Instruction Set

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.

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 instructions.

Reset (RST)
An output that resets a point(s).

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 is not equal to 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 is not equal to 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 is not equal to 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 open comparative contact in parallel with another contact. The contact will be on when A > B.

Or Not (OR NOT)
Connects a normally open 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 open 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 a 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 a 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.

Differential Instructions

Positive differential (PD)
One-shot output coil. When the input logic produces an off to on transition, the output will energize for one CPU scan.

Store Positive Differential (STRD)
Leading edge triggered one-shot contact. When the corresponding memory location transitions from low to high, the contact comes on for one CPU scan.

Store Negative Differential (STRND)
Trailing edge triggered one-shot contact. When the corresponding memory location transitions from high to low, the contact comes on for one CPU scan.

Or Positive Differential (ORD)
Logically ORs a leading edge triggered one-shot contact in parallel with another contact in a rung.

Or Negative Differential (ORND)
Logically ORs a trailing edge triggered one-shot contact in parallel with another contact in a rung.

And Positive Differential (ANDD)
Logically ANDs a leading edge triggered one-shot contact in series with another contact in a rung.

And Negative Differential (ANDND)
Logically ANDs a trailing edge triggered one-shot contact in series with another contact in a rung.

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.

Load Immediate (LDI)
Loads the accumulator with the contents of a specified 16-bit V-memory location. The status for each bit of the specified V-memory location is loaded into the accumulator. Typically used for input module V-memory addresses. Allows you to specify the V location instead of the X location and the number of points as with the LDIF.

Load immediate Formatted (LDIF)
Loads the accumulator with a specified number of consecutive inputs. The field device status for the specified inputs points is loaded into the accumulator when the instruction is executed.

Out Immediate (OUTI)
Outputs the contents of the accumulator to a specified V-memory location. The status for each bit of the specified V-memory location will reflect the status of the lower 16 bits of the accumulator. Typically used for output module V-memory addresses. Allows you to specify the V location instead of the Y location and the number of points as with the OUTIF.

Out immediate Formatted (OUTIF)
Outputs the contents of the accumulator to a specified number of consecutive outputs. The output field devices are updated when the instruction is processed by the program scan.

Timer, Counter, and Shift Register Instructions

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

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

Accumulating Timer (TMRA)
Two input incrementing timer with 0.1 second resolution (0-9999999.9 sec.). Time and enable/reset inputs control the timer.

Accumulating Fast Timer (TMRAF)
Two input incrementing timer with 0.01 second resolution (0-999999.99 sec.).Time input and enable/reset input control timer.

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

Stage Counter (SGCNT)
Single input incrementing 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.

Accumulator/Data Stack Load & Output Load (LD)

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 a 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) from the accumulator to the specified discrete memory locations.

Out Least (OUTL)
Copies the value in the lower 8 bits of the accumulator to the lower 8 bits of a specified V-memory location.

Out Most (OUTM)
Copies the value in the upper 8 bits of the lower accumulator word (1st 16 bits) to the upper 8 bits of a specified V-memory location.

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.

Instruction Set

Accumulator Logic Instructions

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 two consecutive V-memory locations.

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

And with Stack (ANDs)
Only Logically ANDs the value in the accumulator with the first value in the accumulator stack.

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 two consecutive V-memory locations.

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

Or with Stack (ORs)
Only) Logically ORs the value in the accumulator with the first value in the accumulator stack.

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 two consecutive V-memory locations.

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

Exclusive Or with Stack (XORs)
Performs an exclusive or of the value in the accumulator and the first accumulator stack location.

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

Compare with Stack (CMPS)
Compares the value in the accumulator with the first accumulator stack location.

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

Math Instructions

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 in a V-memory location 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 a real number 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 an 8-digit 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 locations 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)
Only Divides a BCD value in the accumulator by a BCD value in two consecutive V-memory locations. 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.

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.

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.

Add Binary Double (ADDBD)
Adds the binary value in the accumulator to a value which is either two consecutive V-memory locations or a 32 bit constant. The result resides in the accumulator.

Subtract Binary (SUBB)
Subtract 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.

Subtract Binary Double (SUBBD)
Subtracts a 32 bit binary value, which is either two consecutive V-memory locations or a 32 bit constant, from the value 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 bits 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.

