DLX Instruction Set

Complete list of the instructions in DLX.

Instruction meaning	T ()	Complete list of the instructions in DLA.
Institute	Instruction type/opcode	
LH, LHU, SH Load halfword, load halfword unsigned, store halfword LW, SW Load word, store word (to/from integer registers) LF, LD, SF, SD Load SP float, load DP float, store SP float, store DP float (SP - single precision, DP - double precision) MOV12S, MOVS21 Move from/to GPR to/from a special register MOVF, MOVD Copy one floating-point register or a DP pair to another register or pair MOVF2I, MOV12FP Move 32 bits from/to FP tegister to/from integer registers Arithmetic / Logical Operations on integer or logical data in GPRs; signed arithmetics trap on overflow ADD, ADDI, ADDU, ADDU, ADDU, ADDU, ADDU, ADDU, ADDU, SUB, SUBI,	Data transfers	
LW, SW Load word, store word (to/from integer registers) LF, LD, SF, SD Load SP float, load DP float, store SP float, store DP float (SP - single precision, DP - double precision) MOVIZS, MOVS21 Move from/to GPR to/from a special register MOVF, MOVD Copy one floating-point register or a DP pair to another register or pair MOVF2I, MOV12PP Move 32 bits from/to FP tegister to/from integer registers Arithmetic / Logical Operations on integer or logical data in GPRs; signed arithmetics trap on overflow ADD, ADDI, ADDU, ADDU, ADDU, ADDUI Subtract, subtract immediate; signed and unsigned SUB, SUBI, SUBI, SUBU, SUBI, SUBU, SUB, SUBI, SRA, SLIL, SRA	LB, LBU, SB	Load byte, load byte unsigned, store byte
LF, LD, SF, SD Load SP float, load DP float, store SP float, store DP float (SP - single precision, DP - double precision) MOVI2S, MOVS2I Move from/to GPR to/from a special register MOVF, MOVD Copy one floating-point register or a DP pair to another register or pair MOVFP2I, MOV12FP Move 32 bits from/to FP tegister to/from integer registers Arithmetic / Logical Operations on integer or logical data in GPRs; signed arithmetics trap on overflow ADD, ADDI, ADDU, ADDU, ADDU, ADDUI SUB, SUBI,	LH, LHU, SH	Load halfword, load halfword unsigned, store halfword
MOVI2S, MOVS21 Move from/to GPR to/from a special register MOVF, MOVD Copy one floating-point register or a DP pair to another register or pair MOVFP2I, MOVI2FP Move 32 bits from/to FP tegister to/from integer registers Arithmetic / Logical ADD, ADDI, ADDU, ADDU, ADDU, ADDU, ADDU, ADDUI SUB, SUBI, SUBIF, SUBID, SUBF MULTD, MULTF Multiply DP, SP floating point MULTD, MULTF Multiply DP, SP floating point CONVETDLE, CVTIZE,	LW, SW	Load word, store word (to/from integer registers)
MOVF, MOVD Copy one floating-point register or a DP pair to another register or pair MOVFP2I, MOV12FP Arithmetic / Logical ADD, ADDI, ADDU, ADDI, ADDU, ADDI, SUBI,	LF, LD, SF, SD	Load SP float, load DP float, store SP float, store DP float (SP - single precision, DP - double precision)
MOVFP2I, MOV12FP Move 32 bits from/to FP tegister to/from integer registers Arithmetic / Logical Operations on integer or logical data in GPRs; signed arithmetics trap on overflow ADD, ADDI, ADDI, ADDU, Add, add immediate (all immediates are 16-bits); signed and unsigned SUB, SUBI, SUB, SUB, SUB, SUB, SUB, SUB, SUB, SUB	MOVI2S, MOVS2I	Move from/to GPR to/from a special register
Arithmetic / Logical Operations on integer or logical data in GPRs; signed arithmetics trap on overflow ADD, ADDI, ADDU, ADDU, ADDUI ADDUI Add, add immediate (all immediates are 16-bits); signed and unsigned SUB, SUBI, SUB, SUB, SUB, SUB, SUB, SUB, SUB, SUB	MOVF, MOVD	Copy one floating-point register or a DP pair to another register or pair
ADD, ADDI, ADDU SUB, SUBI, SUBU, SUBU, SUBU, SUBU, SUBU, SUBU, SUBU, SUBU, SUBI, SUBU, SUBI, SUBU, SUBI, SUBU, SUBI, SUBU, SUB	MOVFP2I, MOVI2FP	Move 32 bits from/to FP tegister to/from integer registers
