Facility Dynamics Engineering

Logic Diagram Symbols

A Starting Point for Creating Your Own Logic Diagrams

Presented By:

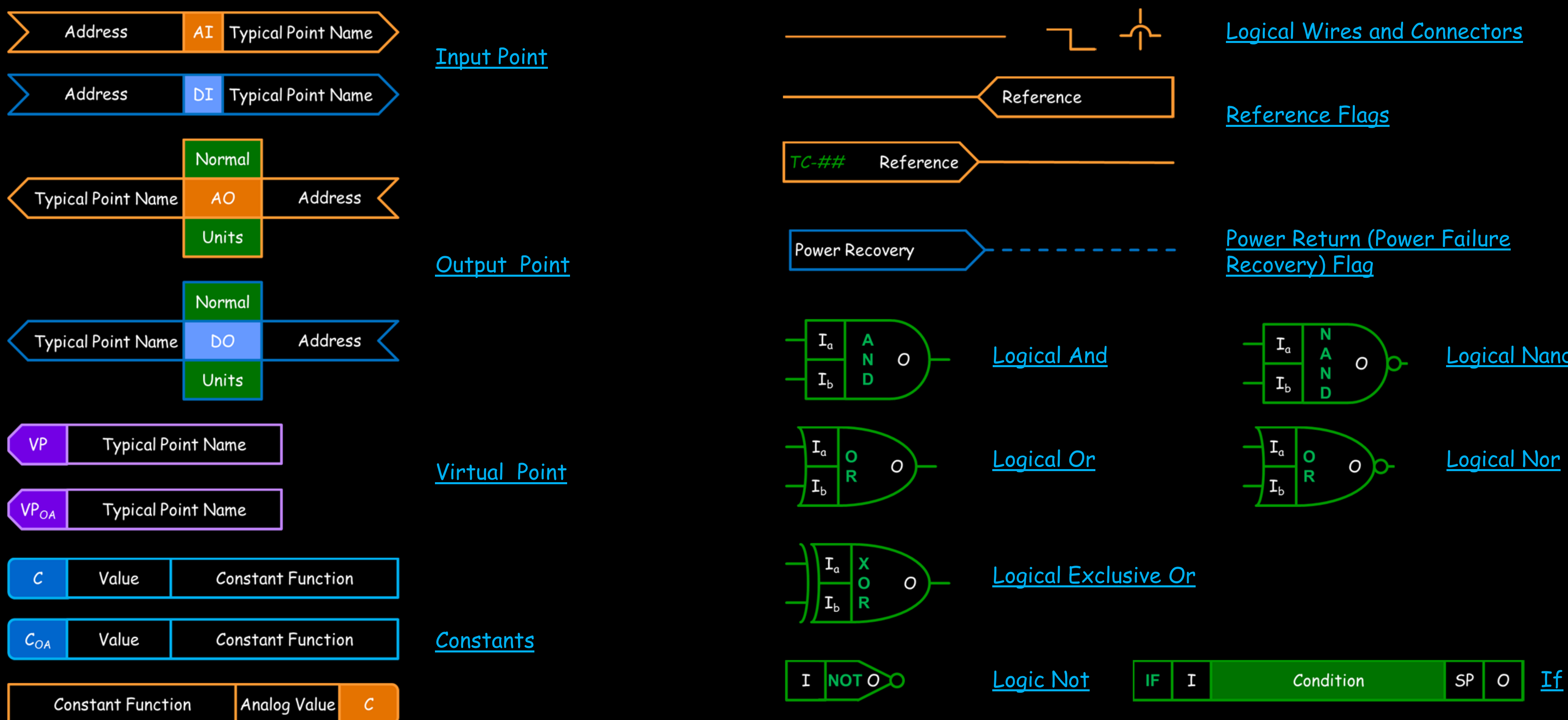
David Sellers, Facility Dynamics Engineering

Senior Engineer

Rev 2 – 2018-03-14

DDC Logic Function Blocks

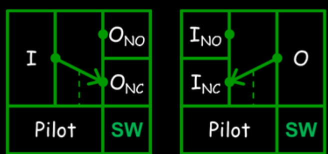
Contents (Click the link to jump to a detailed description)



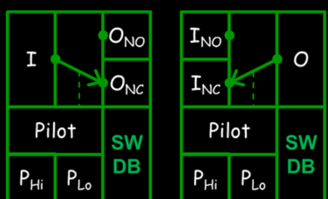
[Back to Contents Page 1](#)

DDC Logic Function Blocks

Contents (Page 2; Click the link to jump to a detailed description)



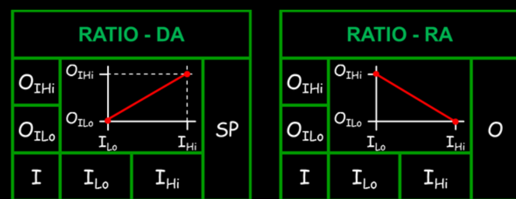
Piloted Switch



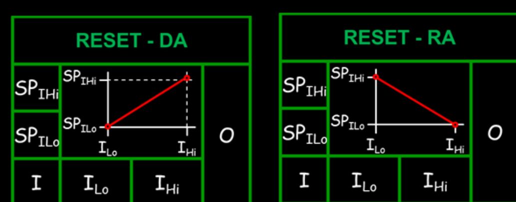
Piloted Switch with Dead Band

PID Loop (Proportional + Derivative+ Integral)

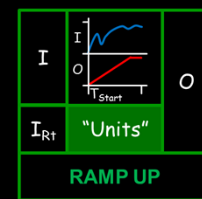
Floating Control



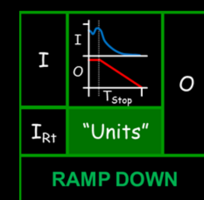
Ratio



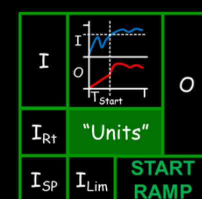
Reset



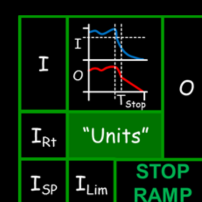
Ramp Up



Ramp Down



Start Ramp

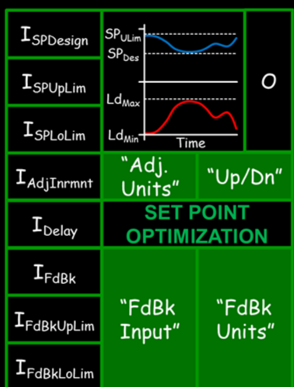


Stop Ramp

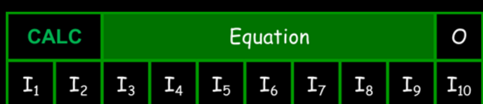
Back to Contents Page 2

DDC Logic Function Blocks

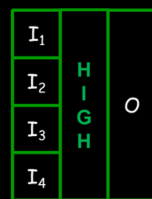
Contents (Page 3; Click the link to jump to a detailed description)



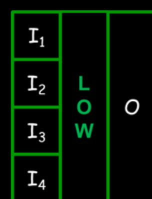
Set Point Optimization



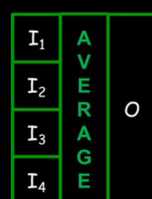
Calculation



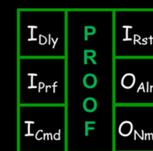
High Signal Selector



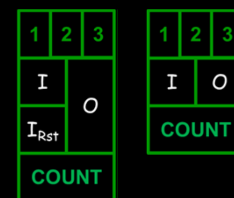
Low Signal Selector



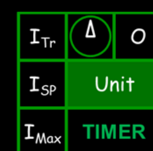
Average



Proof of Operation



Counter



Interval Timer

Time Clock/Schedule

[Back to Contents Page 3](#)

DDC Logic Function Blocks

Contents (Page 4; Click the link to jump to a detailed description)

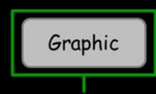
On Delay

Off Delay

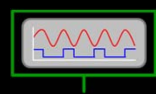
One Shot

Latch On

Latch Off



Display on System Graphic

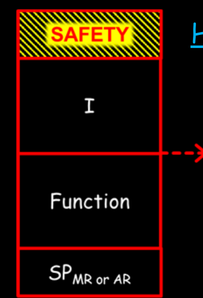


Trend and Archive Parameter

Generate an Alarm Based on the Parameter

Generate a Message Based on the Parameter

Run Time Monitor



Hardwired External Safety

[Back to Contents Page 4](#)

DDC Logic Function Blocks

Contents – Place Holder

This page intentionally left blank (except for this and the other place holders, which were intentionally left on the page)

[Back to Contents Page 5](#)

DDC Logic Function Blocks

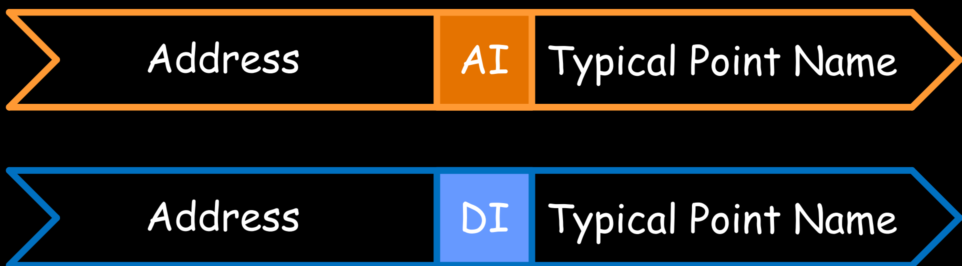
Contents – Place Holder

This page intentionally left blank (except for this and the other place holders, which were intentionally left on the page)

[Back to Contents](#)

DDC Logic Function Blocks

Input Point



Input Point – Reads a value from a physical input channel in the controller and converts it for use by the DDC function block logic.

The address is the physical location where the field connections associated with the input connect to the controller. It will be manufacturer specific but typically is in the form of <Network>-<Controller>-<Circuit Board>-<Terminal Number>. For example 01-04-02-08 would designate the 8th input terminal on circuit board 2 in controller 4 on network 1.

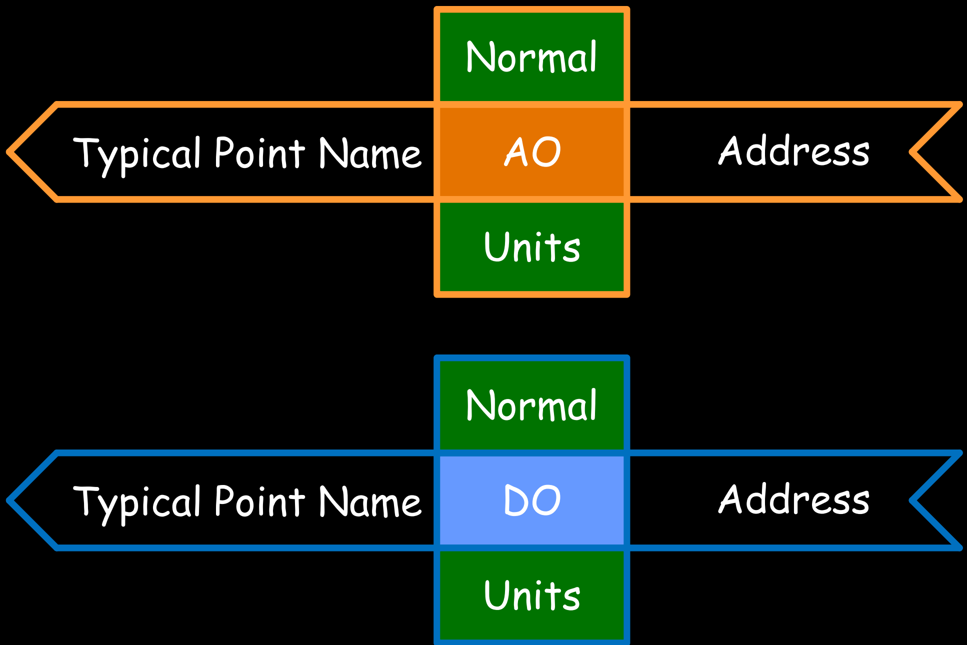
Knowing the address allows you to correlate which physical wires out in the field are associated with an input logic block if you are troubleshooting.

- AI = Analog Input
- DI = Digital Input, also referred to as a Binary Input or BI.

Back to Contents

DDC Logic Function Blocks

Output Point



Output Point – Transmits a value from the function block logic to a physical output channel in the controller.

The address is the physical location where the field connections associated with the output connect to the controller. It will be manufacturer specific but typically is in the form of <Network>-<Controller>-<Circuit Board>-<Terminal Number>. For example 02-018-01-12 would designate the 12th output terminal on circuit board 1 in controller 18 on network 2.

Knowing the address allows you to correlate which physical wires out in the field are associated with an input logic block if you are troubleshooting.

