DL-305 Instruction Set

Boolean Instructions

Store (STR)
Begins a new rung or an additional branch in a rung with a normally open contact.

Store Not (STRN)
Begins a new rung or an additional branch in a rung with a normally closed contact.

Or (OR)
Logically ORs a normally open contact in parallel with another contact in a rung.

Or not (OR NOT)
Logically ORs a normally closed contact in parallel with another contact in a rung.

And (AND)
Logically ANDs a normally open contact in series with another contact in a rung.

And not (ANDN)
Logically ANDs a normally closed contact in series with another contact in a rung.

And store (AND STR)
Logically ANDs two branches in a rung in series.

Or store (OR STR)
Logically ORs two branches of a rung in parallel.

Out (OUT)
Reflects the status of the rung (ON/OFF) and outputs the discrete (ON/OFF) state to the specified image register.

Reset (RST)
Resets or turns OFF an output or resets a counter.

Set out (SET OUT)
Reflects the status of the rung (ON/OFF) and outputs the discrete (ON/OFF) state to the specified image register.

Set out reset (SET OUT RST)
Typically known as a one shot, when the input logic produces an OFF to ON transition the output will turn ON for one CPU scan.

Client control set (MCS)/Client control reset (MCR)
The Client control set and Client Control Reset instructions are used to provide an additional left power rail which is controllable by an input contact. This is sometimes known as a sub power rail. Any number of rungs of ladder logic can be disabled using these instructions.

Accumulator Load and Output Instructions

Data store (F50)
Loads the value of a 16-bit register, two consecutive 8-bit registers, or a 4-digit BCD value into the accumulator.

Data store 1(F51)
Loads the value from a specified 8-bit register into the lower 8 bits of the accumulator.

Data store 2 (F52)
Loads the value of the most significant 4 bits of a specified 8 bit register into the least significant 4 bits of the accumulator.

Data store 3 (F53)
Loads the value of the least significant 4 bits of a specified 8 bit register into the least significant 4 bits of the accumulator.

Data store 5 (F55)
Loads the value of 16-image register locations for a specified 16-point input module into the accumulator.

Data out (F60)
Copies the 16-bit value in the accumulator to a 16-bit reference or two consecutive 8-bit registers.

Data out 1 (F61)
Copies the value in the lower 8 bits of the accumulator to a specified 8-bit register.

Data out 2 (F62)
Copies the value in the least significant 4 bits of the accumulator into the most significant 4 bits of a specified 8-bit register.

Data out 3 (F63)
Copies the value in the least significant 4 bits of the accumulator to the least significant 4 bits of a specified 8-bit register.

Data out 5 (F65)
Copies the 16-bit value in the accumulator to the image register of a specified 16 point output module.

Bit Operation Instructions

Shift left (F80)
Shifts the value in the accumulator a specified number of bits (15 maximum) to the left.

Shift right (F81)
Shifts the value in the accumulator a specified number of bits (15 maximum) to the right.

Decode (F82)
Decodes a 4-bit binary number in the accumulator by setting the appropriate bit position to a one.

Encode (F83)
Encodes the accumulator bit position that contains a 1 by returning the appropriate 4-bit binary representation.

Binary (F85)
Converts a BCD value in the accumulator to the binary/HEX equivalent value.

Binary coded decimal (F86)
Converts a binary/HEX equivalent value in the accumulator to the BCD equivalent.

Inverse (F84)
Generates the one's complement of the number in the accumulator.

Accumulator Logic Instructions

Data and (F75)
Logically ANDs the value in a 16-bit reference, two consecutive 8-bit registers, or a 4-digit BCD constant with the value in the accumulator.

Data or (F76)
Logically ORs the value in a 16-bit reference, two consecutive 8-bit registers, or 4-digit BCD constant with the value in the accumulator.

Compare (F70)
Compares the value in a 16-bit reference, two consecutive 8-bit registers, or 4-digit BCD constant with the value in the accumulator.

Math Instructions

Add (F71)
Adds the value of a 16-bit reference, two consecutive 8 bit registers, or a 4-digit BCD constant with the value in the accumulator.

Subtract (F72)
Subtracts the value in a 16-bit register, two consecutive 8-bit registers, or a 4-digit BCD constant from the value in the accumulator.