Add Formatted (ADDF)
Adds the BCD value in the accumulator to a value which is a range of discrete bits (1-32). The result resides in the accumulator.

Subtract Formatted (SUBF)
Subtracts a BCD value which is a range of discrete bits (1-32) from the BCD value in the accumulator. The result resides in the accumulator.

Multiply Formatted (MULF)
Multiplies a BCD value in the lower 16 bits in the accumulator by a BCD value which is a range of discrete bits (1-16). The result resides in the accumulator.

Divide Formatted (DIVF)
Only Divides the BCD value in the lower 16 bits in the accumulator by the BCD value which is a range of discrete bits (1-16). The result resides in the accumulator.

Add Top of Stack (ADDS)
Adds the BCD value in the accumulator with the BCD value in the first level of the accumulator stack. The result resides in the accumulator.

Subtract Top of Stack (SUBS)
Subtracts the BCD value in the first level of the accumulator stack from the BCD value in the accumulator. The result resides in the accumulator.

Multiply Top of Stack (MULS)
Multiplies a 4-digit BCD value in the first level of the accumulator stack by a 4-digit BCD value in the accumulator. The result resides in the accumulator.

Divide by Top of Stack (DIVS)
Divides the 8-digit BCD value in the accumulator by the 4-digit BCD value in the first level of the accumulator stack. The result resides in the accumulator.

Add Binary Top of Stack (ADDBS)
Adds the binary value in the accumulator with the binary value in the first accumulator stack location. The result resides in the accumulator.

Subtract Binary Top of Stack (SUBBS)
Subtracts the binary value in the first level of the accumulator stack from the binary value in the accumulator. The result resides in the accumulator.

Multiply Binary Top of Stack (MULBS)
Multiplies the 16 bit binary value in the first level of the accumulator stack by the 16 bit binary value in the accumulator. The result resides in the accumulator.

Divide Binary Top of Stack (DIVBS)
Divide a value in the accumulator by the binary value in the top location of the stack. The accumulator contains the result.

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.

Number Conversion Instructions

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)
Only Converts a ta ble of ASCII values to a table of hexadecimal values.

HEX to ASCII (HTA)
OOnly 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 binary value in the accumulator into a real number. The result resides in the accumulator.

Real to Binary (RTOB)
Converts the real number in the accumulator into a binary value. The result resides in the accumulator.

Radian Real Conversion (RADR)
Converts the real degree value in the accumulator to the equivalent real number in radians. The result resides in the accumulator.

Degree Real Conversion (DEGR)
Converts the real radian value in the accumulator to the equivalent real number of degrees. The result resides in the accumulator.

Trigonometric Instructions

Square Root Real (SQRTR)
Takes the square root of the real number stored in the accumulator. The result resides in the accumulator.

Sine Real (SINR)
Takes the sine of the real number stored in the accumulator. The result resides in the accumulator.

Cosine Real (COSR)
Takes the cosine of the real number stored in the accumulator. The result resides in the accumulator.

Tangent Real (TANR)
Takes the tangent of the real number stored in the accumulator. The result resides in the accumulator.

Arc Sine Real (ASINR)
Takes the inverse sine of the real number stored in the accumulator. The result resides in the accumulator.

Arc Cosine Real (ACOSR)
Takes the inverse cosine of the real number stored in the accumulator. The result resides in the accumulator.

Arc Tangent real (ATANR)
Takes the inverse tangent of the real number stored in the accumulator. The result resides in the accumulator.

Instruction Set

Bit Operation Instructions

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.

Set Bit (SETBIT)
Sets a single bit (to a 1) in a V-memory location.

Reset Bit (RSTBIT)
Resets a single bit (to a 0) in a V-memory location.

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.

Table Instructions

Fill (FILL)
Fills a table of specified V-memory locations with a value which is either a V-memory location or a 4-digit constant.

Find (FIND)
Finds a value in a V-memory table and returns the table position, containing the value, to the accumulator.

Find Greater Than (FDGT)
Finds a value in a V-memory table which is greater than the specified search value. The table position containing the value is returned to the accumulator.

Find Block (FINDB)
Finds a block of data values in a V-memory table and returns the starting address of the table containing the values to the accumulator.

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

Table To Destination (TTD)
Moves a value from the top of a V-memory table to a specified V-memory location. The table pointer increments each scan.

