub>D</sub> , CONST10	$R_D = \text{CONST10}^{\pm} \parallel \text{CONST10}^{\pm}$
REPLV.PH	R <sub>D</sub> , R <sub>S</sub>	$R_D = R_{S_{15:0}} \parallel R_{S_{15:0}}$
SUBQ.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\pm} - R_{T_L}^{\pm}) \parallel (R_{S_R}^{\pm} - R_{T_R}^{\pm})$
SUBQ_S.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\pm} - R_{T_L}^{\pm}] \parallel [R_{S_R}^{\pm} - R_{T_R}^{\pm}]$
SUBQH.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = ((R_{S_L}^{\pm} - R_{T_L}^{\pm}) \ggg 1) \parallel ((R_{S_R}^{\pm} - R_{T_R}^{\pm}) \ggg 1)$
SUBQH_R.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = @((R_{S_L}^{\pm} - R_{T_L}^{\pm}) \ggg 1) \parallel @((R_{S_R}^{\pm} - R_{T_R}^{\pm}) \ggg 1)$
SUBU.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\circ} - R_{T_L}^{\circ}) \parallel (R_{S_R}^{\circ} - R_{T_R}^{\circ})$
SUBU_S.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\circ} - R_{T_L}^{\circ}] \parallel [R_{S_R}^{\circ} - R_{T_R}^{\circ}]$


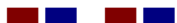

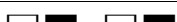
SHIFT OPERATIONS: 16-BIT DATA		
SHLL.PH	R <sub>D</sub> , R <sub>S</sub> , SHIFT4	$R_D = (R_{S_L} \ll \text{SHIFT4}) \parallel (R_{S_R} \ll \text{SHIFT4})$
SHLL_S.PH	R <sub>D</sub> , R <sub>S</sub> , SHIFT4	$R_D = [R_{S_L} \ll \text{SHIFT4}] \parallel [R_{S_R} \ll \text{SHIFT4}]$
SHLLV.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L} \ll R_{T_{3:0}}) \parallel (R_{S_R} \ll R_{T_{3:0}})$
SHLLV_S.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L} \ll R_{T_{3:0}}] \parallel [R_{S_R} \ll R_{T_{3:0}}]$
SHRA.PH	R <sub>D</sub> , R <sub>S</sub> , SHIFT4	$R_D = (R_{S_L}^{\pm} \ggg \text{SHIFT4}) \parallel (R_{S_R}^{\pm} \ggg \text{SHIFT4})$
SHRA_R.PH	R <sub>D</sub> , R <sub>S</sub> , SHIFT4	$R_D = @((R_{S_L}^{\pm} \ggg \text{SHIFT4}) \parallel @((R_{S_R}^{\pm} \ggg \text{SHIFT4}))$
SHRAV.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\pm} \ggg R_{T_{3:0}}) \parallel (R_{S_R}^{\pm} \ggg R_{T_{3:0}})$
SHRAV_R.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = @((R_{S_L}^{\pm} \ggg R_{T_{3:0}}) \parallel @((R_{S_R}^{\pm} \ggg R_{T_{3:0}}))$
SHRL.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , SHIFT4	$R_D = (R_{S_L}^{\circ} \ggg \text{SHIFT4}) \parallel (R_{S_R}^{\circ} \ggg \text{SHIFT4})$
SHRLV.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\circ} \ggg R_{T_{3:0}}) \parallel (R_{S_R}^{\circ} \ggg R_{T_{3:0}})$




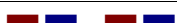
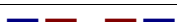
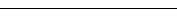
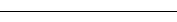
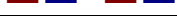

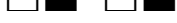

PRECISION REDUCTION (DATA PACKING) OPERATIONS: 16-BIT DATA		
PRECR.QB.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = R_{S_{LR}} \parallel R_{S_{RR}} \parallel R_{T_{LR}} \parallel R_{T_{RR}}$ 
PRECRQ.QB.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = R_{S_{LL}} \parallel R_{S_{RL}} \parallel R_{T_{LL}} \parallel R_{T_{RL}}$ 
PRECRQU_S.QB.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\circ}]_{14:7} \parallel [R_{S_R}^{\circ}]_{14:7} \parallel [R_{T_L}^{\circ}]_{14:7} \parallel [R_{T_R}^{\circ}]_{14:7}$