Multiply (F73)
Multiplies the value in a 16-bit register, two consecutive 8-bit registers, or a 4-digit BCD constant by the value in the accumulator.

Divide (F74)
Divides the value in the accumulator by the value in a 16-bit register, two consecutive 8-bit registers, or a 4-digit BCD constant.

Message Instructions

Fault (F20)
Used to display a 4-digit BCD constant, 16-bit register, or two consecutive 8-bit data registers on the handheld programmer or DirectSoft.

Boolean Instructions

Store (STR)
Begins a new rung or an additional branch in a rung with a normally open contact.

Store not (STR NOT)
Begins a new rung or an additional branch in a rung with a normally closed contact.

Or (OR)
Logically ORs a normally open contact in parallel with another contact in a rung.

Or Not (OR NOT)
Logically ORs a normally closed contact in parallel with another contact in a rung.

And (AND)
Logically ANDS a normally open contact in series with another contact in a rung.

And Not (AND NOT)
Logically ANDS a normally closed contact in series with another contact in a rung.

And Store (AND STR)
Logically ANDS two branches of a rung in series.

Or Store (OR STR)
Logically ORS two branches of a rung in parallel.

Out (OUT)
Reflects the status of the rung (on/off) and outputs the discrete (on/off) state to the specified image register point or memory location.

Or Out (OR OUT)
Reflects the status of the rung and outputs the discrete (ON/OFF) state to the image register. Multiple OR OUT instructions referencing the same discrete point can be used in the program.

Not (NOT)
Inverts the status of the rung at the point of the instruction.

Positive Differential (PD)
Is typically known as a a one shot. When the input logic produces an off to on transition, the output will energize for one CPU scan.

Set (SET)
An output that turns on a point or a range of points. The reset instruction is used to turn the point(s) OFF that were set ON with the set instruction.

Reset (RST)
An output that resets a point or a range of points.

Pause outputs (PAUSE)
Disables the update for a range of specified output points.

Comparative Boolean Instructions

Store if Equal (STR E)
Begins a new rung or additional branch in a rung with a normally open comparative contact. The contact will be on when A = B.

Store if Not Equal (STR NOT E)
Begins a new rung or additional branch in a rung with a normally closed comparative contact. The contact will be on when A =/ B.

Or if Equal (OR E)
Connects a normally open comparative contact in parallel with another contact. The contact will be on when A = B.

Or if Not Equal (OR NOT E)
Connects a normally closed comparative contact in parallel with another contact. The contact will be on when A =/ B.

And if Equal (AND E)
Connects a normally open comparative contact in series with another contact. The contact will be on when A = B.

And if Not Equal (AND NOT E)
Connects a normally closed comparative contact in series with another contact. The contact will be on when A =/ B.

Store (STR)
Begins a new rung or additional branch in a rung with a normally open comparative contact. The contact will be on when A > B.

Store not (STR NOT)
Begins a new rung or additional branch in a rung with a normally closed comparative contact. The contact will be on when A > B.

Or (OR)
Connects a normally closed comparative contact in parallel with another contact. The contact will be on when A _> B.

Or Not (OR NOT)
Connects a normally closed comparative contact in parallel with another contact. The contact will be on when A<B.

And (AND)
Connects a normally open comparative contact in series with another contact. The contact will be on when A _> B.

And Not (AND NOT)
Connects a normally closed comparative contact in series with another contact. The contact will be on when A < B.

Bit of Word Boolean Instructions

Store Bit of Word (STRB)
Begins a new rung or an additional branch in a rung with a normally open contact that examines single bit of a V-memory location.

Store Not Bit of Word (STRNB)
Begins a new rung or an additional branch in a rung with a normally closed contact that examines single bit of a V-memory location.

Or Bit of Word (ORB)
Logically ORS a normally open bit of word contact in parallel with another contact in a rung.

Or Not Bit of Word (ORNB)
Logically ORS a normally closed bit of word contact in parallel with another contact in a rung.

And Bit of Word (ANDB)
Logically ANDS a normally open bit of word contact in series with another contact in a rung.

And Not Bit of Word (ANDNB)
Logically ANDS a normally closed bit of word contact in series with another contact in a rung.