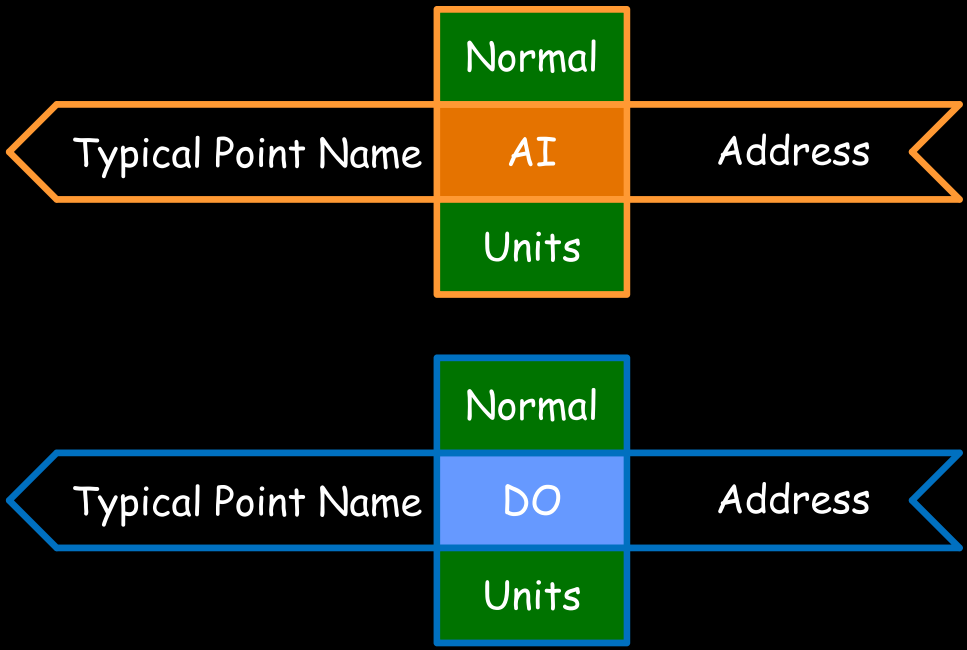
- *AO = Analog Output*
- DO = Digital Output, also referred to as a Binary Output or BO.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Output Point



Output Point (continued) :

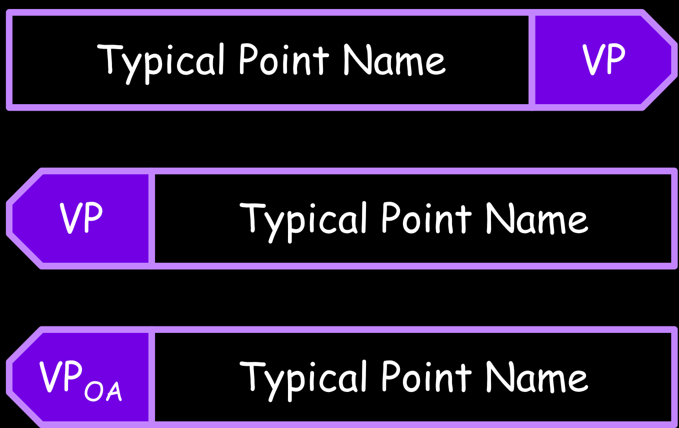
- "Normal" = The output's "normal" state; i.e. the position it would assume if power and control signals were removed. Options include:
  - NO = Normally Open
  - NC = Normally Closed
  - Lst = Last state (no spring return)
- "Units" = Engineering unit for the output. For analog outputs, examples include psig, 4-20 ma, 2-10 vdc, and Tristate. For digital points, examples include Pilot Relay, Low Voltage, and EP Switch

The actuator normal position and units are typically covered by the point list. But they can impact the control logic so it can be helpful to show them on the logic diagram. For instance, the normal position of an output can impact the "Action" (direct or reverse acting) of a control process and it is frequently desirable to force an actuator to a fixed position for events like a safety trip or a system shut down.

[Back to Contents]

DDC Logic Function Blocks

Virtual Point

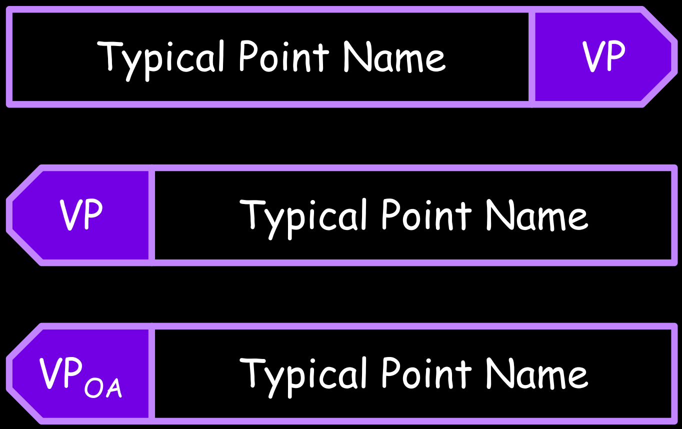


Virtual Point – A memory location used to store analog or digital data for use by the DDC function block logic or for transmitting over the DDC network for use by DDC function block logic in other controllers. "OA" indicates the value is operator adjustable for an operator with the correct access level. See the specifications and point list for additional information about operator access levels.

Back to Contents

DDC Logic Function Blocks

Virtual Point



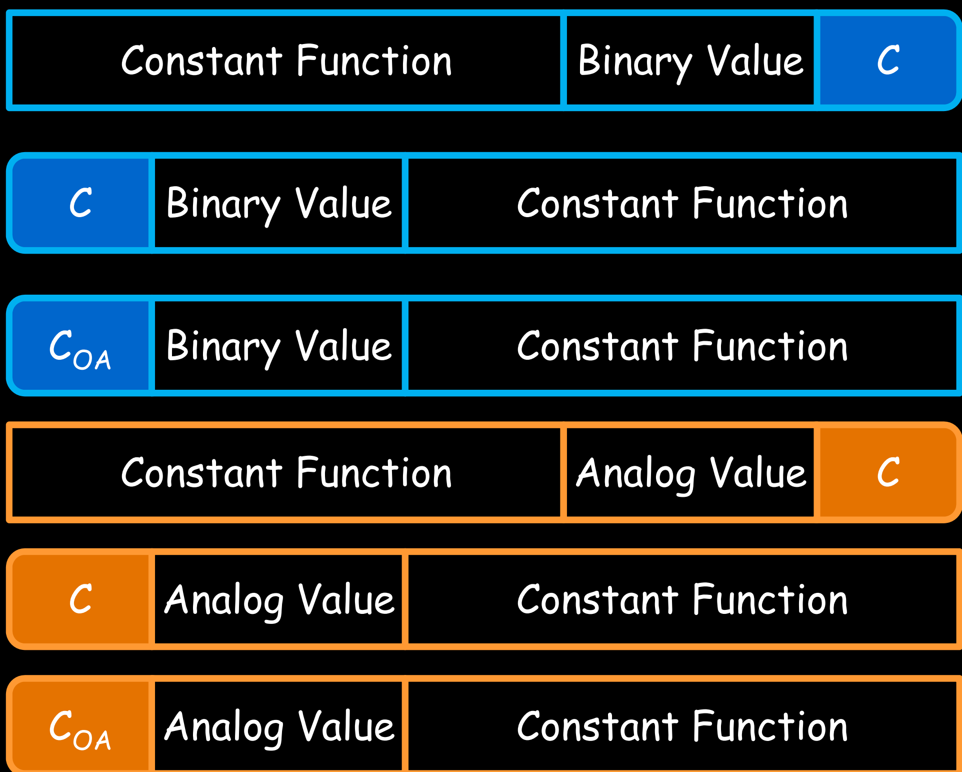
<u>Virtual Point (continued)</u> – Examples of virtual points include:

- Retransmitted information like an outdoor air temperature measured at one location and then transmitted to other controllers in the network.
- Calculated values like flow rates or kW.
- Totalizers to accumulate things like kWh or ton-hours
- Operator adjustable set points.
- Operator adjustable loop tuning parameters (for example, proportional, integral, and derivative gain values for a PID loop)

[Back to Contents](#)

DDC Logic Function Blocks

Constants



<u>Constant</u>– Fixed values used by the DDC logic function blocks. Examples of constants include.

- Set points that are "hard coded"
- Loop tuning parameters that are "hard coded"
- Engineering constants.
- Calibration offsets, slopes and intercepts.

The *Constant Function* specifies what the constant is used for and the *Constant Value* is the actual numerical value associated with the constant in the DDC function block logic.

Orange constants are analog; blue constants are digital.

The subscript "OA" after the constant function indicates that it is to be operator adjustable by an operator with the correct credentials; see specifications for details.

[Back to Contents](#)

DDC Logic Function Blocks

Connectors (Logical Wires)

Analog Connector – Indicates a logic connection transmitting an analog value between DDC logic function blocks.

Digital (Binary) Connector – Indicates a logic connection transmitting a digital (a.k.a. binary) value between DDC logic function blocks.

Virtual Connector – Indicates a logic connection transmitting virtual point information between DDC logic function blocks.

Constant Connector – Indicates a logic connection transmitting a constant to DDC logic function blocks.

Connectors Cross – The "hump" indicates that one connector is crossing the other.

Note that the "Z" shaped connector is a special MS Office object that, if used to connect other objects, will reconfigure itself to keep the objects connected if you move them. You can tell that the "Z" connector and the straight line connectors are truly connected to other objects if

(Continued on the next slide)

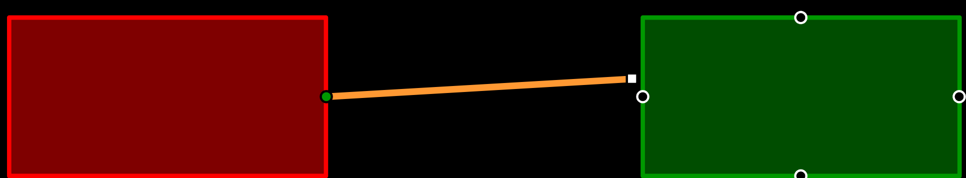
Back to Contents

DDC Logic Function Blocks

Connectors (Logical Wires)

The red rectangle is connected to the orange connector (the green dot indicates this) but the green rectangle is not (as indicated by the white square).

Note that the straight and "Z" shaped connectors can literally "grab" onto other shapes and will stay connected to them when you move the shape. You can tell the connectors have "grabbed" the shape if their end points (when you select them) turn into green circles instead of white squares, as illustrated to the left.



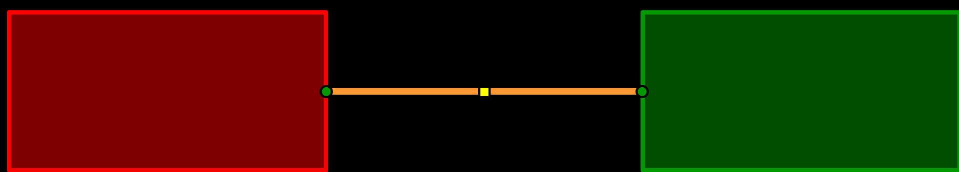
Different shapes have different connection points and they are not always where you would expect. For instance, the connection points on a rectangle are on the middle of each side, not the corner. The show up as small black circles when you drag the end of a connector near the shape, as illustrated to the left.

(Continued on the next slide)

Back to Contents

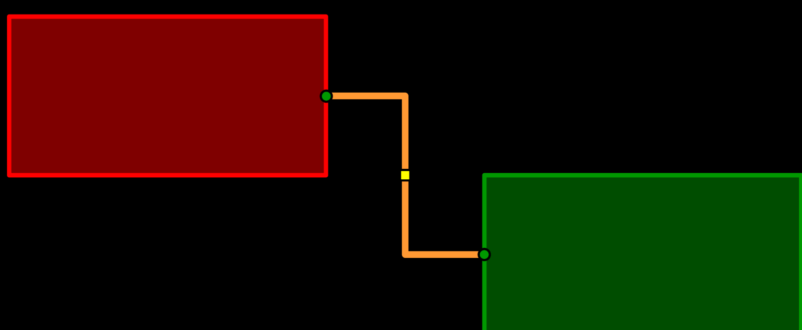
DDC Logic Function Blocks

Connectors (Logical Wires)



Even though the orange line looks like a straight connector, it is actually a "Z" connector without any bends in it. You can tell because when you select it, there is a little yellow square in the middle

Note also that the "Z" shaped connector is a special connector that will reconfigure its offset to keep the objects connected if you move them. In other words, if you offset two objects connected with a straight connector, it will slope to keep the objects connected. But if you use a "Z" connector, it will add one or more 90° bends to itself to keep the objects connected. You can shift the offsets by dragging the little yellow square that shows up when you select the connector.