ADDUI SUB, SUBI, SUBI, SUBI, SUBU, SUBU, SUBUI MULT, MULTU, Divided 32-bit values AND, ANDI And, and immediate (ali mimediate; signed and unsigned) Multiply and divide, signed and unsigned; operands must be floating-point registers; all operations take a yield 32-bit values AND, ANDI And, and immediate OR, ORI, XOP, XOPI LHI Load high immediate - loads upper half of register with immediate SLL, SRL, SRA, SLLI, SRLI, SRAI S., S. I Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF COVTP21, CVT127, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul cVTD21, CVT121, CVT	Arithmetic / Logical	Operations on integer or logical data in GPRs; signed arithmetics trap on overflow
Subtract, subtract immediate, signed and unsigned MULT, MULTU, DIVU Multiply and divide, signed and unsigned; operands must be floating-point registers; all operations take a yield 32-bit values AND, ANDI And immediate OR, ORI, XOP, XOPI Or, or immediate, exclusive or, exclusive or immediate LHI Load high immediate - loads upper half of register with immediate SLL, SRA, SRA, SLLI, SRAI Shifts: both immediate(S_I) and variable form(S_); shifts are shift left logical, right logical, right ariths S_, S_I Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) of a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF21, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.		Add, add immediate (all immediates are 16-bits); signed and unsigned
DIV, DIVU yield 32-bit values AND, ANDI And, and immediate OR, ORI, XOP, XOPI LHI Load high immediate, exclusive or, exclusive or immediate SLL, SRL, SRA, SLLI, SRAI Shifts: both immediate(S_I) and variable form(S_); shifts are shift left logical, right logical, right ariths S_, S_I Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Subtract DP, SP numbers MULTD, MULTF Divide DP, SP floating point CVTF2D, CVTF21, CONVET15, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.		Subtract, subtract immediate; signed and unsigned
OR, ORI, XOP, XOPI Or, or immediate, exclusive or, exclusive or immediate LHI Load high immediate - loads upper half of register with immediate SLL, SRL, SRA, SLLI, SRLI, SRAI Shifts: both immediate(S_I) and variable form(S_); shifts are shift left logical, right logical, right aritht S_, S_I Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF SUBD, SUBF Subtract DP, SP floating point MULTD, MULTF DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF21, CVTP2T, CVTD2T, CV		Multiply and divide, signed and unsigned; operands must be floating-point registers; all operations take and yield 32-bit values
LHI Load high immediate - loads upper half of register with immediate SLL, SRL, SRA, SLLI, SRAI Shifts: both immediate(S_I) and variable form(S_); shifts are shift left logical, right logical, right ariths S_, S_I Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTF2I, COVTF2I, COVTD2F, COVTD2F, COVTD2F, CVTD2F,	AND, ANDI	And, and immediate
SLL, SRL, SRA, SLLI, SRAI Shifts: both immediate(S_I) and variable form(S_); shifts are shift left logical, right arithn S_, S_I Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF21, CONTEST, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.	OR, ORI, XOP, XOPI	Or, or immediate, exclusive or, exclusive or immediate
SLLI, SRLI, SRAI Set conditional: "_"may be LT, GT, LE, GE, EQ, NE Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J. JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, CVTD2F, CVTD2F, CVTD2I, CVTI2F, CVTP2D, CVTI2F, CVTP2D, CVTI2F, CVTP2D, CVTI2F, CVTP2D, CVTP3D,	LHI	Load high immediate - loads upper half of register with immediate