Out Bit of Word (OUTB)
Reflects the status of the rung (on/off) and outputs the discrete (on/off) state to the specified bit of a V-memory location.

Set Bit of Word (SETB)
An output that turns on a single bit of a V-memory location. The bit remains on until it is reset. The reset bit of word instruction is used to turn off the bit.

Reset Bit of Word (RSTB)
An output that resets a single bit of a V-memory location.

Immediate Instructions

Store Immediate (STR I)
Begins a rung/branch of logic with a normally open contact. The contact will be updated with the current input field status when processed in the program scan.

Store Not Immediate (STR NOT I)
Begins a rung/branch of logic with a normally closed contact. The contact will be updated with the current input field status when processed in the program scan.

Or Immediate (OR I)
Connects a normally open contact in parallel with another contact. The contact will be updated with the current input field status when processed in the program scan.

Or Not Immediate (OR NOT I)
Connects a normally closed contact in parallel with another contact. The contact will be updated with the current input field status when processed in the program scan.

And Immediate (AND I)
Connects a normally open contact in series with another contact. The contact will be updated with the current input field status when processed in the program scan.

And Not Immediate (AND NOT I)
Connects a normally closed contact in series with another contact. The contact will be updated with the current input field status when processed in the program scan.

Out Immediate (OUT I)
Reflects the status of the rung. The output field device status is updated when the instruction is processed in the program scan.

Or out immediate (OR OUTI)
Reflects the status of the rung and outputs the discrete (ON/OFF) state to the image register. Multiple OR OUT instructions referencing the same discrete point can be used in the program. The output field device status is updated when the instruction is processed in the program scan.

Set Immediate (SET I)
An output that turns on a point or a range of points. The reset instruction is used to turn the point(s) off that were set. The output field device status is updated when the instruction is processed in the program scan.

Reset Immediate (RST I)
An output that resets a point or a range of points. The output field device status is updated when the instruction is processed in the program scan.

Timer, Counter, and Shift Register Instructions

Timer (TMR)
Single input incremental timer with 0.1 second resolution (0-999.9 seconds).

Fast Timer (TMRF)
Single input incremental timer with 0.01 second resolution (0-99.99 seconds).

Accumulating Timer (TMRA)
Two input incremental timer with 0.1 second resolution (0-9,999,999.9 sec.). Time enable/reset inputs control the timer.

Accumulating Fast Timer (TMRAF)
Two input incremental timer with 0.01 second resolution (0-999,999.99 sec.). Time and enable/reset inputs control timer.

Counter (CNT)
Two input incremental counter (0-9999). Count and reset inputs control the counter.

Stage Counter (SGCNT)
Single input incremental counter (0-9999). RST instruction must be used to reset count.

Up Down Counter (UDC)
Three input counter (0-99999999). Up, down, and reset inputs control the counter.

Shift Register (SR)
Shifts data through a range of control relays with each clock pulse. The data, clock, and reset inputs control the shift register.

D3-350 Instruction Set

Accumulator / Stack Load and Output Data

Load (LD)
Loads a 16-bit word into the lower 16-bits of the accumulator / stack.

Load Double (LDD)
Loads a 32-bit word into the accumulator / stack.

Load Real Number (LDR)
Loads a real number contained in two consecutive V-memory locations or an 8-digit constant into the accumulator.

Load Formatted (LDF)
Loads the accumulator with a specified number of consecutive discrete memory bits.

Load Address (LDA)
Loads the accumulator with the HEX value for an octal constant (address).

Load Accumulator Indexed (LDX)
Loads the accumulator with a V-memory address to be offset by the value in the accumulator stack.

Load Accumulator Indexed from Data Constants (LDSX)
Loads the accumulator with an offset constant value (ACON/NCON) from a data label area (DLBL).

Out (OUT)
Copies the value in the lower 16-bits of the accumulator to a specified V memory location.

Out Double (OUTD)
Copies the value in the accumulator to two consecutive V-memory locations.

Out Formatted (OUTF)
Outputs a specified number of bits (1-32) form the accumulator to the specified discrete memory locations.

Output Indexed (OUTX)
Copies a 16-bit value from the first level of the accumulator stack to a source address offset by the value in the accumulator.

