

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 ^{R2}	Rd, Rs	$R_{DXY} = \llbracket R_{SXY}^{\pm} \rrbracket$	$XY \in \{ LL, LR, RL, RR \}$
ADDU.QB	Rd, Rs, Rt	$R_{DXY} = R_{SXY}^{\ominus} + R_{TXY}^{\ominus}$	$XY \in \{ LL, LR, RL, RR \}$
ADDU_S.QB	Rd, Rs, Rt	$R_{DXY} = \llbracket R_{SXY}^{\ominus} + R_{TXY}^{\ominus} \rrbracket$	$XY \in \{ LL, LR, RL, RR \}$
ADDUH.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} + R_{TXY}^{\ominus}) >> 1$	$XY \in \{ LL, LR, RL, RR \}$
ADDUH_R.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} + R_{TXY}^{\ominus} + 1^{\ominus}) >> 1$	$XY \in \{ LL, LR, RL, RR \}$
RADDU.W.QB	Rd, Rs	$R_D = R_{S_{LL}}^{\ominus} + R_{S_{LR}}^{\ominus} + R_{S_{RL}}^{\ominus} + R_{S_{RR}}^{\ominus}$	
REPL.QB	Rd, CONST8	$R_D = \text{CONST8} \parallel \text{CONST8} \parallel \text{CONST8} \parallel \text{CONST8}$	
REPLV.QB	Rd, Rs	$R_D = R_{S_{7,0}} \parallel R_{S_{7,0}} \parallel R_{S_{7,0}} \parallel R_{S_{7,0}}$	
SUBU.QB	Rd, Rs, Rt	$R_{DXY} = R_{SXY}^{\ominus} - R_{TXY}^{\ominus}$	$XY \in \{ LL, LR, RL, RR \}$
SUBU_S.QB	Rd, Rs, Rt	$R_{DXY} = \llbracket R_{SXY}^{\ominus} - R_{TXY}^{\ominus} \rrbracket$	$XY \in \{ LL, LR, RL, RR \}$
SUBUH.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} - R_{TXY}^{\ominus}) >> 1$	$XY \in \{ LL, LR, RL, RR \}$
SUBUH_R.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} - R_{TXY}^{\ominus} + 1^{\ominus}) >> 1$	$XY \in \{ LL, LR, RL, RR \}$

SHIFT OPERATIONS: 8-BIT DATA			
SHLL.QB	Rd, Rs, SHIFT3	$R_{DXY} = R_{SXY} << \text{SHIFT3}$	$XY \in \{ LL, LR, RL, RR \}$
SHLLV.QB	Rd, Rs, Rt	$R_{DXY} = R_{SXY} << R_{T_{2,0}}$	$XY \in \{ LL, LR, RL, RR \}$
SHRA.QB ^{R2}	Rd, Rs, SHIFT3	$R_{DXY} = R_{SXY}^{\pm} >> \text{SHIFT3}$	$XY \in \{ LL, LR, RL, RR \}$
SHRA_R.QB ^{R2}	Rd, Rs, SHIFT3	$R_{DXY} = \textcircled{R}_{SXY}^{\pm} >> \text{SHIFT3}$	$XY \in \{ LL, LR, RL, RR \}$
SHRAV.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = R_{SXY}^{\pm} >> R_{T_{2,0}}$	$XY \in \{ LL, LR, RL, RR \}$
SHRAV_R.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = \textcircled{R}_{SXY}^{\pm} >> R_{T_{2,0}}$	$XY \in \{ LL, LR, RL, RR \}$
SHRL.QB	Rd, Rs, SHIFT3	$R_{DXY} = R_{SXY}^{\ominus} >> \text{SHIFT3}$	$XY \in \{ LL, LR, RL, RR \}$
SHRLV.QB	Rd, Rs, Rt	$R_{DXY} = R_{SXY}^{\ominus} >> R_{T_{2,0}}$	$XY \in \{ LL, LR, RL, RR \}$