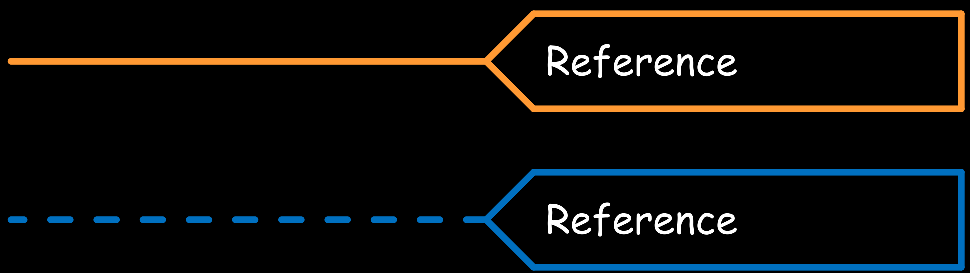
In this example, sliding the green square down and to the left (relative to the arrangement above) resulted in the orange connector adding two offsets in order to keep the shapes connected. You can shift the location of the offset by using the mouse to slide the yellow square back and forth.

(Continued on the next slide)

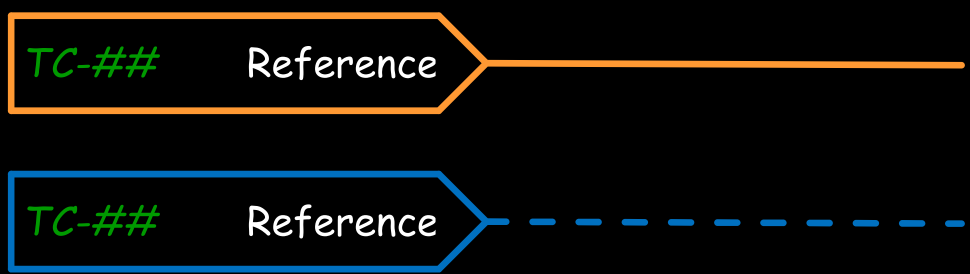
Back to Contents

DDC Logic Function Blocks

Reference Flags



Reference Origin; Reference To Another Sheet



Reference From Another Sheet

Reference Flag – Allows interconnection of DDC function blocks with out logical wires as illustrated in the next slide.

- Reference – A descriptive name for the reference such as "Building OAT", "Zone 1 Temperature", etc.
- *TC-##* - The drawing sheet of origin for the reference

(Continued on the next slide)

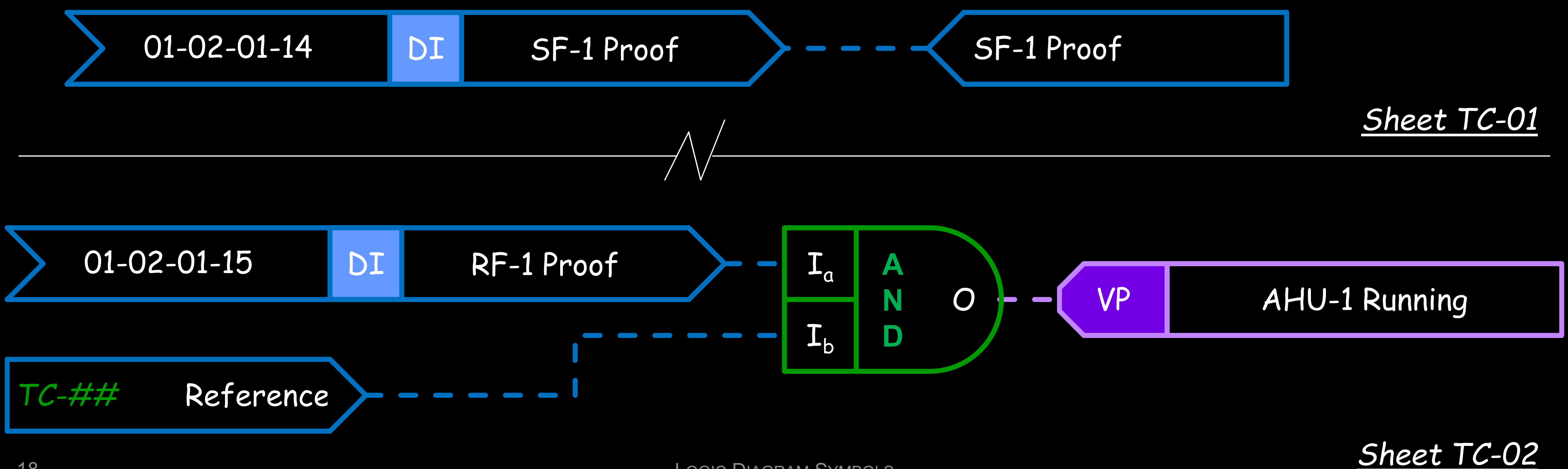
Back to Contents

DDC Logic Function Blocks

Reference Flags

In the example below, the digital input associated with SF-1 is located on sheet TC-01 and will be used by multiple logic functions, including an "And" logic block on sheet TC-02.

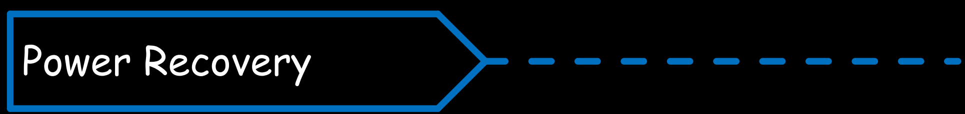
On sheet TC-02, it is referenced along with the digital input associated with RF-1 by an AND logic block. If both inputs are "On", then the virtual point indicating the SF1 System is running is turned on by the AND logic.



Back to Contents

DDC Logic Function Blocks

Power Failure Reference Flag

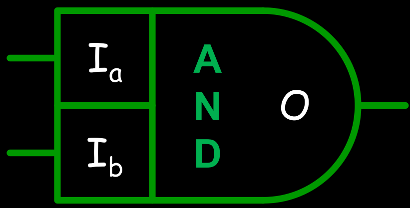


Power Failure Reference Flag – This flag is turned on when power is applied to the controller. The intended use is to trigger a power failure recover sequence to be executed after the controller experiences a power failure and is recovering from that event.

Back to Contents

DDC Logic Function Blocks

Logical "AND"



Logical AND – The output is "On" or "True" if both all inputs are "On" or "True".

Truth Table

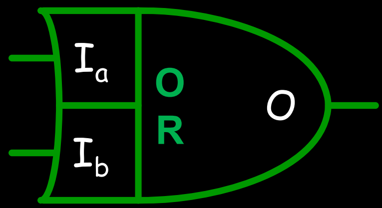
- The "dot" (.) is Boolean notation for "AND"
- T = True or On or 1
- F = False or Off or 0

I_a	I_b	$O=I_a.I_b$
F	F	F
F	T	F
T	F	F
T	T	T

Back to Contents

DDC Logic Function Blocks

Logical "OR"



Logical OR – The output is "On" or "True" if any input is "On" or "True".

Truth Table

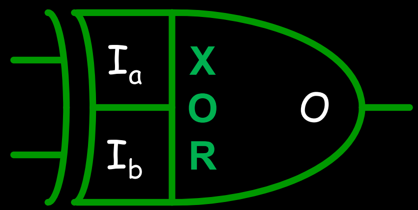
- The "plus" (+) is Boolean notation for OR
- T = True or On or 1
- F = False or Off or 0

I_a	I_b	$O=I_a+I_b$
F	F	F
F	T	T
T	F	T
T	T	T

Back to Contents

DDC Logic Function Blocks

Logical “EXCLUSIVE OR”



Logical EXCLUSIVE OR – The output is “On” or “True” if only one input is “On” or “True.

Truth Table

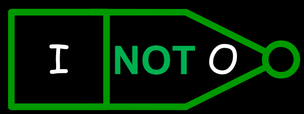
- The “circled plus” (⊕) is Boolean notation for EXOR
- T = True or On or 1,
- F = False or Off or 0

I_a	I_b	$O = I_a \oplus I_b$
F	F	F
F	T	T
T	F	T
T	T	F

[Back to Contents](#)

DDC Logic Function Blocks

Logical "NOT" or "INVERSE"



Logical NOT – The output is inverted from the input; for example the output is "On" if the input is "Off".

Truth Table

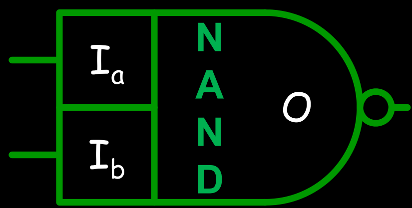
- The "bar" ($\overline{I}$) is Boolean notation for NOT
- T = True or On or 1,
- F = False or Off or 0

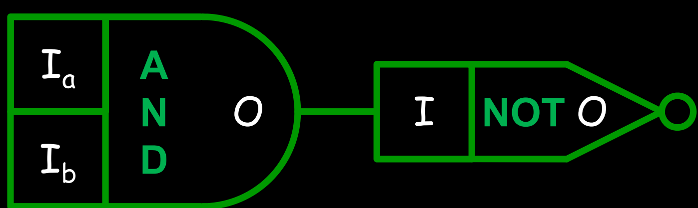
I	$O=\overline{I}$
F	T
T	F

Back to Contents

DDC Logic Function Blocks

Logical “NAND”; Combining Functions in One Symbol





Logical NAND – Combines the functions of “NOT” and “AND” into one function block.

Truth Table

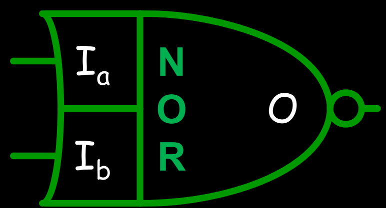
- The “dot” (.) is Boolean notation for “AND”
- T = True or On or 1
- F = False or Off or 0

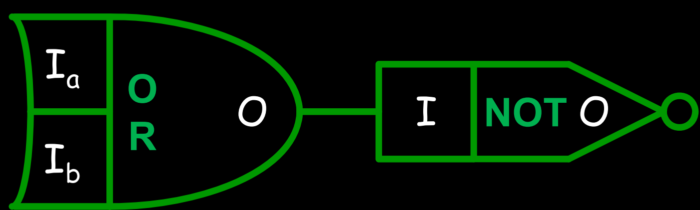
I_a	I_b	$O=I_a.I_b$	$O=\overline{I_a.I_b}$
F	F	F	T
F	T	F	T
T	F	F	T
T	T	T	F

Back to Contents

DDC Logic Function Blocks

Logical "NOR"





__Logical NOR__ – Combines the functions of "NOT" and "OR" into one function block.

Truth Table

- T = True or On or 1
- F = False or Off or 0

I_a	I_b	$O=I_a+I_b$	$O=\overline{I_a+I_b}$
F	F	F	T
F	T	T	F
T	F	T	F
T	T	T	F

Back to Contents

DDC Logic Function Blocks

If

IF	I	Condition	SP	"Units"	O

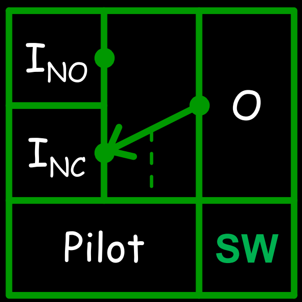
__If__ – The output is turned "On" if the input meets the indicated condition relative to the set point

- I = Input
- Condition = A comparison that will be made between the input and the set point such as:
  - Equal (=)
  - Greater than (>)
  - Less than (<)
  - Greater than or equal to (>=)
  - Less than or equal to (<=)
- SP = Set point; the information that will be compared to the input.
- "Units" = Unit of measure associated with SP
- O = Output – "On" if the condition is "True", otherwise "Off" (binary/digital).

Back to Contents

DDC Logic Function Blocks

Piloted Switch – Two Inputs, One Output



__Piloted Switch – 2 Inputs, 1 Output__ – The output is connected to one input or the other input depending on the status of a digital (binary) pilot signal.

If the pilot signal is "True" or "On" or "1", I_{NO} is connected to output *O*.

If the pilot signal is "False" or "Off" or "0", I_{NC} is connected to output *O*. The block is shown in the "Normal" position.

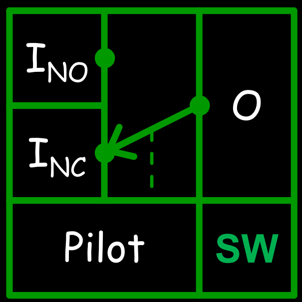
- I_{NC} = Normally Closed; the connection between this input and the output is closed unless the pilot is "On".
- I_{NO} = Normally Open; the connection between this input and the output is open unless the pilot is "On".
- *O* = Output
- Pilot = Binary pilot signal

(Continued on the next slide)

[Back to Contents]

DDC Logic Function Blocks

Piloted Switch – Two Inputs, One Output



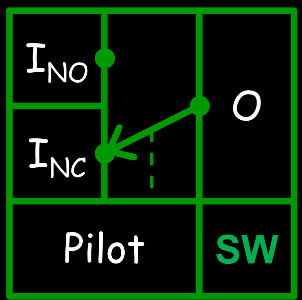
Piloted Switch – 2 Inputs, 1 Output (continued) – This type of logic is also called a "Relay" because its function is identical to how a physical electrical relay with a "Form C" contact works. Specifically, if you energize the relay coil (the pilot signal), the voltage on the normally open terminal is connected to the output or common terminal. If you de-energize the relay coil, the voltage on the normally closed terminal is connected to the output or common terminal.