Pop (POP)
Moves the value from the first level of the accumulator stack to the accumulator and shifts each value in the stack up one level.

Logical Instructions (Accumulator)

And (AND)
Logically ANDs the lower 16 bits in the accumulator with a V memory location.

And Double (ANDD)
Logically ANDs the value in the accumulator with an 8 digit constant.

And Formatted (ANDF)
Logically ANDs the value in the accumulator and a specified range of discrete memory bits (1-32).

Or (OR)
Logically ORs the lower 16-bits in the accumulator with a V-memory location.

Or Double (ORD)
Logically ORs the value in the accumulator with an 8-digit constant.

Or Formatted (ORF)
Logically ORs the value in the accumulator with a range of discrete bits (1-32).

Exclusive Or (XOR)
Performs an Exclusive OR of the value in the lower 16-bits of the accumulator and a V-memory location.

Exclusive Or Double (XORD)
Performs an Exclusive OR of the value in the accumulator and an 8 digit constant.

Exclusive Or Formatted (XORF)
Performs an exclusive OR of the value in the accumulator and a range of discrete bits (1-32).

Compare (CMP)
Compares the value in the lower 16 bits of the accumulator with a V-memory location.

Compare Double (CMPD)
Compares the value in the accumulator with two consecutive V-memory locations or an 8-digit constant.

Compare Formatted (CMPF)
Compares the value in the accumulator with a specified number of discrete bits (1-32).

Compare Real Number (CMPR)
Compares the real number in the accumulator with two consecutive V-memory locations or an 8-digit real number constant.

Math Instructions (Accumulator)

Add (ADD)
Adds a BCD value in the lower 16-bits in the accumulator with a V-memory location. The result resides in the accumulator.

Add Double (ADDD)
Adds a BCD value in the accumulator with two consecutive V-memory locations or an 8-digit constant. The result resides in the accumulator.

Add Real Number (ADDR)
Adds a real number in the accumulator with a real number constant or a real number contained in two consecutive V-memory locations. The result resides in the accumulator.

Subtract (SUB)
Subtract a BCD value, which is either a V-memory location or a 4-digit constant, from the lower 16-bits in the accumulator. The result resides in the accumulator.

Subtract Double (SUBD)
Subtracts a BCD value, which is either two consecutive V-memory locations or an 8-digit constant, from a value in the accumulator. The result resides in the accumulator.

Subtract Real Number (SUBR)
Subtract a real number, which is either two consecutive V-memory locations or a real number constant, from the real number in the accumulator. The result resides in the accumulator.

Multiply (MUL)
Multiplies a BCD value, which is either a V-memory location or a 4-digit constant, by the value in the lower 16-bits in the accumulator. The result resides in the accumulator.

Multiply Double (MULD)
Multiplies a BCD value contained in two consecutive V-memory location by the value in the accumulator. The result resides in the accumulator.

Multiply Real Number (MULR)
Multiplies a real number, which is either two consecutive V-memory locations or a real number constant, by the real number in the accumulator. The result resides in the accumulator.

Divide (DIV)
Divides a BCD value in the lower 16-bits of the accumulator by a BCD value which is either a V-memory location or a 4-digit constant. The result resides in the accumulator.

Divide Double (DIVD)
Divides a BCD value in the accumulator by a BCD value which is either two consecutive V-memory locations or an 8-digit constant. The result resides in the accumulator.

Divide Real Number (DIVR)
Divides a real number in the accumulator by a real number which is either two consecutive V-memory locations or a real number constant. The result resides in the accumulator.

Add Binary (ADDB)
Adds the binary value in the lower 16 bits of the accumulator to a value which is either a V-memory location, or a 16-bit constant. The result resides in the accumulator.

Subtract Binary (SUBB)
Subtracts a 16-bit binary value, which is either a V-memory location or a 16 bit constant, from the lower 16 bits in the accumulator. The result resides in the accumulator.

Multiply Binary (MULB)
Multiplies a 16-bit binary value, which is either a V-memory location or a 16-bit constant, by the lower 16 bits in the accumulator. The result resides in the accumulator.

Divide Binary (DIVB)
Divides the binary value in the lower 16 bits in the accumulator by a value which is either a V-memory location or a 16-bit constant. The result resides in the accumulator.