UNSIGNED INTEGER MULTIPLY OPERATIONS: 8-BIT/16-BIT DATA » GPR			
MULEU_S.PH.QBL	Rd, Rs, Rt	$R_D = \llbracket R_{S_{LL}}^{\ominus} \times R_{T_{RL}}^{\ominus} \rrbracket \parallel \llbracket R_{S_{LR}}^{\ominus} \times R_{T_{RR}}^{\ominus} \rrbracket$	
MULEU_S.PH.QBR	Rd, Rs, Rt	$R_D = \llbracket R_{S_{RL}}^{\ominus} \times R_{T_{LL}}^{\ominus} \rrbracket \parallel \llbracket R_{S_{RR}}^{\ominus} \times R_{T_{RR}}^{\ominus} \rrbracket$	

UNSIGNED INTEGER MULTIPLY OPERATIONS: 8-BIT DATA » ACCUMULATOR			
DPAU.H.QBL	Ac, Rs, Rt	$AC += (R_{S_{LL}}^{\ominus} \times R_{T_{LL}}^{\ominus}) + (R_{S_{LR}}^{\ominus} \times R_{T_{LR}}^{\ominus})$	
DPAU.H.QBR	Ac, Rs, Rt	$AC += (R_{S_{RL}}^{\ominus} \times R_{T_{RL}}^{\ominus}) + (R_{S_{RR}}^{\ominus} \times R_{T_{RR}}^{\ominus})$	
DPSU.H.QBL	Ac, Rs, Rt	$AC -= (R_{S_{LL}}^{\ominus} \times R_{T_{LL}}^{\ominus}) + (R_{S_{LR}}^{\ominus} \times R_{T_{LR}}^{\ominus})$	
DPSU.H.QBR	Ac, Rs, Rt	$AC -= (R_{S_{RL}}^{\ominus} \times R_{T_{RL}}^{\ominus}) + (R_{S_{RR}}^{\ominus} \times R_{T_{RR}}^{\ominus})$	

PRECISION EXPANSION (DATA UNPACKING) OPERATIONS: 8-BIT DATA			
PRECEQU.PH.QBL	Rd, Rs	$R_D = (R_{S_{LL}} << 7) \parallel (R_{S_{LR}} << 7)$	
PRECEQU.PH.QBLA	Rd, Rs	$R_D = (R_{S_{LL}} << 7) \parallel (R_{S_{RL}} << 7)$	
PRECEQU.PH.QBR	Rd, Rs	$R_D = (R_{S_{RL}} << 7) \parallel (R_{S_{RR}} << 7)$	
PRECEQU.PH.QBRA	Rd, Rs	$R_D = (R_{S_{LR}} << 7) \parallel (R_{S_{RR}} << 7)$	
PRECEU.PH.QBL	Rd, Rs	$R_D = 0^8 \parallel R_{S_{LL}} \parallel 0^8 \parallel R_{S_{LR}}$	
PRECEU.PH.QBLA	Rd, Rs	$R_D = 0^8 \parallel R_{S_{LL}} \parallel 0^8 \parallel R_{S_{RL}}$	
PRECEU.PH.QBR	Rd, Rs	$R_D = 0^8 \parallel R_{S_{RL}} \parallel 0^8 \parallel R_{S_{RR}}$	
PRECEU.PH.QBRA	Rd, Rs	$R_D = 0^8 \parallel R_{S_{LR}} \parallel 0^8 \parallel R_{S_{RR}}$	



COMPARE AND PICK OPERATIONS: 8-BIT DATA			
CMPGDU.EQ.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = CC_{XY} = (R_{SXY}^{\ominus} = R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGDU.LT.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = CC_{XY} = (R_{SXY}^{\ominus} < R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGDU.LE.QB ^{R2}	Rd, Rs, Rt	$R_{DXY} = CC_{XY} = (R_{SXY}^{\ominus} \leq R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGU.EQ.QB	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} = R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGU.LT.QB	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} < R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPGU.LE.QB	Rd, Rs, Rt	$R_{DXY} = (R_{SXY}^{\ominus} \leq R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPU.EQ.QB	Rd, Rs, Rt	$CC_{XY} = (R_{SXY}^{\ominus} = R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPU.LT.QB	Rd, Rs, Rt	$CC_{XY} = (R_{SXY}^{\ominus} < R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
CMPU.LE.QB	Rd, Rs, Rt	$CC_{XY} = (R_{SXY}^{\ominus} \leq R_{TXY}^{\ominus}) ? 1 : 0$	$XY \in \{ LL, LR, RL, RR \}$
PICK.QB	Rd, Rs, Rt	$R_{DXY} = CC_{XY} ? R_{SXY} : R_{TXY}$	$XY \in \{ LL, LR, RL, RR \}$