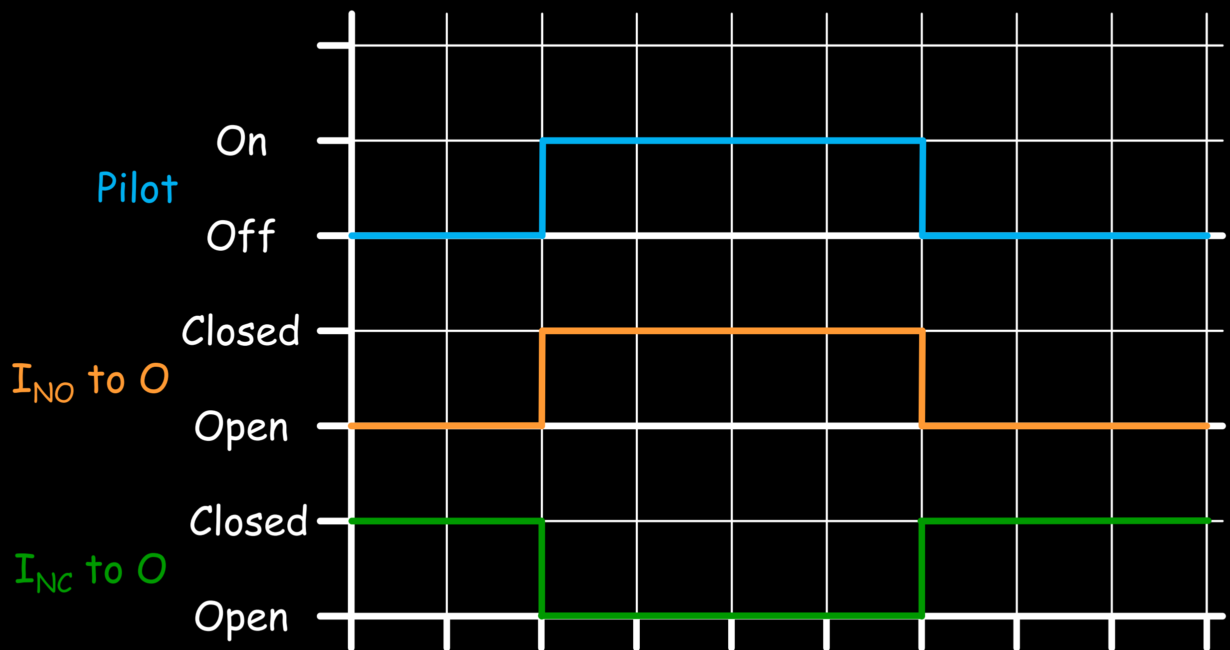
It is also identical to how a physical electro-pneumatic switch works. Specifically, if the electrical circuit to the coil is energized (the pilot signal), the air pressure on the normally open air port is connected to the output or common air port. If the electrical circuit to the coil is de-energized, the air pressure on the normally closed air port is connected to the output or common air port.

Back to Contents

DDC Logic Function Blocks

Piloted Switch – Two Inputs, One Output

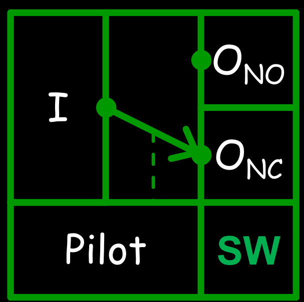




Back to Contents

DDC Logic Function Blocks

Piloted Switch – One Input and Two Outputs



Piloted Switch – 1 Output, 2 Inputs – This is very similar to the Piloted Switch, 2 Inputs and 1 Output.

If the pilot signal is "True" or "On" or "1", O_{NO} is connected to input I.

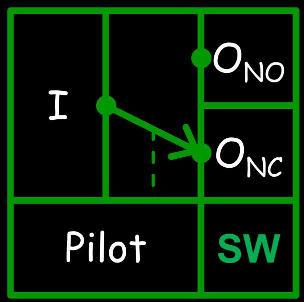
If the pilot signal is "False" or "Off" or "0", O_{NC} is connected to input I. The block is shown in the "Normal" position.

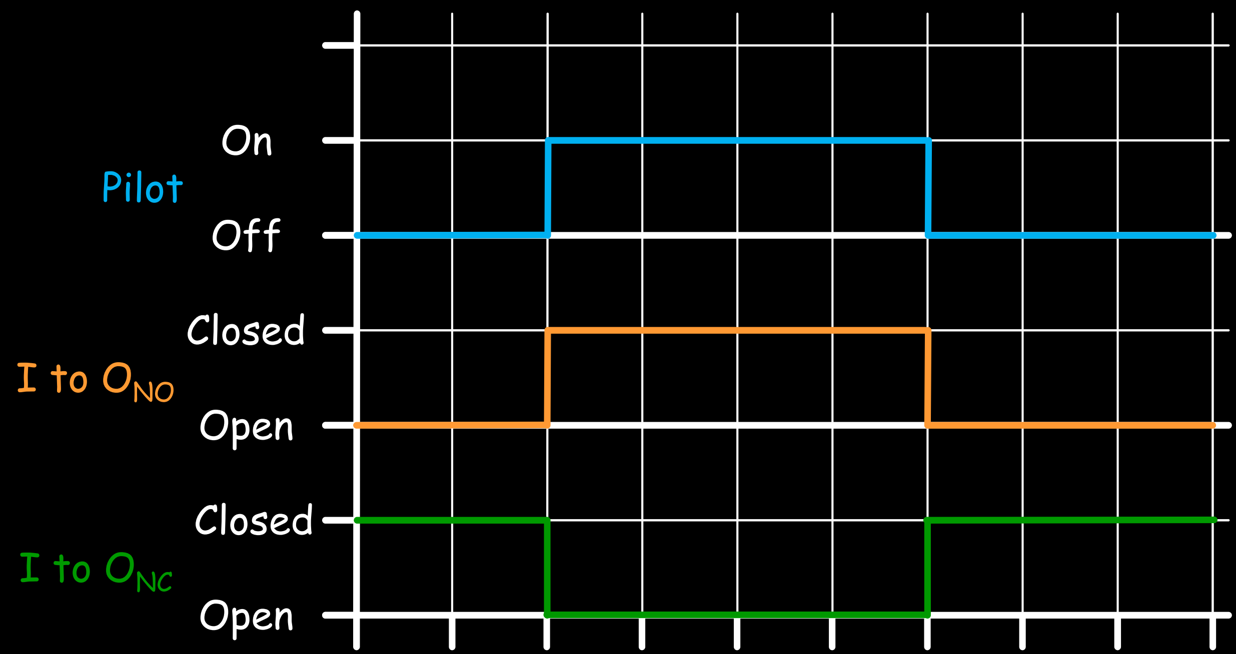
- O_{NC} = Normally Closed; the connection between this output and the input is closed unless the pilot is "On".
- O_{NO} = Normally Open; the connection between this output and the input is open unless the pilot is "On".
- I = Input
- Pilot = Binary pilot signal

Back to Contents

DDC Logic Function Blocks

Piloted Switch – Two Inputs, One Output

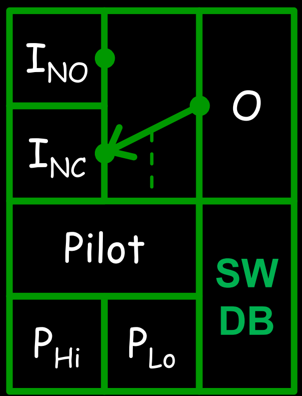




Back to Contents

DDC Logic Function Blocks

Piloted Switch with Dead Band – Two Inputs and One Output



Piloted Switch with Dead Band – 2 Inputs and 1 Output – The output is connected to one input or the other input depending on the status of the analog Pilot input.

If the Pilot input is above the value associated with P_{Hi}, I_{NO} is connected to output *O* and remains connected to output *O* until the pilot signal drops below P_{Lo}.

If the Pilot input is below the value associated with P_{Lo}, I_{NC} is connected to output *O* and remains connected to output *O* until the pilot signal rises above P_{Hi}.

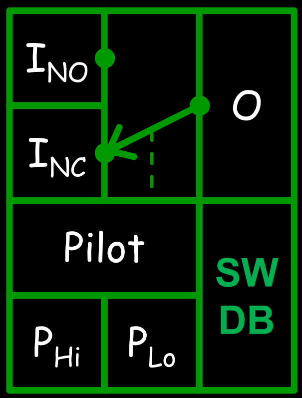
The block is shown in the "Normal" position.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Piloted Switch with Dead Band – Two Inputs and One Output



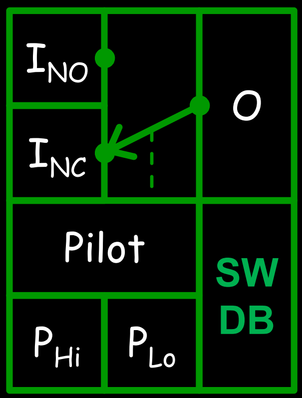
Piloted Switch with Dead Band (continued)

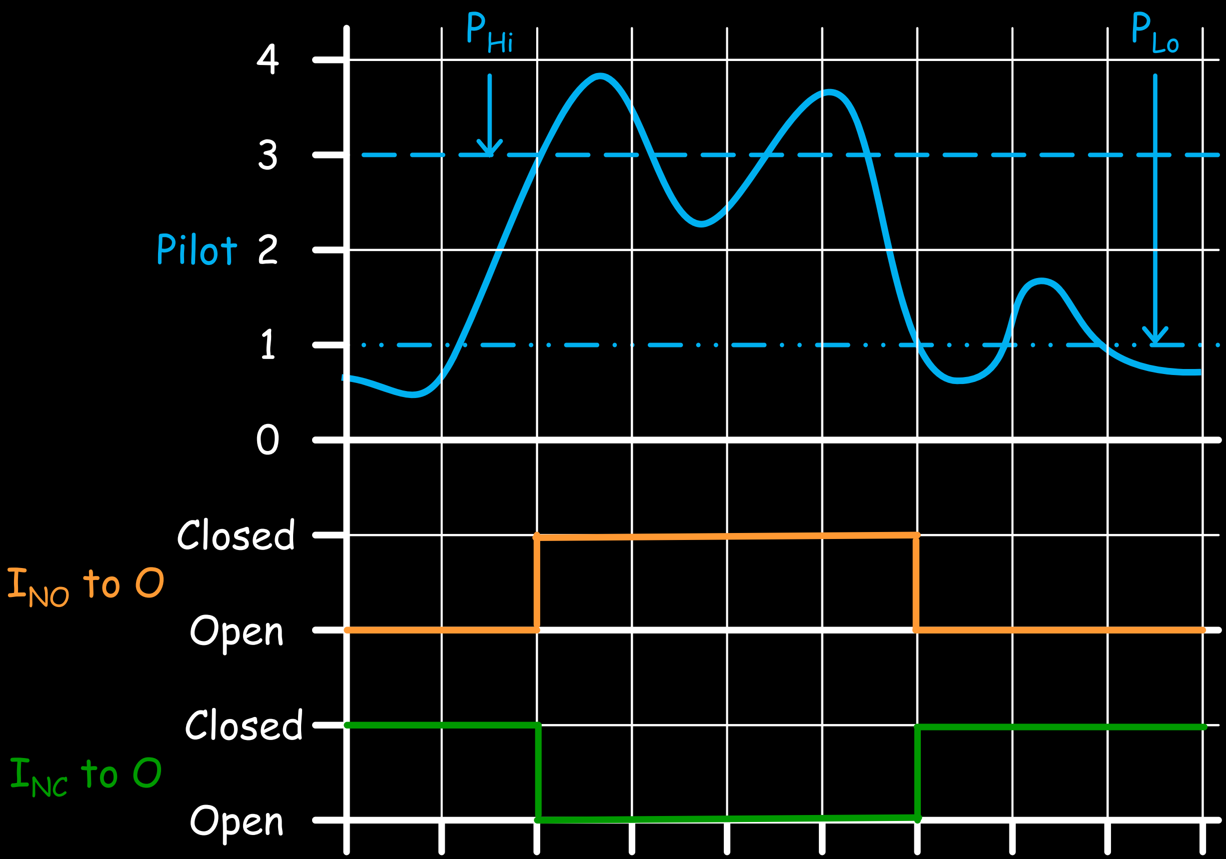
- I_{NC} = Normally Closed; operation as described previously.
- I_{NO} = Normally Open; operation as described previously.
- O = Output.
- Pilot = Analog pilot signal.
- P_{Hi} = Dead band upper limit.
- P_{Lo} = Dead band lower limit.