Increment (INC)
Increments a BCD value in a specified V-memory location by 1 each time the instruction is executed.

Decrement (DEC)
Decrements a BCD value in a specified V-memory location by 1 each time the instruction is executed.

Increment Binary (INCB)
Increments a binary value in a specified V-memory location by 1 each time the instruction is executed.

Decrement Binary (DECB)
Decrements a binary value in a specified V-memory location by 1 each time the instruction is executed.

Bit Instructions (Accumulator)

Sum (SUM)
Counts the number of bits in set to "1" in the accumulator. The HEX result resides in the accumulator.

Shift Left (SHFL)
Shifts the bits in the accumulator a specified number of places to the left.

Shift Right (SHFR)
Shifts the bits in the accumulator a specified number of places to the right.

Rotate Left (ROTL)
Rotates the bits in the accumulator a specified number of places to the left.

Rotate Right (ROTR)
Rotates the bits in the accumulator a specified number of places to the right.

Encode (ENCO)
Encodes the bit position set to 1 in the accumulator, and returns the appropriate binary representation in the accumulator.

Decode (DECO)
Decodes a 5-bit binary value (0-31) in the accumulator by setting the appropriate bit position to 1 in the accumulator.

Number Conversion Instructions (Accumulator)

Binary (BIN)
Converts the BCD value in the accumulator to the equivalent binary value. The result resides in the accumulator.

Binary Coded Decimal (BCD)
Converts the binary value in the accumulator to the equivalent BCD value. The result resides in the accumulator.

Invert (INV)
Takes the one's complement of the 32-bit value in the accumulator. The result resides in the accumulator.

Ten's complement (BCDCPL)
Takes the ten's complement of the BCD value in the accumulator. The result resides in the accumulator.

ASCII to HEX (ATH)
Converts the table of ASCII values to a table of hexadecimal values.

HEX to ASCII (HTA)
Converts a table of hexadecimal values to a table of ASCII values.

Segment (SEG)
Converts a 4-digit HEX number in the accumulator to a corresponding bit pattern for interfacing to seven segment displays. The result resides in the accumulator.

Gray code to BCD (GRAY)
Converts a 16-bit GRAY code value in the accumulator to a corresponding BCD value. The result resides in the accumulator.

Shuffle digits (SFLDGT)
Shuffles a maximum of 8 digits rearranging them in a specified order. The result resides in the accumulator.

Binary to Real Number (BTOR)
Converts the integer value in the accumulator into a real number. The result resides in the accumulator.

Real Number to Binary (RTOB)
Converts the real number in the accumulator into an integer value. The result resides in the accumulator.

Table Instructions

Move (MOV)
Moves the values from on V-memory table to another V-memory table.

Move Memory Cartridge/Load Label (MOVMC/LDBL)
Copies data from data label area in program ladder memory to V-memory.

Move Memory Cartridge/Load Label (MOVMC/LDLBL)
Copies data between V-memory and program ladder memory.

Clock/Calendar Instructions

Date (DATE)
Sets the date (year, month, day, day of the week) in the CPU calendar using two consecutive V-memory locations.

Time (TIME)
Sets the time (hour, seconds, and minutes) in the CPU using two consecutive V-memory locations.

CPU Control Instructions

No Operation (NOP)
Inserts a no operation coil at a specified program address.

End (END)
Marks the termination point for the normal program scan. An End instruction is required at the end of the main program body.

Stop (STOP)
Changes the operational mode of the CPU form Run to Program (Stop).

Reset Watchdog Timer (RSTWT)
Resets the CPU watchdog timer.

Program Control Instructions

Goto/Label (GOTO/LBL)
Skips (does not execute) all instructions between the GOTO and the corresponding label (LBL) instruction.

For/Next (FOR/NEXT)
Executes the logic between the FOR and NEXT instructions a specified number of times.

Goto Subroutine/Subroutine Return Conditional/ Subroutine Return (GTS/SBR w/RT)
When a GTS instruction is executed, the program jumps to the SBR (subroutine). The subroutine is terminated with an RT instruction (unconditional return). When a return is executed, the program continues from the instruction after the calling GTS instruction.