Remove From Bottom (RFB)
Moves a value from the bottom of a V-memory table to a specified V-memory location. The table pointer decrements each scan.

Source To Table (STT)
Moves a value from a specified V-memory location to a V-memory table. The table pointer increments each scan.

Remove From Table (RFT)
Pops a value from the top of a V-memory table and stores it in a specified V-memory location. The values in the V-memory table are shifted up each time a value is moved.

Add To Top of Table (ATT)
Pushes a value from a specified V-memory location onto the top of a V-memory table. All other values in the V-memory table are shifted down each time a value is pushed onto the table.

Table Shift Left (TSHFL)
Shifts a specified number of bits to the left in a V-memory table.

Table Shift Right (TSHFR)
Shifts a specified number of bits to the right in a V-memory table.

Move Block (MOVBLK)
Copies a specified number of words from a Data Label Area of program memory (ACON, NCON) to a V-memory area.

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

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/RTC or RT)
When a GTS instruction is executed, the program jumps to the SBR (subroutine). The subroutine is terminated with an RT instruction (unconditional return). An RTC (conditional return) can be used in conjunction with the RT. When a conditional/unconditional 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 Conditional/Interrupt Return (INT/IRTC/IRT)
When a hardware or software interrupt has occurred, 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). An IRTC (conditional interrupt return) can be used in conjunction with the IRT. When a conditional/unconditional 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.

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.

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 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 from Run to Program (Stop).

Break (BREAK)
Changes the operational mode of the CPU from Run to the Test Program mode.

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

Intelligent I/O 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 (1-128 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.

RLL PLUS Programming Instructions

Initial stage (ISG)
The initial stage instruction is used for a starting point for 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 w/BLK 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 which 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 the 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 and Output Mask (MDRMD)
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 the 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 (MDRMW)
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 the counting. Once the time has expired, a transition to the next step occurs. Also define preset step as destination when reset occurs.

Here are some of the IBox Instructions Available

The IBox instructions are available when using a D4-454 CPU and DirectSOFT6.1 or later.

Analog Scale 12 Bit BCD to BCD (ANSCL)
Scales a 12 bit BCD analog value (0-4095 BCD) into BCD engineering units. Only works with unipolar unsigned raw values.

Analog Scale 12 Bit Binary to Binary (ANSCLB)
Scales a 12 bit binary analog value (0-4095 decimal) into Binary engineering units. Only works with unipolar unsigned raw values.

Filter Over Time - BCD (FILTER)
Performs a first-order filter on the Raw Data on a defined time interval (BCD).

Filter Over Time - Binary (FILTERB)
Perform a first-order filter on the Raw Data on a defined time interval (binary).

Hi/Low Alarm - BCD (HILOAL)
Monitors a BCD value V memory location and sets four possible alarm states, High-High, High, Low, and Low-Low.

Hi/Low Alarm - Binary (HILOALB)
Monitors a binary (decimal) value V memory location and sets four possible alarm states, High-High, High, Low, and Low-Low.

IBox Instructions - Discrete Helper

Off Delay Timer - (OFFDTMR)
Delays the "turning off" of the Output parameter by the specified Off Delay Time (in hundredths of a second).

On Delay Timer - (ONDTMR)
Delays the "turning on" of the Output parameter by the specified amount of time (in hundredths of a second).

One Shot - (ONESHOT)
Turns on the given bit output parameter for one scan on an OFF to ON transition.

Push On / Push Off Circuit (PONOFF)
Toggles an output state whenever its input power flow transitions from off to on. Also known as a "flip-flop" circuit.

IBox Instructions - Memory

Move Single Word (MOVEW)
Moves (copies) a word to a memory location directly or indirectly via a pointer, either as a HEX constant, from a memory location, or indirectly through a pointer.

Move Double Word (MOVED)
Moves (copies) a double word to two consecutive memory locations directly or indirectly via a pointer, either as a double HEX constant, from a double memory location, or indirectly through a pointer to a double memory location.

IBox Instructions - Math

BCD to Real with Implied Decimal Point (BCDTOR)
Converts the given 4 digit WORD BCD value to a Real number, with the implied number of decimal points (K0-K4).

Double BCD to Real with Implied Decimal Point (BCDTORD)
Converts the given 8 digit DWORD BCD value to a Real number, given an implied number of decimal points (K0-K8).