Back to Contents

DDC Logic Function Blocks

Piloted Switch with Dead Band – Two Inputs and One Output

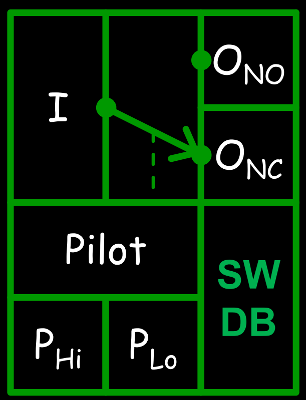




[Back to Contents]

DDC Logic Function Blocks

Piloted Switch with Dead Band – One Input and Two Outputs



Piloted Switch with Dead Band – 1 Input and 2 Outputs – The input is connected to one output or the other output depending on the status of the analog Pilot input.

If the Pilot input is above the value associated with P_{Hi}, O_{NO} is connected to input I and remains connected to input I until the pilot signal drops below P_{Lo}.

If the Pilot input is below the value associated with P_{Lo}, O_{NC} is connected to input I and remains connected to input I until the pilot signal rises above P_{Hi}.

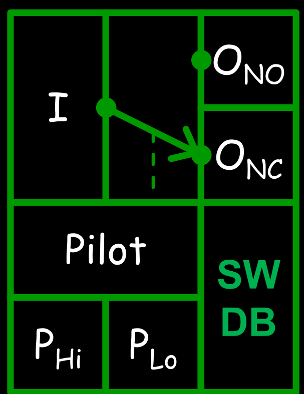
The block is shown in the "Normal" position.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Piloted Switch with Dead Band – One Input and Two Outputs



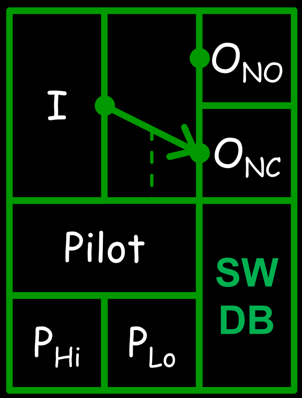
Piloted Switch with Dead Band (continued)

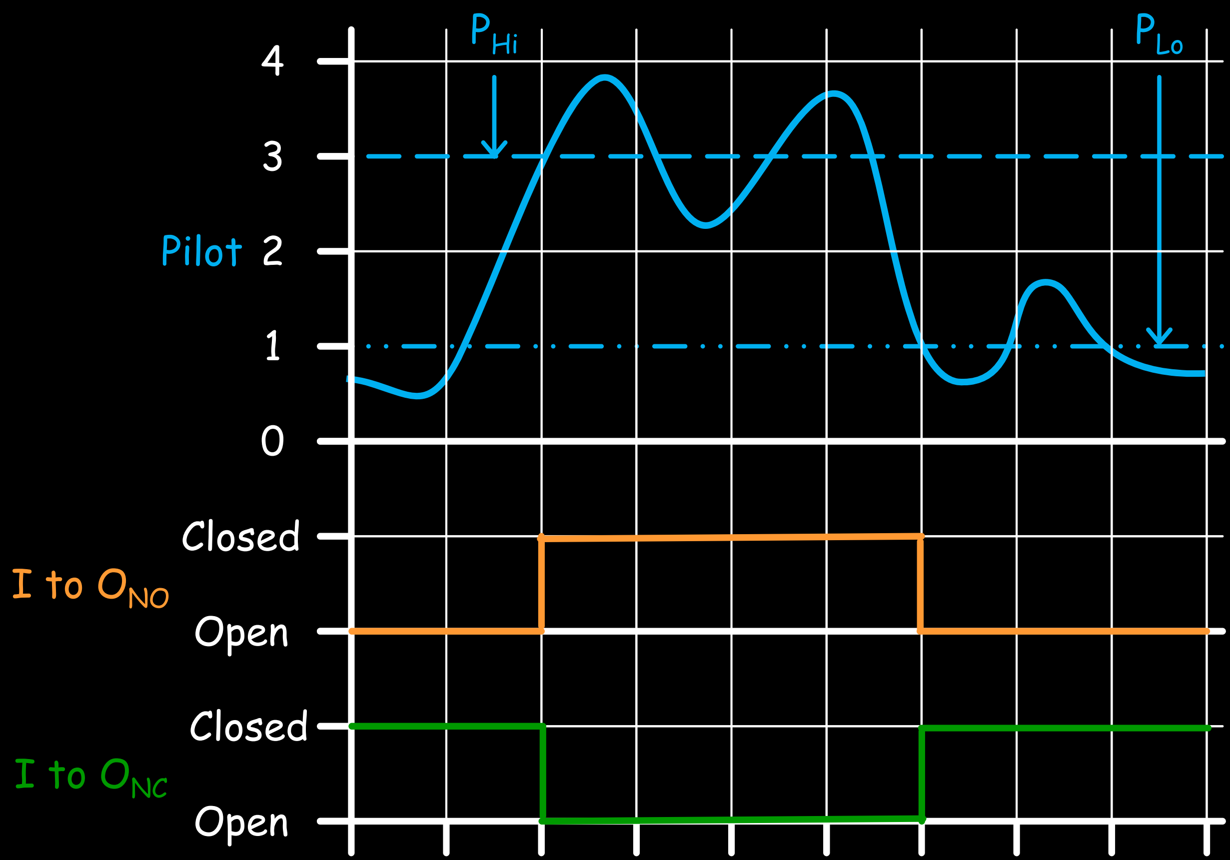
- O_{NC} = Normally Closed; operation as described previously.
- O_{NO} = Normally Open; operation as described previously.
- I = Input.
- Pilot = Analog pilot signal.
- P_{Hi} = Dead band upper limit.
- P_{Lo} = Dead band lower limit.

Back to Contents

DDC Logic Function Blocks

Piloted Switch with Dead Band – Two Inputs and One Output





Back to Contents

DDC Logic Function Blocks

PID Loop (Proportional + Integral + Derivative Loop)

PID Loop – Proportional plus Integral plus Derivative control process.

- SP = Set point
- I = Input
- Go = Anti-windup input; used to shut down the control process and set it to the GO_{OFF} value
- I_{GOff} = The value the output should be set to when the process is shut down in %
- K_P = Proportional gain; adjusted during Cx
- K_I = Integral gain; adjusted during Cx
- K_D = Derivative gain; adjusted during Cx
- O = Output
- "Action" = The control process action ; DA for "direct acting" or RA for "reverse acting".

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

PID Loop (Proportional + Integral + Derivative Loop)

PID Loop (continued):

- "Units" – The engineering units associated with I_{GOff} such as milli-amps, volts, psig, etc.
- Integral gain impacts the process output based on how long the proportional error has persisted.
- Derivative gain impacts the process output based on the rate of change of proportional error.
- Values of "0" for K_I and K_D indicate the design intent is a proportional only control process.
- A value of "0" for K_D indicate the design intent is a proportional plus integral control process; apply derivative only if deemed necessary during Cx.
- Direct acting means that when the input deviates above set point, the output increases.
- Reverse acting means that when the input deviates above set point, the output decreases.

Back to Contents

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

Floating Control – A control process where no action is taken if input I is between the set point SP plus or minus the dead band I_{DB}.

If input I gets outside of the dead band, then the appropriate output O_{Opn} or O_{Cls} is energized to drive the process back inside the dead band.

The maximum on and off times for either O_{Opn} or O_{Cls} are based on inputs I_{On} and I_{Off}. The purpose of this is to allow the control system to react to a deviation outside of the dead band but then to allow the process some time to react to the changes made by the control process.

I_{On}, I_{Off}, and I_{DB} are tuned during the commissioning process as required to provide stable operation.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

Floating Control (continued)

- SP = Set point
- I = Input
- Go = Shuts down the control process if it is "Off".
- I_{GOff} = The position of the final control element when the process is "Off" in terms of percent open.
- I_{DB} = The allowable deviation above or below set point SP before the control process responds.
- I_{On} = The maximum on time that O_{inc} or O_{dec} will be allowed to remain active when set point SP shifts outside the dead band created by I_{DB} before stopping actuator motion to allow time for the process to react in seconds.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

Floating Control (continued)

- I_{Off} = The maximum time that O_{inc} or O_{dec} will be allowed to remain inactive if set point SP is outside the dead band created by I_{DB}.
- O_{Opn} = The output that is energized to drive the actuator open.
- O_{Cls} = The output that is energized to drive the actuator closed.
- "Action/Normal" = The control process action and the normal position of the final control element. Action can be DA for "direct acting" or RA for "reverse acting". Normal position can be NO for "normally open" or NC for "Normally Closed".

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Floating Control

Floating Control (continued)

- RT=### = Actuator full stroke run time. This is used to establish the position of the output when the process is off and to sequence other floating actuators if the process controls more than one.

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

__Floating Control (continued)__

Because a floating control process uses two digital/binary outputs and a tri-state actuator to mimic an analog output and modulating actuator, the terms “Direct Acting” (DA) and “Reverse Acting” (RA) need to be defined in the context of the “normal” position of the actuator served by the loop.

The design of this function block assumes:

1. A spring return actuator with the normal position being the position the actuator would assume if power was removed and the spring drove the actuator to its limit.
2. If the actuator is not spring return, then there is really no “normal” position and removing power causes the actuator to retain its last state.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

__Floating Control (continued)__:

3. A direct acting process is a process where:
   - A normally open final control element would need to be driven closed to drive the process variable towards set point when it deviates above set point. A normally open heating valve serving a preheat coil is an example.
   - A normally closed final control element would need to be driven open to drive the process variable towards set point when it deviates above set point. A normally closed valve serving a cooling coil is an example.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

<u>Floating Control (continued)</u>:

4. A reverse acting process is a process where:
   - A normally closed final control element would need to be driven open to drive the process variable towards set point when it deviates below set point. A normally closed heating valve serving a preheat coil is an example.
   - A normally open final control element would need to be driven closed to drive the process variable towards set point when it deviates below set point. A normally open valve serving a cooling coil is an example.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Floating Control

This function block is designed to work with an electric tri-state actuator. A tri-state actuator is controlled by two digital outputs, one of which drives the actuator towards the open position and the other of which drives the actuator towards the closed position.

__Floating Control (continued)__:

5. The off position established by I_{GOff} is achieved by driving the actuator in the closed direction for the full run time and then driving to the desired percentage based on percent of full stroke run time.

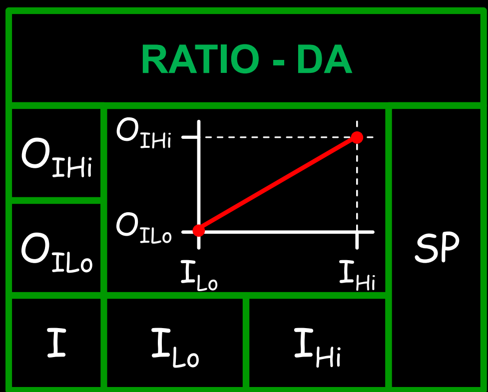
   For example, if I_{GOff} = 50% and RT = 30 seconds, when the Go input is set to "Off" the O_{Cls} output would be energized for 30 seconds and then the O_{Opn} output would be energized for 15 seconds (50% of 30 seconds).

6. If a second floating actuator needs to be sequenced with this floating actuator, the sequencing should be based on the run time in the open or closed direction exceeding the RT value for this actuator.

Back to Contents

DDC Logic Function Blocks

Direct Acting Ratio



Direct Acting Ratio – The output is adjusted linearly based on the relationship shown to the right. In a direct ratio, an increase in the input causes an increase in the output

$$y = (m \times x) + b$$

Where:

$y =$ The calculated output or "O" based on the current input or "I"

$m =$ The slope of the line defining the reset

$$= \left(\frac{\Delta y}{\Delta x}\right) = \left(\frac{SP_{IHi} - SP_{ILo}}{I_{Hi} - I_{Lo}}\right)$$

$SP_{IHi} =$ The desired set point when the input is at the I_{Hi} value

$SP_{ILo} =$ The desired set point when the input is at the I_{Lo} value

$I =$ The input to the calculation, usually a current process variable such as supply temperature or system differential pressure.