PRECISION EXPANSION (DATA UNPACKING) OPERATIONS: 16-BIT DATA		
PRECEQ.W.PHL	R <sub>D</sub> , R <sub>S</sub>	$R_D = R_{S_L} \ll 16$ 
PRECEQ.W.PHR	R <sub>D</sub> , R <sub>S</sub>	$R_D = R_{S_R} \ll 16$ 

INTEGER MULTIPLY OPERATIONS: 16-BIT DATA » ACCUMULATOR		
DPA.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += (R_{S_L}^{\pm} \times R_{T_L}^{\pm}) + (R_{S_R}^{\pm} \times R_{T_R}^{\pm})$
DPAX.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += (R_{S_L}^{\pm} \times R_{T_R}^{\pm}) + (R_{S_R}^{\pm} \times R_{T_L}^{\pm})$
DPS.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac -= (R_{S_L}^{\pm} \times R_{T_L}^{\pm}) + (R_{S_R}^{\pm} \times R_{T_R}^{\pm})$
DPSX.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac -= (R_{S_L}^{\pm} \times R_{T_R}^{\pm}) + (R_{S_R}^{\pm} \times R_{T_L}^{\pm})$
MULSA.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += (R_{S_L}^{\pm} \times R_{T_L}^{\pm}) - (R_{S_R}^{\pm} \times R_{T_R}^{\pm})$

INTEGER MULTIPLY OPERATIONS: 16-BIT DATA » GPR		
MUL.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = (R_{S_L}^{\pm} \times R_{T_L}^{\pm})_R \parallel (R_{S_R}^{\pm} \times R_{T_R}^{\pm})_R$
MUL_S.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\pm} \times R_{T_L}^{\pm}]_R \parallel [R_{S_R}^{\pm} \times R_{T_R}^{\pm}]_R$

FRACTIONAL MULTIPLY OPERATIONS: 16-BIT DATA » GPR		
MULQ_RS.PH	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\pm} \odot R_{T_L}^{\pm}]_L \parallel [R_{S_R}^{\pm} \odot R_{T_R}^{\pm}]_L$ 
MULQ_S.PH <sup>R2</sup>	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}]_L \parallel [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]_L$ 
MULEQ_S.W.PHL	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}]$ 
MULEQ_S.W.PHR	R <sub>D</sub> , R <sub>S</sub> , R <sub>T</sub>	$R_D = [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]$ 

FRACTIONAL MULTIPLY OPERATIONS: 16-BIT DATA » ACCUMULATOR		
DPAQ_S.W.PH	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}] + [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]$ 
DPAQX_S.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += [R_{S_L}^{\pm} \bullet R_{T_R}^{\pm}] + [R_{S_R}^{\pm} \bullet R_{T_L}^{\pm}]$ 
DPAQX_SA.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac = [Ac + [R_{S_L}^{\pm} \bullet R_{T_R}^{\pm}] + [R_{S_R}^{\pm} \bullet R_{T_L}^{\pm}]]$ 
DPSQ_S.W.PH	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac -= [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}] + [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]$ 
DPSQX_S.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac -= [R_{S_L}^{\pm} \bullet R_{T_R}^{\pm}] + [R_{S_R}^{\pm} \bullet R_{T_L}^{\pm}]$ 
DPSQX_SA.W.PH <sup>R2</sup>	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac = [Ac - [R_{S_L}^{\pm} \bullet R_{T_R}^{\pm}] + [R_{S_R}^{\pm} \bullet R_{T_L}^{\pm}]]$ 
MULSAQ_S.W.PH	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}] - [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]$ 
MAQ_S.W.PHL	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}]$ 
MAQ_S.W.PHR	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac += [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]$ 
MAQ_SA.W.PHL	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac = [Ac^+ + [R_{S_L}^{\pm} \bullet R_{T_L}^{\pm}]]$ 
MAQ_SA.W.PHR	Ac, R <sub>S</sub> , R <sub>T</sub>	$Ac = [Ac^+ + [R_{S_R}^{\pm} \bullet R_{T_R}^{\pm}]]$ 

ACCUMULATOR EXTRACT OPERATIONS: 16-BIT DATA		
EXTR_S.H	R <sub>D</sub> , Ac, SHIFT5	$R_D = [Ac \ggg \text{SHIFT5}]_R$
EXTRV_S.H	R <sub>D</sub> , Ac, R <sub>T</sub>	$R_D = [Ac \ggg R_{T_{4:0}}]_R$