Control Conditional branches and jumps; PC-relative or through register BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF21, CVTD2F, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.		Shifts: both immediate(S_I) and variable form(S_); shifts are shift left logical, right logical, right arithmetic
BEQZ, BNEZ Branch GPR equal/not equal to zero; 16-bit offset from PC BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, CVTD2F, CVTD2F, CVTD2I, CVTI2F, CVTD2I, CVTI2F, CVTD2I, CVTI2F, CVTG2I, CVTI2F, CVTG2I, CVTI2F, CVTD2I, CVTI2F, CVTD2I, CVTI2F, CVTG2I, CVTI2F, CVTC2DE, CVTC3DE, CV	S, SI	Set conditional: ""may be LT, GT, LE, GE, EQ, NE
BFPT, BFPF Test comparison bit in the FP status register and branch; 16-bit offset from PC J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.	Control	Conditional branches and jumps; PC-relative or through register
J, JR Jumps: 26-bit offset from PC(J) or target in register(JR) JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.	BEQZ, BNEZ	Branch GPR equal/not equal to zero; 16-bit offset from PC
JAL, JALR Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR) TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, COnvert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul precision), or F(Single precision). Both operands are in the FP registers.	BFPT, BFPF	Test comparison bit in the FP status register and branch; 16-bit offset from PC
TRAP Transfer to operating system at a vectored address RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul CVTD2I, CVTI2F, precision), or F(Single precision). Both operands are in the FP registers.	J, JR	Jumps: 26-bit offset from PC(J) or target in register(JR)
RFE Return to user code from an exception; restore user code Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Double CVTD2I, CVTI2F, CVTI2F, CVTI2F, Description). Both operands are in the FP registers.	JAL, JALR	Jump and link: save PC+4 to R31, target is PC-relative(JAL) ot a register(JALR)
Floating point Floating-point operations on DP and SP formats ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, COnvert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul CVTD2I, CVTI2F, precision), or F(Single precision). Both operands are in the FP registers.	TRAP	Transfer to operating system at a vectored address
ADDD, ADDF Add DP, SP numbers SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, CVTD2F, CVTD2I, CVTI2F, CVTD2	RFE	Return to user code from an exception; restore user code
SUBD, SUBF Subtract DP, SP numbers MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, CVTD2F, CVTD2I, CVTI2F, C	Floating point	Floating-point operations on DP and SP formats
MULTD, MULTF Multiply DP, SP floating point DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, CVTD2F, CVTD2I, CVTI2F, Divide DP, SP floating point Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Double precision), or F(Single precision). Both operands are in the FP registers.	ADDD, ADDF	Add DP, SP numbers
DIVD, DIVF Divide DP, SP floating point CVTF2D, CVTF2I, CVTD2F, CVTD2I, CVTI2F, CVTD2I, CVTI	SUBD, SUBF	Subtract DP, SP numbers
CVTF2D, CVTF2I, CVTD2F, Convert instructions: CVTx2y converts from type x to type y, where x and y are one of I(Integer), D(Doul CVTD2I, CVTI2F, precision), or F(Single precision). Both operands are in the FP registers.	MULTD, MULTF	Multiply DP, SP floating point
CVTD2F, CVTD2I, CVTI2F, CVTI2F, CVTI2F, CVTD2I, CVTD2I, CVTD2I, CVTD2I, CVTI2F, CVTI2F	DIVD, DIVF	Divide DP, SP floating point
<u> </u>	CVTD2F,	Convert instructions: $CVTx2y$ converts from type x to type y , where x and y are one of I(Integer), D(Double precision), or F(Single precision). Both operands are in the FP registers.
D,F DP and SP compares: "" may be LT, GT, LE, GE, EQ, NE; set comparison bit in FP status register.	D,F	DP and SP compares: "" may be LT, GT, LE, GE, EQ, NE; set comparison bit in FP status register.