$b =$ The y axis intercept when x is equal to zero

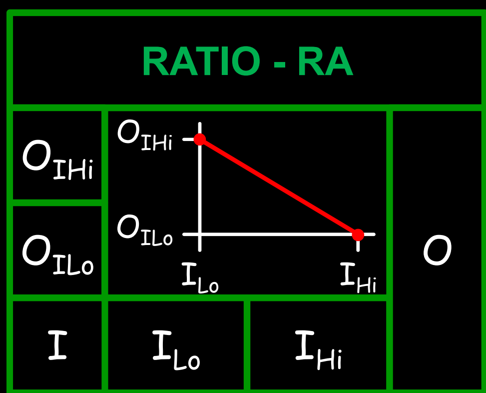
$= y\text{-}(m \times x)$

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Reverse Acting Ratio



Reverse Acting Ratio – The output is adjusted linearly based on the relationship shown to the right. In a reverse ratio, an increase in the input causes an decrease in the output

$$y = (m \times x) + b$$

Where:

$y =$ The calculated output or "O" based on the current input or "I"

$m =$ The slope of the line defining the reset

$$= \left(\frac{\Delta y}{\Delta x}\right) = \left(\frac{SP_{IHi} - SP_{ILo}}{I_{Hi} - I_{Lo}}\right)$$

$SP_{IHi} =$ The desired set point when the input is at the I_{Hi} value

$SP_{ILo} =$ The desired set point when the input is at the I_{Lo} value

$I =$ The input to the calculation, usually a current process variable such as supply temperature or system differential pressure.

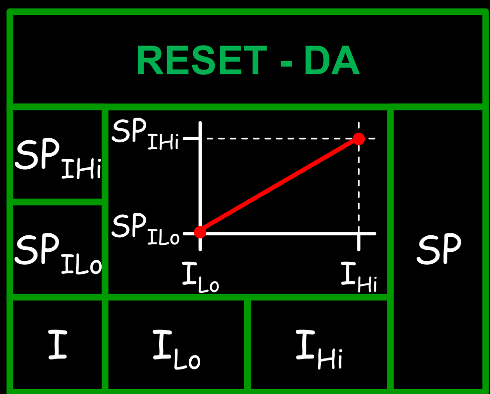
$b =$ The y axis intercept when x is equal to zero

$$= y - (m \times x)$$

Back to Contents

DDC Logic Function Blocks

Direct Acting Reset



__Direct Acting Reset__ – This is just a special case of the direct acting ratio block where the output is a set point

$$y = (m \times x) + b$$

Where:

$y =$ The calculated set point or "SP" based on the current input or "I"

$m =$ The slope of the line defining the reset

$$= \left(\frac{\Delta y}{\Delta x}\right) = \left(\frac{SP_{IHi} - SP_{ILo}}{I_{Hi} - I_{Lo}}\right)$$

$SP_{IHi} =$ The desired set point when the input is at the I_{Hi} value

$SP_{ILo} =$ The desired set point when the input is at the I_{Lo} value

$I =$ The input to the calculation, usually a current process variable such as supply temperature or system differential pressure.

$b =$ The y axis intercept when x is equal to zero

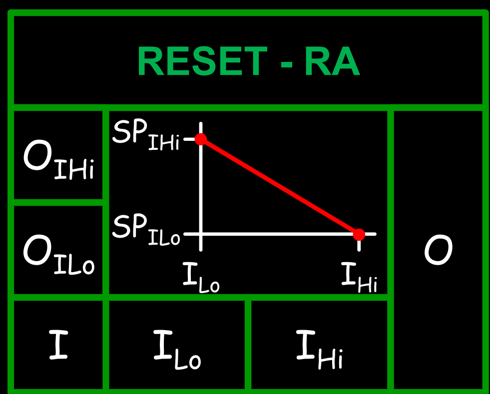
$$= y - (m \times x)$$

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Reverse Acting Reset



Reverse Acting Reset – This is just a special case of the reverse acting ratio block where the output is a set point

$y = (m \times x) + b$

Where:

$y =$ The calculated set point or "SP" based on the current input or "I"

$m =$ The slope of the line defining the reset

$$= \left(\frac{\Delta y}{\Delta x}\right) = \left(\frac{SP_{IHi} - SP_{ILo}}{I_{Hi} - I_{Lo}}\right)$$

$SP_{IHi} =$ The desired set point when the input is at the I_{Hi} value

$SP_{ILo} =$ The desired set point when the input is at the I_{Lo} value

$I =$ The input to the calculation, usually a current process variable such as supply temperature or system differential pressure.

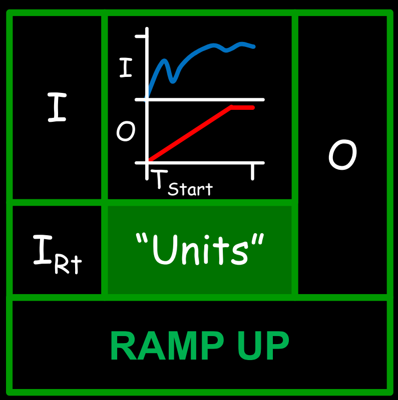
$b =$ The y axis intercept when x is equal to zero

$= y - (m \times x)$

Back to Contents

DDC Logic Function Blocks

Ramp Up



When combined with a Ramp Down logic block, this function can smooth out the response of a process to change. However, it will also introduce a permanent lag into the process, which can adversely impact the settling time after an upset.

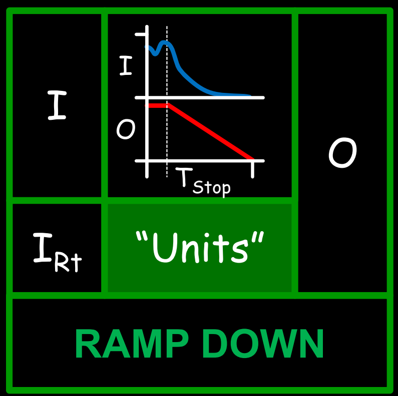
Ramp Up - Limits the rate at which the output O can increase, no matter how fast the input I is increasing.

- I = Input (analog)
- I_{Rt} = Rate at which the output O will be allowed to increase. Tune during Cx as required.
- "Units" = Unit of measure for the allowable rate change. Examples include:
  - Hz/sec (Hertz per second)
  - °F/min (Degrees Fahrenheit per minute)
  - in.wc./min. (inches water column per minute)
- O = Rate limited output (analog)

Back to Contents

DDC Logic Function Blocks

Ramp Down



When combined with a Ramp Up logic block, this function can smooth out the response of a process to change. However, it will also introduce a permanent lag into the process, which can adversely impact the settling time after an upset.

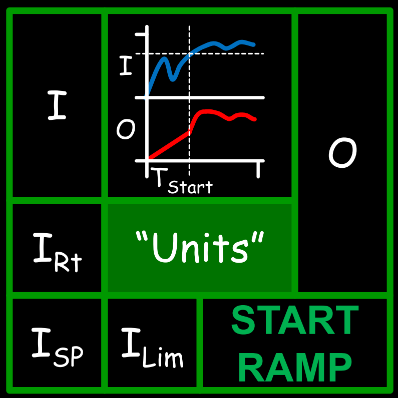
Ramp Down - Limits the rate at which the output *O* can decrease, no matter how fast the input I is decreasing.

- I = Input (analog)
- I_{Rt} = Rate at which the output *O* will be allowed to decrease. Tune during Cx as required.
- "Units" = Unit of measure for the allowable rate change. Examples include:
  - Hz/sec (Hertz per second)
  - °F/min (Degrees Fahrenheit per minute)
  - in.wc./min. (inches water column per minute)
- *O* = Rate limited output (analog)

Back to Contents

DDC Logic Function Blocks

Start Ramp



Start Ramp - Limits the rate at which the output O can increase, no matter how fast the input I is increasing until the input I is within a certain range (I_{Lim}) of the process set point (I_{SP}).

This allows a process to come on line gradually with out limiting its responsiveness once the set point is operating with-in the normal range.

If the ramp was always functional, then it would become a permanent lag in the process. Lags are the enemy of tight control because:

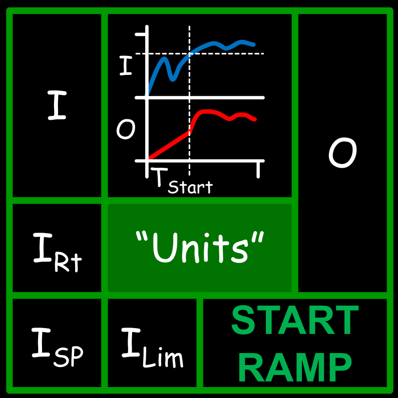
- They limit the ability of the control loop to minimize overshoot because it can not react fast enough due to the accumulated lags.
- They increase the settling time after an upset because the accumulated lags slow down the process reaction time.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Start Ramp



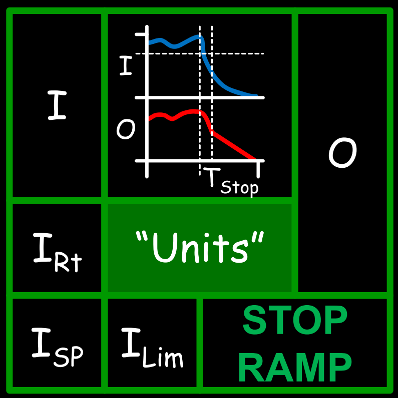
Start Ramp (continued) :

- I = Input (analog)
- I_{Rt} = Rate at which the output O will be allowed to increase. Field tune during Cx to prevent start-up problems like over-pressurization and static pressure safety trips.
- "Units" = Unit of measure for the allowable rate change. See Ramp Up logic block for examples
- I_{SP} = The set point associated with the process to which the ramp is being applied. Coordinate tuning during Cx with I_{Rt} tuning.
- I_{Lim} = The point at which the ramp will no longer limit the rate of output change relative to the input in percent of set point. Above the limit, output O will be the same as input I.
- O = Rate limited output (analog)

Back to Contents

DDC Logic Function Blocks

Stop Ramp

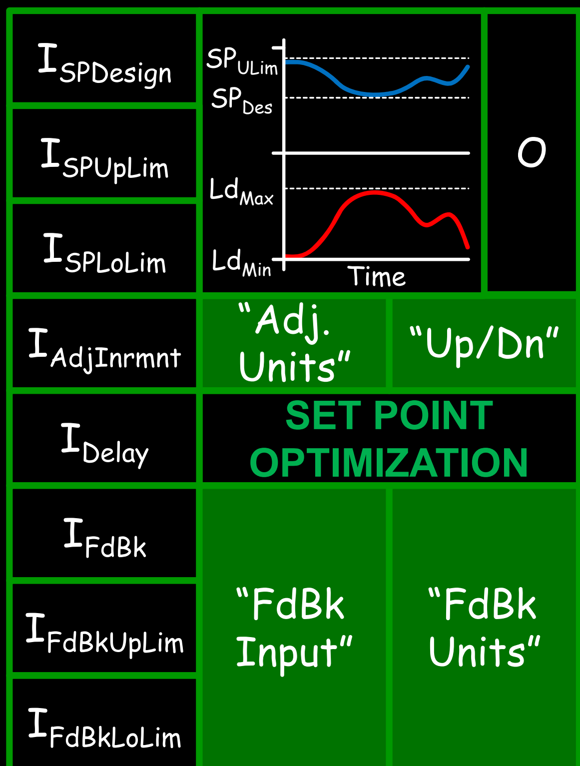


Stop Ramp – The shut down version of a start ramp.

- I = Input (analog)
- I_{Rt} = Rate at which the output *O* will be allowed to decrease. Field tune during *Cx* to prevent shut-down problems like undesirable pressure excursions in adjacent systems.
- "Units" = Unit of measure for the allowable rate change. See Ramp Up logic block for examples
- I_{SP} = The set point associated with the process to which the ramp is being applied. Coordinate tuning during *Cx* with I_{Rt} tuning.
- I_{Lim} = The point above which the ramp will no longer limit the rate of output change relative to the input in percent of set point. Above the limit, output *O* will be the same as input I.
- *O* = Rate limited output (analog)

DDC Logic Function Blocks

Set Point Optimization (Trim and Respond)



__Set Point Optimization__ – Optimizes a set point by making a small change to it (trimming it), observing the response, and then making another adjustment based on the reaction (responding).

- O = Output, analog value that represents the adjusted set point.
- $I_{SPDesign}$ = Input, an analog value that represents the design set point for the process.
- $I_{SPUpLim}$ = The maximum allowable set point.

  If the optimization adjustment direction is "Up", then this will be a different number from the design set point that is determined based on the operating requirements of the load served by the system.