DSP CONTROL REGISTER ACCESS OPERATIONS		
RDDSP	R <sub>D</sub> , MASK10	$R_D = \text{READDSPCONTROL}(\text{MASK10}_{5:0})$
WRDSP	R <sub>S</sub> , MASK10	$\text{WRITEDSPCONTROL}(\text{MASK10}_{5:0}, R_S)$


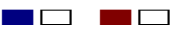
ARITHMETIC OPERATIONS: 32-BIT DATA		
ABSQ_S.W	Rd, Rs	$R_D = [Rs]$
ADDQ_S.W	Rd, Rs, Rt	$R_D = [Rs^+ + Rt^+]$
ADDQH.W <sup>R2</sup>	Rd, Rs, Rt	$R_D = (Rs^+ + Rt^+) \gg 1$
ADDQH_R.W <sup>R2</sup>	Rd, Rs, Rt	$R_D = @((Rs^+ + Rt^+) \gg 1)$
ADDSC	Rd, Rs, Rt	$C::R_D = Rs^\ominus + Rt^\ominus$
ADDWC	Rd, Rs, Rt	$R_D = Rs^\ominus + Rt^\ominus + C$
SUBQ_S.W	Rd, Rs, Rt	$R_D = [Rs - Rt]$
SUBQH.W <sup>R2</sup>	Rd, Rs, Rt	$R_D = (Rs^+ - Rt^+) \gg 1$
SUBQH_R.W <sup>R2</sup>	Rd, Rs, Rt	$R_D = @((Rs^+ - Rt^+) \gg 1)$

SHIFT OPERATIONS: 32-BIT DATA		
SHLL_S.W	Rd, Rs, SHIFT5	$R_D = [Rs \ll SHIFT5]$
SHLLV_S.W	Rd, Rs, Rt	$R_D = [Rs \ll Rt_{40}]$
SHRA_R.W	Rd, Rs, SHIFT5	$R_D = @([Rs^+ \gg SHIFT5])$
SHRAV_R.W	Rd, Rs, Rt	$R_D = @([Rs^+ \gg Rt_{40}])$

FRACTIONAL MULTIPLY OPERATIONS: 32-BIT DATA » GPR		
MULQ_RS.W <sup>R2</sup>	Rd, Rs, Rt	$R_D = [Rs^+ \odot Rt^+]_L$
MULQ_S.W <sup>R2</sup>	Rd, Rs, Rt	$R_D = [Rs^+ \bullet Rt^+]_L$

FRACTIONAL MULTIPLY OPERATIONS: 32-BIT DATA » ACCUMULATOR		
DPAQ_SA.L.W	Ac, Rs, Rt	$Ac = [Ac^+ + [Rs^+ \bullet Rt^+]]$
DPSQ_SA.L.W	Ac, Rs, Rt	$Ac = [Ac^+ - [Rs^+ \bullet Rt^+]]$

INTEGER MULTIPLY OPERATIONS: 32-BIT DATA » ACCUMULATOR		
MADD <sup>R2</sup>	Ac, Rs, Rt	$Ac += Rs^+ \times Rt^+$
MADDU <sup>R2</sup>	Ac, Rs, Rt	$Ac += Rs^\ominus \times Rt^\ominus$
MSUB <sup>R2</sup>	Ac, Rs, Rt	$Ac -= Rs^+ \times Rt^+$
MSUBU <sup>R2</sup>	Ac, Rs, Rt	$Ac -= Rs^\ominus \times Rt^\ominus$
MULT <sup>R2</sup>	Ac, Rs, Rt	$Ac = Rs^+ \times Rt^+$
MULTU <sup>R2</sup>	Ac, Rs, Rt	$Ac = Rs^\ominus \times Rt^\ominus$