COMPARE AND PICK OPERATIONS: 16-BIT DATA			
CMP.EQ.PH	Rd, Rs, Rt	$CC = ((R_{S_L}^{\pm} = R_{T_L}^{\pm}) ? 1 : 0) \parallel ((R_{S_R}^{\pm} = R_{T_R}^{\pm}) ? 1 : 0)$	
CMP.LT.PH	Rd, Rs, Rt	$CC = ((R_{S_L}^{\pm} < R_{T_L}^{\pm}) ? 1 : 0) \parallel ((R_{S_R}^{\pm} < R_{T_R}^{\pm}) ? 1 : 0)$	
CMP.LE.PH	Rd, Rs, Rt	$CC = ((R_{S_L}^{\pm} \leq R_{T_L}^{\pm}) ? 1 : 0) \parallel ((R_{S_R}^{\pm} \leq R_{T_R}^{\pm}) ? 1 : 0)$	
PICK.PH	Rd, Rs, Rt	$R_D = (CC_L ? R_{S_L} : R_{T_L}) \parallel (CC_R ? R_{S_R} : R_{T_R})$	



ARITHMETIC OPERATIONS: 16-BIT DATA		
ABSQ_S.PH	R _D , R _S	$R_D = [R_{SL}^{\pm}] \parallel [R_{SR}^{\pm}]$
ADDQ.PH	R _D , R _S , R _T	$R_D = (R_{SL}^{\pm} + R_{TL}^{\pm}) \parallel (R_{SR}^{\pm} + R_{TR}^{\pm})$
ADDQ_S.PH	R _D , R _S , R _T	$R_D = [R_{SL}^{\pm} + R_{TL}^{\pm}] \parallel [R_{SR}^{\pm} + R_{TR}^{\pm}]$
ADDQH.PH ^{R2}	R _D , R _S , R _T	$R_D = ((R_{SL}^{\pm} + R_{TL}^{\pm}) >> 1) \parallel ((R_{SR}^{\pm} + R_{TR}^{\pm}) >> 1)$
ADDQH_R.PH ^{R2}	R _D , R _S , R _T	$R_D = @((R_{SL}^{\pm} + R_{TL}^{\pm}) >> 1) \parallel @((R_{SR}^{\pm} + R_{TR}^{\pm}) >> 1)$
ADDU.PH ^{R2}	R _D , R _S , R _T	$R_D = (R_{SL}^{\odot} + R_{TL}^{\odot}) \parallel (R_{SR}^{\odot} + R_{TR}^{\odot})$
ADDU_S.PH ^{R2}	R _D , R _S , R _T	$R_D = [R_{SL}^{\odot} + R_{TL}^{\odot}] \parallel [R_{SR}^{\odot} + R_{TR}^{\odot}]$
REPL.PH	R _D , CONST10	$R_D = \text{CONST10}^{\pm} \parallel \text{CONST10}^{\pm}$
REPLV.PH	R _D , R _S	$R_D = R_{S15:0} \parallel R_{S15:0}$
SUBQ.PH	R _D , R _S , R _T	$R_D = (R_{SL}^{\pm} - R_{TL}^{\pm}) \parallel (R_{SR}^{\pm} - R_{TR}^{\pm})$
SUBQ_S.PH	R _D , R _S , R _T	$R_D = [R_{SL}^{\pm} - R_{TL}^{\pm}] \parallel [R_{SR}^{\pm} - R_{TR}^{\pm}]$
SUBQH.PH ^{R2}	R _D , R _S , R _T	$R_D = ((R_{SL}^{\pm} - R_{TL}^{\pm}) >> 1) \parallel ((R_{SR}^{\pm} - R_{TR}^{\pm}) >> 1)$
SUBQH_R.PH ^{R2}	R _D , R _S , R _T	$R_D = @((R_{SL}^{\pm} - R_{TL}^{\pm}) >> 1) \parallel @((R_{SR}^{\pm} - R_{TR}^{\pm}) >> 1)$
SUBU.PH ^{R2}	R _D , R _S , R _T	$R_D = (R_{SL}^{\odot} - R_{TL}^{\odot}) \parallel (R_{SR}^{\odot} - R_{TR}^{\odot})$
SUBU_S.PH ^{R2}	R _D , R _S , R _T	$R_D = [R_{SL}^{\odot} - R_{TL}^{\odot}] \parallel [R_{SR}^{\odot} - R_{TR}^{\odot}]$