Math - BCD (MATHBCD)
Allows entry of complex mathematical expressions like in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. Every V-memory reference MUST be to a single word BCD formatted value.

Math - Binary (MATHBIN)
Allows entry of complex mathematical expressions like in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. Every V-memory reference MUST be to a single word binary formatted value.

Math - Real (MATHR)
Allows entry of complex mathematical expressions like in Visual Basic, Excel, or C++ to do complex calculations, nesting parentheses up to 4 levels deep. Every V-memory reference MUST be able to fit into a double word Real formatted value.

Real to BCD with Implied Decimal Point and Rounding (RTOBCD)
Converts the absolute value of the given Real number to a 4 digit BCD number, compensating for an implied number of decimal points (K0-K4) and performs rounding.

Real to Double BCD with Implied Decimal Point and Rounding (RTOBCDD)
Converts the absolute value of the given Real number to an 8 digit DWORDBCD number, compensating for an implied number of decimal points (K0-K8) and performs rounding.

Square BCD (SQUARE)
Squares the given 4-digit WORD BCD number and writes it as an 8-digit DWORD BCD result.

Square Binary (SQUAREB)
Squares the given 16-bit WORD binary number and writes it as a 32-bit DWORD binary result.

Square Real (SQUARER)
Squares the given REAL DWORD number and writes it to a REAL DWORD result.

Sum BCD Numbers (SUMBCD)
Sums a list of consecutive 4-digit WORD BCD numbers into an 8-digit DWORD BCD result.

Sum Binary Numbers (SUMBIN)
Sums a list of consecutive 16-bit WORD binary numbers into an 32-bit DWORD binary result.

Sum Real Numbers (SUMR)
Sums a list of consecutive Real DWORD numbers into a Real DWORD result.

IBox Instructions - Communications

ECOM100 Configuration (ECOM100)
Defines the common information for a specific ECOM100 module which is used by the other ECOM100 IBoxes and resides at the top of the ladder/ stage program. If using more than one ECOM100 in a PLC system, a different ECOM100 Configuration IBox must be used for each ECOM100 module that utilizes ECOM IBox instructions.

ECOM100 Disable DHCP (ECDHCPD)
Commands the ECOM100 to use its internal TCP/IP settings.

ECOM100 Enable DHCP (ECDHCPE)
Commands the ECOM100 to obtain its TCP/IP settings from a DHCP server.

ECOM100 Query DHCP Setting (ECDHCPQ)
Determines if DHCP is enabled in the ECOM100.

ECOM100 Send E-mail (ECEMAIL)
Allows the ECOM100 to behave as an EMail client to send an SMTP request to the SMTP Server for sending the EMail messages to EMail addresses in the To: field and Cc: list hard coded in the ECOM100. Messages are limited to 100 characters for the entire instruction.

ECOM100 Restore Default E-mail Setup (ECEMRDS)
Restores the original EMail Setup data stored in the ECOM100 back to the working copy based on the specified ECOM100#.

ECOM100 E-mail Setup (ECEMSUP)
Modifies the working copy of the EMail setup currently in the ECOM100 based on the specified ECOM100#. You may pick and choose any or all fields to be modified using this instruction.

ECOM100 IP Setup (ECIPSUP)
Configures the three TCP/IP parameters in the ECOM100: IP Address, Subnet Mask and Gateway Address.

ECOM100 Read Description (ECRDDES)
Reads the ECOM100's Description field up to the number of specified characters.

ECOM100 Read Gateway Address (ECRDGWA)
Reads the ECOM100's Gateway address and stores it in 4 consecutive V memory locations in decimal format.

ECOM100 Read IP Address (ECRDIP)
Reads the ECOM100's IP address and stores it 4 consecutive V memory locations in decimal format.

ECOM100 Read Module ID (ECRDMID)
Reads the ECOM100's binary (decimal) WORD sized Module ID and stores it in V memory.

ECOM100 Read Module Name (ECRDNAM)
Reads the ECOM100's Module Name up to the number of specified characters and stores it in V memory.

ECOM100 Read Subnet Mask (ECRDSNM)
Reads the ECOM100's Subnet Mask address and stores it 4 consecutive V memory locations in decimal format.

ECOM100 Write Description (ECWRDES)
Writes the specified Description to the ECOM100 module.