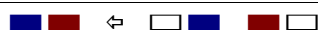
PRECISION REDUCTION (DATA PACKING) OPERATIONS: 32-BIT DATA		
PRECRQ.PH.W	Rd, Rs, Rt	$R_D = Rs_L \parallel Rt_L$ 
PRECRQ_RS.PH.W	Rd, Rs, Rt	$R_D = @[Rs^+]_L \parallel @[Rt^+]_L$ 
PRECR_SRA.PH.W <sup>R2</sup>	Rd, Rs, SHIFT5	$R_D = (Rd^+ \gg SHIFT5)_R \parallel (Rs^+ \gg SHIFT5)_R$
PRECR_SRA_R.PH.W <sup>R2</sup>	Rd, Rs, SHIFT5	$R_D = @((Rd^+ \gg SHIFT5)_R \parallel @(Rs^+ \gg SHIFT5)_R)$

ACCUMULATOR EXTRACT OPERATIONS: 32-BIT DATA		
EXTR.W	Rd, Ac, SHIFT5	$R_D = (Ac \gg SHIFT5)_R$
EXTR_R.W	Rd, Ac, SHIFT5	$R_D = @[Ac \gg SHIFT5]_R$
EXTR_RS.W	Rd, Ac, SHIFT5	$R_D = @[Ac \gg SHIFT5]_R$
EXTRV.W	Rd, Ac, Rt	$R_D = (Ac \gg Rt_{40})_R$
EXTRV_R.W	Rd, Ac, Rt	$R_D = @[Ac \gg Rt_{40}]_R$
EXTRV_RS.W	Rd, Ac, Rt	$R_D = @[Ac \gg Rt_{40}]_R$

BIT-FIELD EXTRACT OPERATIONS		
BPOSGE32	OFF18	IF $POS \geq 32$ , $PC += OFF18^\pm$
EXTP	Rd, Ac, SIZE5	$R_D = Ac_{POS:POS-SIZE5}$
EXTPDP	Rd, Ac, SIZE5	$R_D = Ac_{POS:POS-SIZE5}$ , $POS -= (SIZE5 + 1)$
EXTPV	Rd, Ac, Rs	$R_D = Ac_{POS:POS-RS40}$
EXTPDPV	Rd, Ac, Rs	$R_D = Ac_{POS:POS-RS40}$ , $POS -= (RS40 + 1)$
MTHLIP	Rs, Ac	$Ac_{HI} = Ac_{LO}$ , $Ac_{LO} = Rs$ , $POS += 32$

LOAD DATA OPERATIONS		
BITREV	Rd, Rs	$R_D = Rs_{0:15}$
LBUX	Rd, Rs(Rt)	$R_D = MEM8(Rs + Rt)^\ominus$
LHX	Rd, Rs(Rt)	$R_D = MEM16(Rs + Rt)^\pm$
LWX	Rd, Rs(Rt)	$R_D = MEM32(Rs + Rt)$
MODSUB	Rd, Rs, Rt	$R_D = (Rs \neq 0) ? (Rs - Rt_{7:0}) : Rt_{23:8}$

ACCUMULATOR OPERATIONS		
MFHI	Rd, Ac	$R_D = Ac_{HI}$
MFLO	Rd, Ac	$R_D = Ac_{LO}$
MTHI	Rs, Ac	$Ac_{HI} = Rs$
MTLO	Rs, Ac	$Ac_{LO} = Rs$
SHILO	Ac, SHIFT6	$Ac = (SHIFT6^\pm \geq 0) ? (Ac \gg SHIFT6^\pm) : (Ac \ll -SHIFT6^\pm)$
SHILOV	Ac, Rs	$Ac = (Rs_{5:0}^\pm \geq 0) ? (Ac \gg Rs_{5:0}^\pm) : (Ac \ll -Rs_{5:0}^\pm)$

BIT-FIELD OPERATIONS		
APPEND <sup>R2</sup>	Rd, Rs, SHIFT5	$R_D = (Rd \ll SHIFT5) :: Rs_{SHIFT5-1:0}$
BALIGN <sup>R2</sup>	Rd, Rs, BPOS2	$R_D = (Rd \ll (8 \times BPOS2)) :: (Rs \gg (32 - 8 \times BPOS2))$
INSV	Rd, Rs	$R_{D_{POS+SIZE-1:POS}} = Rs_{SIZE-1:0}$
PACKRL.PH	Rd, Rs, Rt	$R_D = Rs_R \parallel Rt_L$ 
PREPEND <sup>R2</sup>	Rd, Rs, SHIFT5	$R_D = Rs_{SHIFT5-1:0} :: (Rd \gg SHIFT5)$