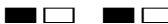

SHIFT OPERATIONS: 16-BIT DATA		
SHLL.PH	R _D , R _S , SHIFT4	$R_D = (R_{SL} << \text{SHIFT4}) \parallel (R_{SR} << \text{SHIFT4})$
SHLL_S.PH	R _D , R _S , SHIFT4	$R_D = [R_{SL} << \text{SHIFT4}] \parallel [R_{SR} << \text{SHIFT4}]$
SHLLV.PH	R _D , R _S , R _T	$R_D = (R_{SL} << R_{T3:0}) \parallel (R_{SR} << R_{T3:0})$
SHLLV_S.PH	R _D , R _S , R _T	$R_D = [R_{SL} << R_{T3:0}] \parallel [R_{SR} << R_{T3:0}]$
SHRA.PH	R _D , R _S , SHIFT4	$R_D = (R_{SL}^{\pm} >> \text{SHIFT4}) \parallel (R_{SR}^{\pm} >> \text{SHIFT4})$
SHRA_R.PH	R _D , R _S , SHIFT4	$R_D = @ (R_{SL}^{\pm} >> \text{SHIFT4}) \parallel @ (R_{SR}^{\pm} >> \text{SHIFT4})$
SHRAV.PH	R _D , R _S , R _T	$R_D = (R_{SL}^{\pm} >> R_{T3:0}) \parallel (R_{SR}^{\pm} >> R_{T3:0})$
SHRAV_R.PH	R _D , R _S , R _T	$R_D = @ (R_{SL}^{\pm} >> R_{T3:0}) \parallel @ (R_{SR}^{\pm} >> R_{T3:0})$
SHRL.PH ^{R2}	R _D , R _S , SHIFT4	$R_D = (R_{SL}^{\odot} >> \text{SHIFT4}) \parallel (R_{SR}^{\odot} >> \text{SHIFT4})$
SHRLV.PH ^{R2}	R _D , R _S , R _T	$R_D = (R_{SL}^{\odot} >> R_{T3:0}) \parallel (R_{SR}^{\odot} >> R_{T3:0})$








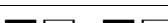
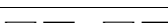
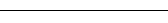
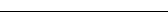
PRECISION REDUCTION (DATA PACKING) OPERATIONS: 16-BIT DATA		
PRECR.QB.PH ^{R2}	R _D , R _S , R _T	$R_D = R_{SLR} \parallel R_{SRR} \parallel R_{TLR} \parallel R_{TRR}$ 
PRECRQ.QB.PH	R _D , R _S , R _T	$R_D = R_{SLL} \parallel R_{SRL} \parallel R_{TLL} \parallel R_{TRL}$ 
PRECRQU_S.QB.PH	R _D , R _S , R _T	$R_D = [R_{SL}^{\odot}]_{14:7} \parallel [R_{SR}^{\odot}]_{14:7} \parallel [R_{TL}^{\odot}]_{14:7} \parallel [R_{TR}^{\odot}]_{14:7}$

PRECISION EXPANSION (DATA UNPACKING) OPERATIONS: 16-BIT DATA		
PRECEQ.W.PHL	R _D , R _S	$R_D = R_{SL} << 16$ 
PRECEQ.W.PHR	R _D , R _S	$R_D = R_{SR} << 16$ 