ECOM100 Write Gateway Address (ECWRGWA)
Writes the specified Gateway IP Address to the ECOM100 module.

ECOM100 Write IP Address (ECWRIP)
Writes the specified IP Address to the ECOM100 module.

ECOM100 Write Module ID (ECWRMID)
Writes the specified Module ID to the ECOM100 module.

ECOM100 Write Name (ECWRNAM)
Writes the specified Name to the ECOM100 module.

ECOM100 Write Subnet Mask (ECWRSNM)
Writes the specified Subnet Mask to the ECOM100 module.

ECOM100 RX Network Read (ECRX)
Performs the RX instruction with built-in interlocking with all other ECOM100 RX (ECRX) and ECOM100 WX (ECWX) IBoxes in your program to simplify communications networking.

ECOM100 WX Network Write (ECWX)
Performs the WX instruction with built-in interlocking with all other ECOM100 RX (ECRX) and ECOM100 WX (ECWX) IBoxes in your program to simplify communications networking.

NETCFG Network Configuration (NETCFG)
Defines all the common information necessary for performing RX/WX Networking using the NETRX and NETWX IBox instructions via a local CPU serial port, DCM or ECOM module.

Network RX Read (NETRX)
Performs the RX instruction with built-in interlocking with all other Network RX (NETRX) and Network WX (NETWX) IBoxes in your program to simplify communications networking.

Network WX Read (NETWX)
Performs the WX instruction with built-in interlocking with all other Network RX (NETRX) and Network WX (NETWX) IBoxes in your program to simplify communications networking.

IBox Instructions - Counter I/O

CTRIO Configuration (CTRIO)
Defines the common information for a specific CTRIO module which is used by the other CTRIO IBox instructions and resides at the top of the ladder/stage program. If using more than one CTRIO module in a PLC system, a different CTRIO Configuration IBox must be used for each CTRIO module that utilizes CTRIO IBox instructions.

CTRIO Add Entry to End of Preset Table (CTRADPT)
Appends an entry to the end of a memory based Preset Table on a specific CTRIO Output resource. Will take more than 1 PLC scan to execute.

CTRIO Clear Preset Table (CTRCLRT)
Clears the RAM based Preset Table on a leading edge transition to this IBox. Will take more than 1 PLC scan to execute.

CTRIO Edit Preset Table Entry (CTREDPT)
Edits a single entry in a Preset Table on a specific CTRIO Output resource. Will take more than 1 PLC scan to execute.

CTRIO Edit Preset Table Entry and Reload (CTREDRL)
Performs this dual operation to a CTRIO Output resource in one CTRIO command. Will take more than 1 PLC scan to execute.

CTRIO Initialize Preset Table (CTRINPT)
Creates a single entry Preset Table in memory not as a file, on a specific CTRIO Output resource. Will take more than 1 PLC scan to execute.

CTRIO Initialize Preset Table on Reset (CTRINTR)
Configures the initial Preset Table to be automatically loaded whenever the Reset event occurs on a specific Output resource. Will take more than 1 PLC scan to execute.

CTRIO Load Profile (CTRLDPR)
Loads a CTRIO Profile File to a CTRIO Output resource on a leading edge transition to this IBox. Will take more than 1 PLC scan to execute.

CTRIO Read Error (CTRRDER)
Gets the decimal error code value from the CTRIO module and places it into the specified Error Code register. Since the Error Code in the CTRIO is only maintained until another CTRIO command is given, this instruction must be used immediately after the CTRIO IBox that reports an error via its Error bit parameter.

CTRIO Run to Limit Mode (CTRRTLM)
Loads the Run to Limit command and given parameters on a specific Output resource. The CTRIO's Input(s) must be configured as Limit(s) for this function to operate. Will take more than 1 PLC scan to execute.

CTRIO Run to Position Mode (CTRRTPM)
Loads the Run to Position command and given parameters on a specific Output resource. Will take more than 1 PLC scan to execute.

CTRIO Velocity Mode (CTRVELO)
Loads the Velocity command and given parameters on a specific Output resource. Will take more than 1 PLC scan to execute.

CTRIO Write File to ROM (CTRWFTR)
Writes the runtime changes made to a loaded CTRIO Preset Table back to Flash ROM. Will take more than 1 PLC scan to execute.