Client Line Set/Client Line Reset (MLS/MLR)
Allows the program to control sections of ladder logic by forming a new power rail. The MLS marks the beginning of a power rail and the MLR marks the end of the power rail control.

Interrupt Instructions

Interrupt Routine/Interrupt Return/Interrupt Return Conditional (INT/IRT/IRTC)
When a hardware or software interrupt occurs, the interrupt routine will be executed. The INT instruction is the beginning of the interrupt routine. The interrupt routine is terminated with an IRT instruction (unconditional interrupt return). When an interrupt return is reached, the execution of the program continues from the instruction where the program execution was prior to the interrupt.

Enable Interrupt (ENI)
Enables hardware and software interrupts to be acknowledged.

Disable Interrupt (DISI)
Disables hardware and software interrupts from being acknowledged.

Intelligent Module Instructions

Read from Intelligent Module (RD)
Reads a block of data (1-128 bytes max.) from an intelligent I/O module.

Write to Intelligent Module (WT)
Writes a block of data (1-28 bytes max.) to an intelligent I/O module.

Network Instructions

Read from network (RX)
Reads a block of data from another CPU on the network.

Write to network (WX)
Writes a block of data from the Client device to a Server device on the network.

Message Instructions

Fault/Data Label (FAULT/DLBL)
Displays a V-memory value or a Data label constant to the handheld programmer or personal computer using DirectSOFT.

Numerical Constant/ASCII constant (NCON/ACON)
Stores constants in numerical or ASCII form for use with other instructions.

Print Message (PRINT)
Prints the embedded text or text/data variable message to the specified communications port. Maximum message length is 255 words.

D3-350 Instruction Set

RLL PLUS Programming Instructions

Initial stage (ISG)
The initial stage instruction is used as a starting point for the user application program. The ISG instruction will be active on power up and PROGRAM to RUN transitions.

Stage (SG)
Stage instructions are used to create structured programs. They are program segments which can be activated or deactivated with control logic.

Jump (JMP)
Normally open coil that deactivates the active stage and activates a specified stage when there is power flow to the coil.

Not Jump (NJMP)
Normally closed coil that deactivates the active stage and activates a specified stage when there is no power flow to the coil.

Converge stages (CV)
Converge stages are a group of stages that when all stages are active the associated converge jump(s) (CVJMP) will activate another stage(s). One scan after the CVJMP is executed, the converge stages will be deactivated.

Converge Jump (CVJMP)
Normally open coil that deactivates the active CV stages and activates a specified stage when there is power flow to the coil.

Block Call/Block/Block End (BCALL and BEND)
BCALL is a normally open coil that activates a block of stages when there is power flow to the coil. BLK is the label that marks the beginning of a block of stages. BEND is a label used to mark the end of a block of stages.

Drum Instructions

Timed Drum with Discrete Outputs (DRUM)
Time driven drum with up to 16 steps and 16 discrete output points. Output status is written to the appropriate output during each step. Specify a time base per count (in milliseconds). Each step can have a different number of counts to trigger the transition to the next step. Also define preset step as destination when reset occurs.

Time & Event Drum with Discrete Outputs (EDRUM)
Time and/or event driven drum with up to 16 steps and 16 discrete output points. Output status is written to the appropriate output during each step. Specify a time base per count (in milliseconds). Each step can have a different number of counts and an event to trigger counting. Once the time has expired, a transition to the next step occurs. Also define preset step as destination when reset occurs.

Time & Event Drum with Discrete Outputs & Output Mask (MDRUMD)
Time and/or event driven drum with up to 16 steps and 16 discrete output points. Actual output status is the result of a bit-by-bit AND between the output mask and the bit mask in the step. Specify a time base per count (in milliseconds). Each step can have a different number of counts and an event to trigger counting. Once the time has expired, a transition to the next step occurs. Also define preset step as destination when reset occurs.

Time & Event Drum with Word Output & Output Mask (MDRUMW)
Time and/or event driven drum with up to 16 steps and a single V-memory output location. Actual output word is the result of a bit-by-bit AND between the word mask and the bit mask in the step. Specify a time base per count (in milliseconds). Each step can have a different number of counts and an event to trigger counting. Once the time has expired, a transition to the next step occurs. Also define preset step as destination when reset occurs.