INTEGER MULTIPLY OPERATIONS: 16-BIT DATA » ACCUMULATOR		
DPA.W.PH ^{R2}	A _C , R _S , R _T	$A_C += (R_{SL}^{\pm} \times R_{TL}^{\pm}) + (R_{SR}^{\pm} \times R_{TR}^{\pm})$
DPAX.W.PH ^{R2}	A _C , R _S , R _T	$A_C += (R_{SL}^{\pm} \times R_{TR}^{\pm}) + (R_{SR}^{\pm} \times R_{TL}^{\pm})$
DPS.W.PH ^{R2}	A _C , R _S , R _T	$A_C -= (R_{SL}^{\pm} \times R_{TL}^{\pm}) + (R_{SR}^{\pm} \times R_{TR}^{\pm})$
DPSX.W.PH ^{R2}	A _C , R _S , R _T	$A_C -= (R_{SL}^{\pm} \times R_{TR}^{\pm}) + (R_{SR}^{\pm} \times R_{TL}^{\pm})$
MULSA.W.PH ^{R2}	A _C , R _S , R _T	$A_C += (R_{SL}^{\pm} \times R_{TL}^{\pm}) - (R_{SR}^{\pm} \times R_{TR}^{\pm})$

INTEGER MULTIPLY OPERATIONS: 16-BIT DATA » GPR		
MUL.PH ^{R2}	R _D , R _S , R _T	$R_D = (R_{SL}^{\pm} \times R_{TL}^{\pm})_R \parallel (R_{SR}^{\pm} \times R_{TR}^{\pm})_R$
MUL_S.PH ^{R2}	R _D , R _S , R _T	$R_D = [R_{SL}^{\pm} \times R_{TL}^{\pm}]_R \parallel [R_{SR}^{\pm} \times R_{TR}^{\pm}]_R$

FRACTIONAL MULTIPLY OPERATIONS: 16-BIT DATA » GPR		
MULQ_RS.PH	R _D , R _S , R _T	$R_D = [R_{SL}^{\pm} \odot R_{TL}^{\pm}]_L \parallel [R_{SR}^{\pm} \odot R_{TR}^{\pm}]_L$ 
MULQ_S.PH ^{R2}	R _D , R _S , R _T	$R_D = [R_{SL}^{\pm} \bullet R_{TL}^{\pm}]_L \parallel [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]_L$ 
MULEQ_S.W.PHL	R _D , R _S , R _T	$R_D = [R_{SL}^{\pm} \bullet R_{TL}^{\pm}]$ 
MULEQ_S.W.PHR	R _D , R _S , R _T	$R_D = [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]$ 

FRACTIONAL MULTIPLY OPERATIONS: 16-BIT DATA » ACCUMULATOR		
DPAQ_S.W.PH	A _C , R _S , R _T	$A_C += [R_{SL}^{\pm} \bullet R_{TL}^{\pm}] + [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]$ 
DPAQX_S.W.PH ^{R2}	A _C , R _S , R _T	$A_C += [R_{SL}^{\pm} \bullet R_{TR}^{\pm}] + [R_{SR}^{\pm} \bullet R_{TL}^{\pm}]$ 
DPAQX_SA.W.PH ^{R2}	A _C , R _S , R _T	$A_C = [A_C + [R_{SL}^{\pm} \bullet R_{TR}^{\pm}] + [R_{SR}^{\pm} \bullet R_{TL}^{\pm}]]$ 
DPSQ_S.W.PH	A _C , R _S , R _T	$A_C -= [R_{SL}^{\pm} \bullet R_{TL}^{\pm}] + [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]$ 
DPSQX_S.W.PH ^{R2}	A _C , R _S , R _T	$A_C -= [R_{SL}^{\pm} \bullet R_{TR}^{\pm}] + [R_{SR}^{\pm} \bullet R_{TL}^{\pm}]$ 
DPSQX_SA.W.PH ^{R2}	A _C , R _S , R _T	$A_C = [A_C - [R_{SL}^{\pm} \bullet R_{TR}^{\pm}] + [R_{SR}^{\pm} \bullet R_{TL}^{\pm}]]$ 
MULSAQ_S.W.PH	A _C , R _S , R _T	$A_C += [R_{SL}^{\pm} \bullet R_{TL}^{\pm}] - [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]$ 
MAQ_S.W.PHL	A _C , R _S , R _T	$A_C += [R_{SL}^{\pm} \bullet R_{TL}^{\pm}]$ 
MAQ_S.W.PHR	A _C , R _S , R _T	$A_C += [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]$ 
MAQ_SA.W.PHL	A _C , R _S , R _T	$A_C = [A_C^+ + [R_{SL}^{\pm} \bullet R_{TL}^{\pm}]]$ 
MAQ_SA.W.PHR	A _C , R _S , R _T	$A_C = [A_C^+ + [R_{SR}^{\pm} \bullet R_{TR}^{\pm}]]$ 