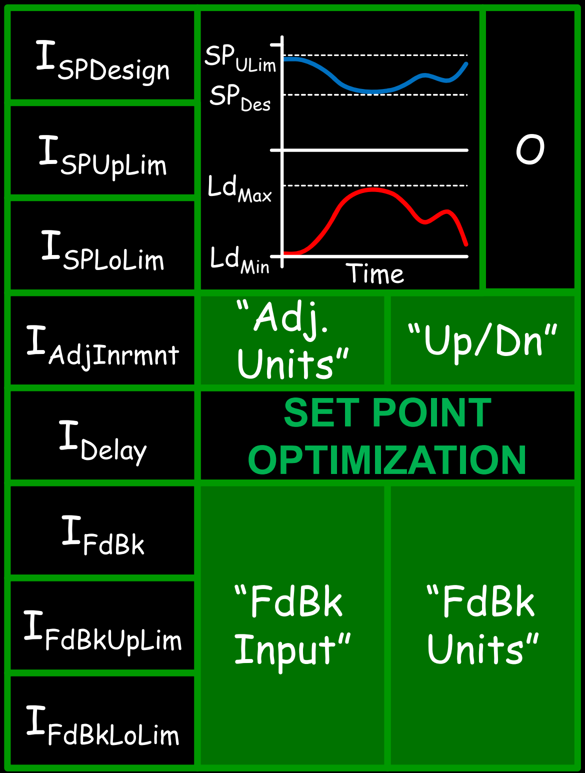
  If optimization adjustment direction is "Down", then this will be the design set point.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Set Point Optimization (Trim and Respond)



Set Point Optimization (continued):

- $I_{SPLoLim}$ = This is the minimum allowable set point. If the optimization adjustment direction is "Up", then this will be the design set point.

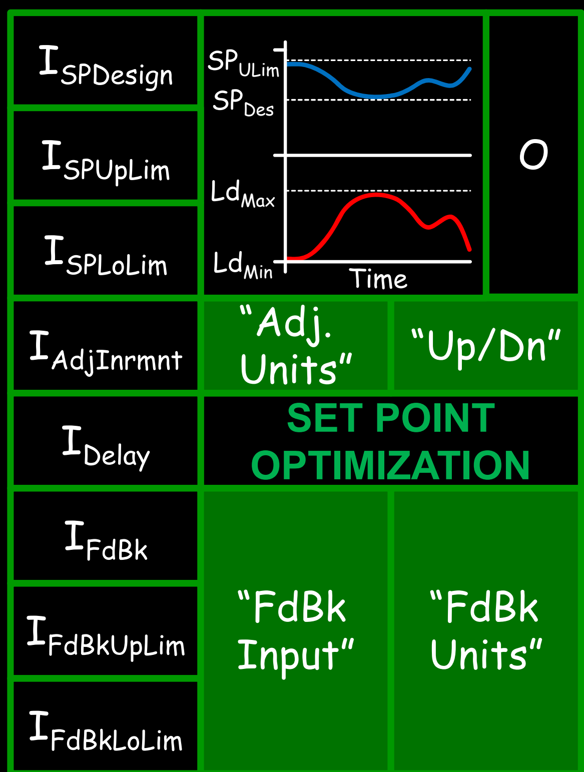
  If optimization adjustment direction is "Down", then this will be a different number from the design set point that is determined based on the operating requirements of the load served by the system.
- $I_{AdjInrmnt}$ = The incremental adjustment that will be added or subtracted from the set point based on the observed reaction.
- "Adj. Units" = The engineering units associated with $I_{AdjInrmnt}$. Examples include inches water column (in.w.c.), °F, psig, etc.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Set Point Optimization (Trim and Respond)



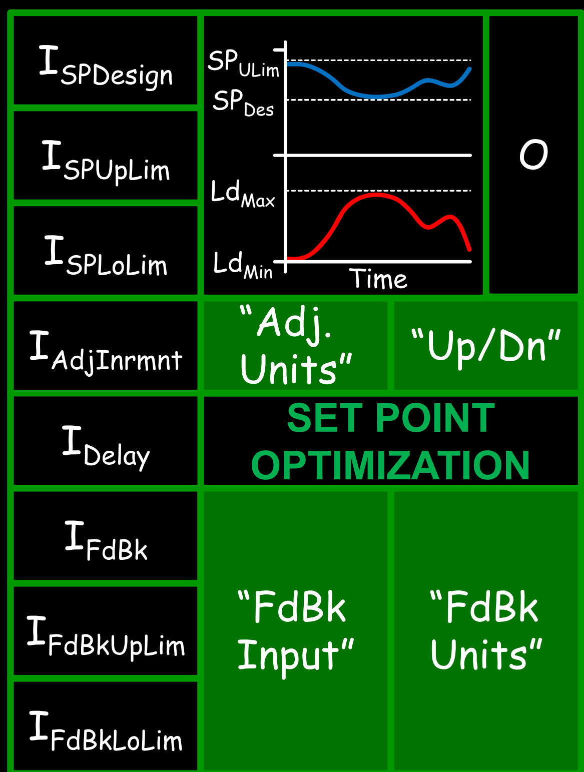
Set Point Optimization (continued):

- "Up/Dn" = Up or Down; this is the direction the set point is adjusted in to optimize it. For example, an AHU discharge temperature is typically adjusted up from the design value to optimize it but the discharge static pressure is typically adjusted down to optimize it.
- I_{Delay} = The delay time between adjustments.
- I_{FdBk} = The system parameter or logic used to assess the impact of an incremental adjustment and make a decision about what the next adjustment should be.

(Continued on the next slide)

DDC Logic Function Blocks

Set Point Optimization (Trim and Respond)



Set Point Optimization (continued):

- $I_{FdBkUpLim}$ = Feedback parameter upper limit.

  The process tries to adjust the set point to keep the feed back parameter inside an acceptable range.

  For instance, an air handling unit discharge temperature reset strategy might raise the leaving air temperature (the adjusted set point) to keep at least one or more reheat valves closed or nearly fully closed (the valves are the feedback parameter).

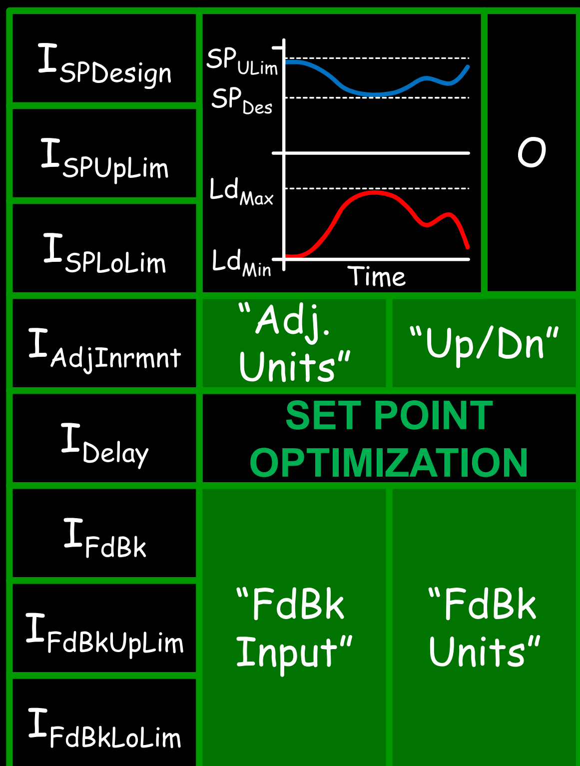
  This input defines the top of the range targeted by the process for the indicator it is monitoring.
- $I_{FdBkLoLim}$ = Feedback parameter lower limit, or the bottom of the range targeted by the process for the indicator it is monitoring.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Set Point Optimization (Trim and Respond)



Set Point Optimization (continued):

- "FdBk Input" = The type of parameter that is being monitored as feedback; for instance, reheat valve position or return temperature.
- "FdBk Units" = The engineering unit for the parameter being monitored for feedback; for instance, "percent open" or "°F".

$I_{AdjInrmnt}$, I_{Delay}, $I_{FdBkUpLim}$, and $I_{FdBkLoLim}$ are all fine-tuned during the Cx process to achieve stable operation.

The Cx process should also identify any rouge zones and temporarily remove them from the reset assessment analysis. Rouge zones should be referred to the designer for recommendations regarding how to deal with them so that they don't defeat the reset strategy but can meet the zone design requirements.

[Back to Contents](#)

DDC Logic Function Blocks

Calculation

CALC		Equation							O
I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}

Calculation – The output is set to the value calculated by the indicated equation using the indicated inputs

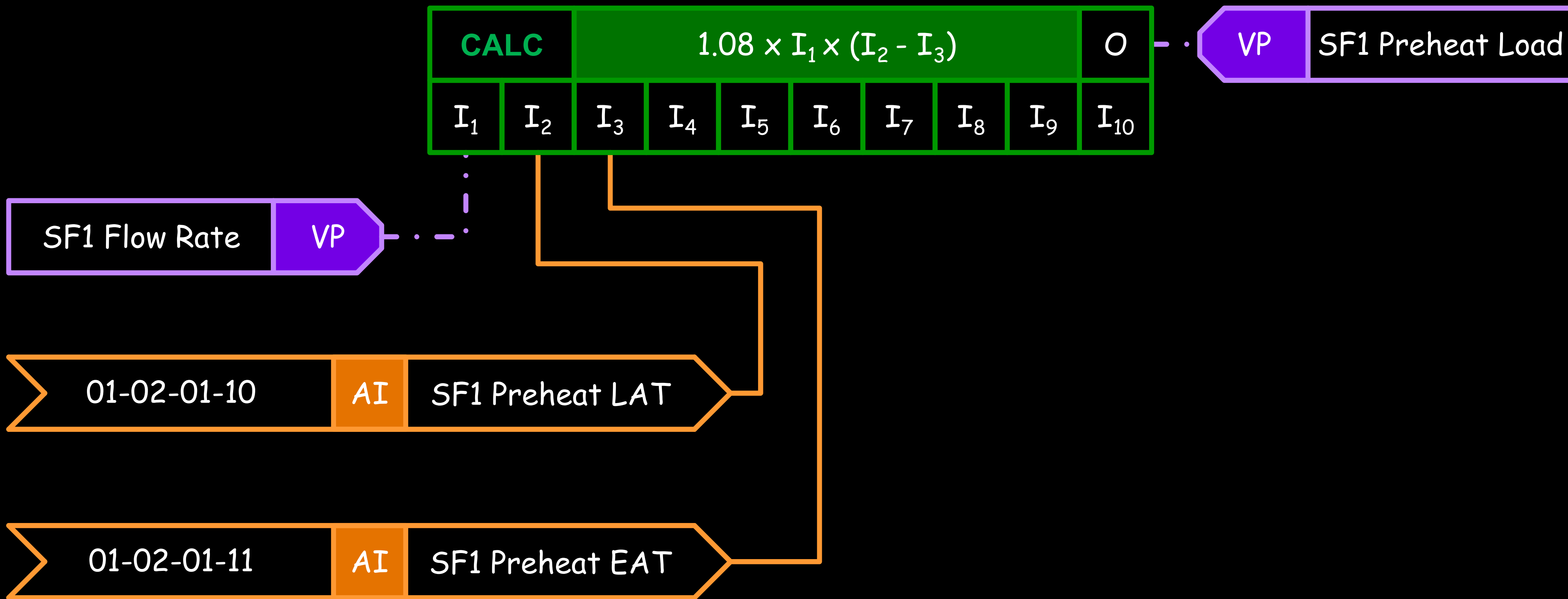
- I_1 - I_{10} = Inputs 1 through 10; unused inputs have no effect on the calculation.
- Equation = the mathematical relationship that is being used for the calculation. The example in the next slide is for the sensible heat equation.
- O = Output

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

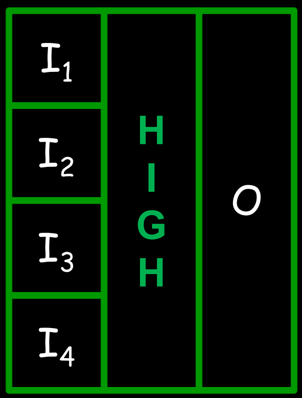
Calculation Example For $Q_{\text{Btu per hour}} = 1.08 \times Flow_{\text{Cubic Feet per Minute}} \times \left(Temp._{\text{Leaving, °F}} - Temp._{\text{Entering, °F}}\right)$



Back to Contents

DDC Logic Function Blocks

High Selector



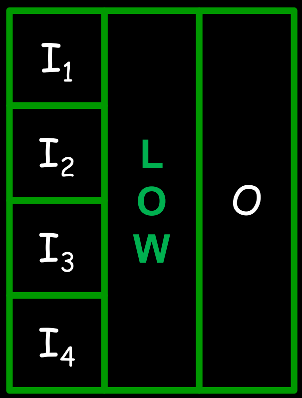
High Selector – Selects the highest input value and passes it to the output.

- I_1 – I_4 = Inputs
- *O* = Output

Back to Contents

DDC Logic Function Blocks

Low Selector



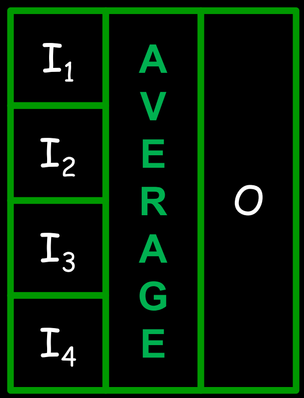
Low Selector – Selects the lowest input value and passes it to the output.