ACCUMULATOR EXTRACT OPERATIONS: 16-BIT DATA		
EXTR_S.H	R _D , A _C , SHIFT5	$R_D = [A_C >> \text{SHIFT5}]_R$
EXTRV_S.H	R _D , A _C , R _T	$R_D = [A_C >> R_{T4:0}]_R$

DSP CONTROL REGISTER ACCESS OPERATIONS		
RDDSP	R _D , MASK10	$R_D = \text{READDSPCONTROL}(\text{MASK10}_{5:0})$
WRDSP	R _S , MASK10	$\text{WRITEDSPCONTROL}(\text{MASK10}_{5:0}, R_S)$

Arithmetic Operations: 32-bit Data		
ABSQ_S.W	R _D , R _S	R _D = [R _S]
ADDQ_S.W	R _D , R _S , R _T	R _D = [R _S [±] + R _T [±]]
ADDQH.W ^{R2}	R _D , R _S , R _T	R _D = (R _S [±] + R _T [±]) >> 1
ADDQH_R.W ^{R2}	R _D , R _S , R _T	R _D = ®((R _S [±] + R _T [±]) >> 1)
ADDSC	R _D , R _S , R _T	C::R _D = R _S [∅] + R _T [∅]
ADDWC	R _D , R _S , R _T	R _D = R _S [∅] + R _T [∅] + C
SUBQ_S.W	R _D , R _S , R _T	R _D = [R _S − R _T]
SUBQH.W ^{R2}	R _D , R _S , R _T	R _D = (R _S [±] − R _T [±]) >> 1
SUBQH_R.W ^{R2}	R _D , R _S , R _T	R _D = ®((R _S [±] − R _T [±]) >> 1)

Shift Operations: 32-bit Data		
SHLL_S.W	R _D , R _S , SHIFT5	R _D = [R _S << SHIFT5]
SHLLV_S.W	R _D , R _S , R _T	R _D = [R _S << R _{T4:0}]
SHRA_R.W	R _D , R _S , SHIFT5	R _D = ®(R _S [±] >> SHIFT5)
SHRAV_R.W	R _D , R _S , R _T	R _D = ®(R _S [±] >> R _{T4:0})

Fractional Multiply Operations: 32-bit Data » GPR		
MULQ_RS.W ^{R2}	R _D , R _S , R _T	R _D = [R _S [±] ⊙ R _T [±]] _L
MULQ_S.W ^{R2}	R _D , R _S , R _T	R _D = [R _S [±] • R _T [±]] _L

Fractional Multiply Operations: 32-bit Data » Accumulator		
DPAQ_SA.L.W	Ac, R _S , R _T	Ac = [Ac [±] + [R _S [±] • R _T [±]]]
DPSQ_SA.L.W	Ac, R _S , R _T	Ac = [Ac [±] − [R _S [±] • R _T [±]]]

Integer Multiply Operations: 32-bit Data » Accumulator		
MADD ^{R2}	Ac, R _S , R _T	Ac += R _S [±] × R _T [±]
MADDU ^{R2}	Ac, R _S , R _T	Ac += R _S [∅] × R _T [∅]
MSUB ^{R2}	Ac, R _S , R _T	Ac -= R _S [±] × R _T [±]
MSUBU ^{R2}	Ac, R _S , R _T	Ac -= R _S [∅] × R _T [∅]
MULT ^{R2}	Ac, R _S , R _T	Ac = R _S [±] × R _T [±]
MULTU ^{R2}	Ac, R _S , R _T	Ac = R _S [∅] × R _T [∅]

Precision Reduction (Data Packing) Operations: 32-bit Data		
PRECRQ.PH.W	R _D , R _S , R _T	R _D = R _{SL} R _{TL} <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> <div>↔</div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div>
PRECRQ_RS.PH.W	R _D , R _S , R _T	R _D = ®[R _S [±]] _L ®[R _T [±]] _L <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div>
PRECR_SRA.PH.W ^{R2}	R _D , R _S , SHIFT5	R _D = (R _D [±] >> SHIFT5) _R (R _S [±] >> SHIFT5) _R
PRECR_SRA_R.PH.W ^{R2}	R _D , R _S , SHIFT5	R _D = ®(R _D [±] >> SHIFT5) _R ®(R _S [±] >> SHIFT5) _R

Accumulator Extract Operations: 32-bit Data		
EXTR.W	R _D , Ac, SHIFT5	R _D = (Ac >> SHIFT5) _R
EXTR_R.W	R _D , Ac, SHIFT5	R _D = ®(Ac >> SHIFT5) _R
EXTR_RS.W	R _D , Ac, SHIFT5	R _D = ®[Ac >> SHIFT5] _R
EXTRV.W	R _D , Ac, R _T	R _D = (Ac >> R _{T4:0}) _R
EXTRV_R.W	R _D , Ac, R _T	R _D = ®(Ac >> R _{T4:0}) _R
EXTRV_RS.W	R _D , Ac, R _T	R _D = ®[Ac >> R _{T4:0}] _R

Bit-Field Extract Operations		
BPOSGE32	OFF18	IF POS ≥ 32, PC += OFF18 [±]
EXTP	R _D , Ac, SIZE5	R _D = Ac _{POS:POS-SIZE5}
EXTPDP	R _D , Ac, SIZE5	R _D = Ac _{POS:POS-SIZE5} , POS -= (SIZE5 + 1)
EXTPV	R _D , Ac, R _S	R _D = Ac _{POS:POS-RS4:0}
EXTPDPV	R _D , Ac, R _S	R _D = Ac _{POS:POS-RS4:0} , POS -= (RS _{4:0} + 1)
MTHLIP	Rs, Ac	Ac _{HI} = Ac _{LO} , Ac _{LO} = R _S , POS += 32

Load Data Operations		
BITREV	R _D , R _S	R _D = R _{S0:15}
LBUX	R _D , R _S (R _T)	R _D = MEM8(R _S + R _T) [∅]
LHX	R _D , R _S (R _T)	R _D = MEM16(R _S + R _T) [±]
LWX	R _D , R _S (R _T)	R _D = MEM32(R _S + R _T)
MODSUB	R _D , R _S , R _T	R _D = (R _S ≠ 0) ? (R _S − R _{T7:0}) : R _{T23:8}

Accumulator Operations		
MFHI	R _D , Ac	R _D = Ac _{HI}
MFLO	R _D , Ac	R _D = Ac _{LO}
MTHI	Rs, Ac	Ac _{HI} = R _S
MTLO	Rs, Ac	Ac _{LO} = R _S
SHILO	Ac, SHIFT6	Ac = (SHIFT6 [±] ≥ 0) ? (Ac >> SHIFT6 [±]) : (Ac << −SHIFT6 [±])
SHILOV	Ac, R _S	Ac = (R _{S5:0} [±] ≥ 0) ? (Ac >> R _{S5:0} [±]) : (Ac << −R _{S5:0} [±])

Bit-Field Operations		
APPEND ^{R2}	R _D , R _S , SHIFT5	R _D = (R _D << SHIFT5) :: R _S SHIFT5-1:0
BALIGN ^{R2}	R _D , R _S , BPOS2	R _D = (R _D << (8 × BPOS2)) :: (R _S >> (32 − 8 × BPOS2))
INSV	R _D , R _S	R _D POS+SIZE-1:POS = R _S SIZE-1:0
PACKRL.PH	R _D , R _S , R _T	R _D = R _{SR} R _{TL} <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> <div>↔</div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div>
PREPEND ^{R2}	R _D , R _S , SHIFT5	R _D = R _S SHIFT5-1:0 :: (R _D >> SHIFT5)