- I_1 – I_4 = Inputs
- O = Output

Back to Contents

DDC Logic Function Blocks

Average



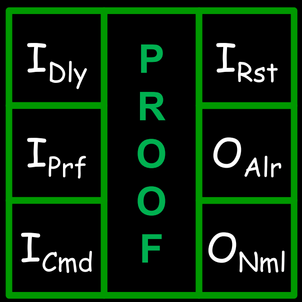
Average – The output is the mathematical average of the inputs. Unused inputs are not included in the average.

- I_1 – I_4 = Inputs
- O = Output

Back to Contents

DDC Logic Function Blocks

Proof of Operation



Proof of Operation – O_{Nml} is turned "On" if I_{prf} turns "On" with-in the time delay window established by I_{Dly} after I_{Cmd} is turned "On". If I_{Prf} does not turn "On" with-in the time delay window, then O_{Alr} is latched "On" and O_{Nml} is latched "Off".

Momentarily turning on I_{Rst} releases the latch, returning O_{Alr} and O_{Nml} to "Off" so the logic block is ready for the next cycle.

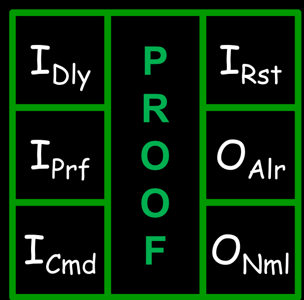
- I_{Dly} = Delay time allowed for the proof of operation signal to turn "On" after the start command is issued. See the specification and point list for point specific requirements.
- I_{Prf}= Connected to the point or logic that will turn "On" when operation of the associated equipment is verified.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Proof of Operation



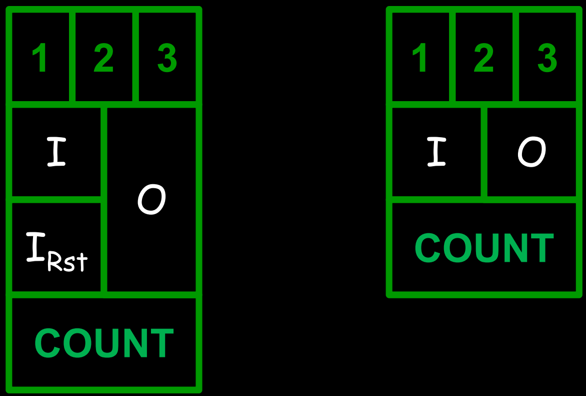
Proof of Operation (continued)

- I_{Cmd} = Connected to the point or logic that is turned "On" to start the associated equipment.
- I_{Rst}= Connected to a point or logic that is momentarily energized to reset the alarm condition latch.
- O_{Alr} = Output that is latched "On" if the proof of operation signal is not received with-in the delay time.
- O_{Nml} = Output that is turned "On" if the proof of operation signal is received with-in the delay time.

Back to Contents

DDC Logic Function Blocks

Counter



Counter - Counts pulses applied to input I.

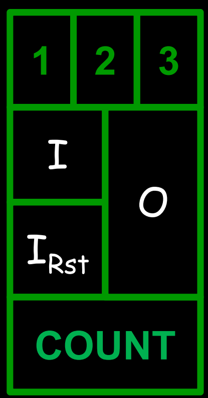
- I = Digital/binary input; each transition of the input from open to closed ("Off" to "On" or "0" to "1") increments the counter by 1. When the counter reaches the largest number it can record, the next "I" transition rolls it over to "0" and it starts again.
- I_{Rst}= Connected to a point or logic that is momentarily energized to reset the counter to "0". If there is no resetting input, the counter accumulates values to its maximum and then rolls over to zero; see above.
- O = Analog output; running total.
- 1,2,3 = Represent the numbers in a counter; no connections are made to them, they are just there to make the symbol look like a counter.

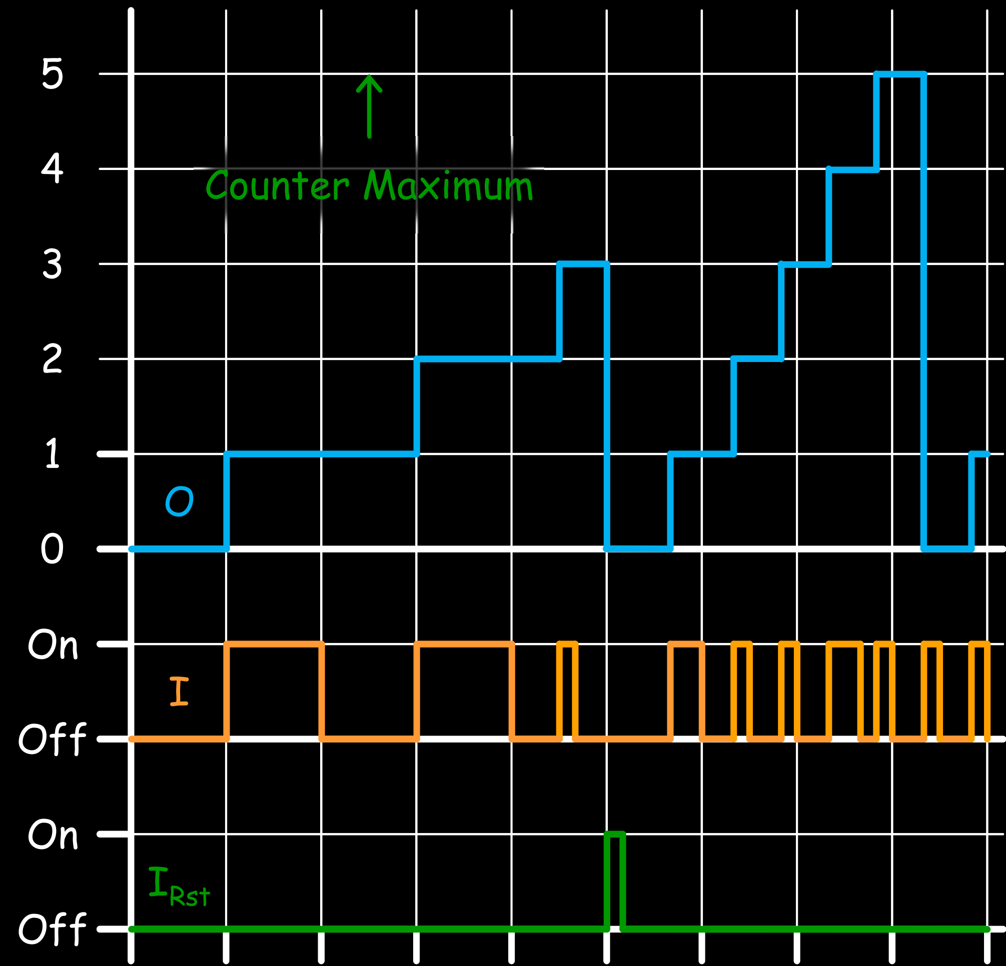
(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Counter



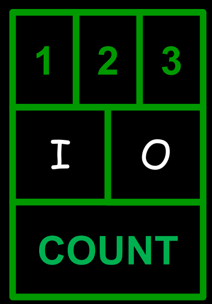


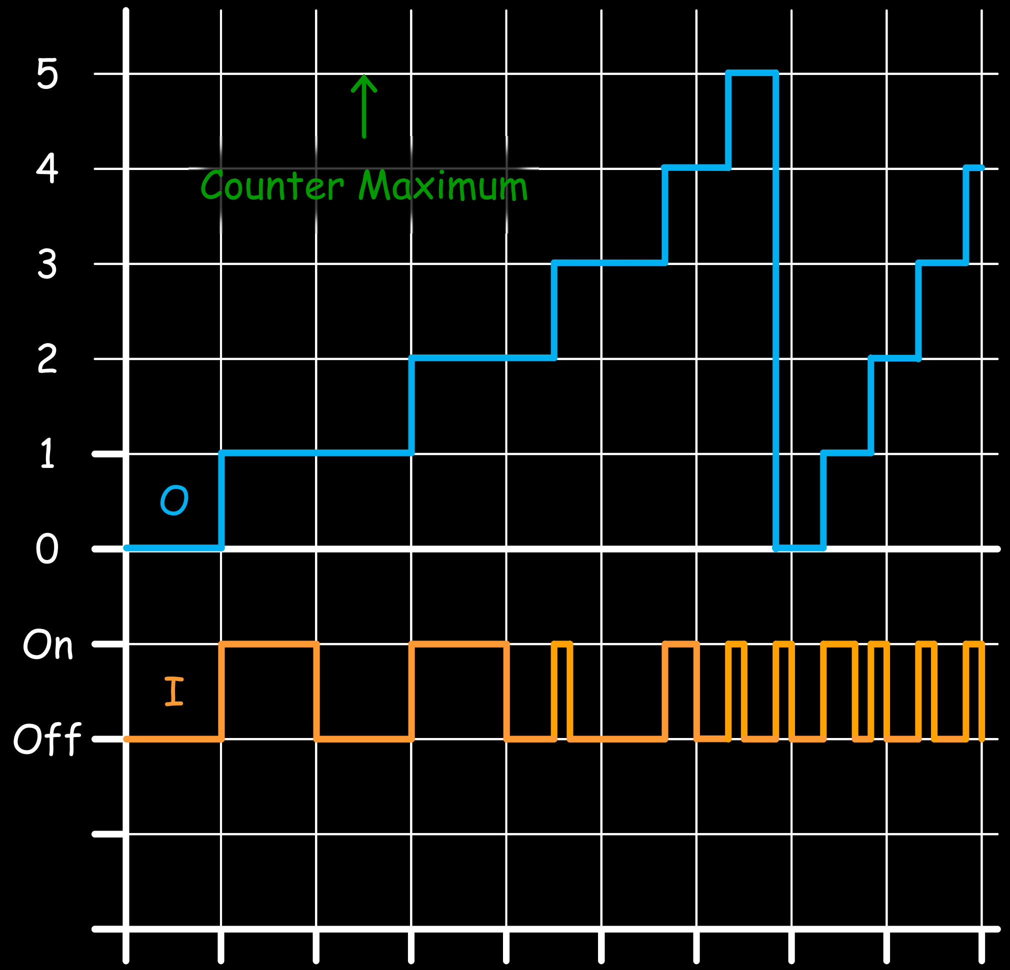
(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Counter

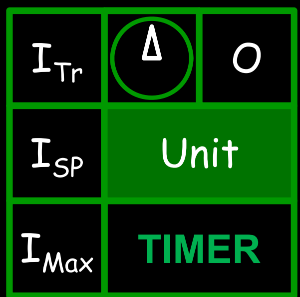


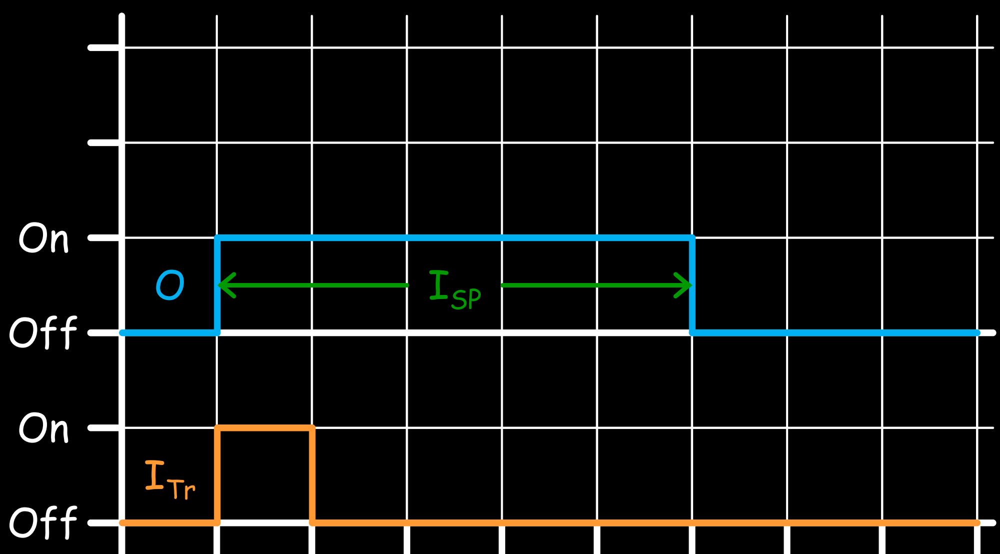


Back to Contents

DDC Logic Function Blocks

Interval Timer





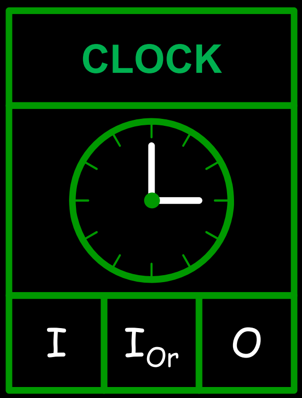
Interval Timer – When I_{Tr} is turned "On", output *O* turns "On" and remains "On" for the time specified by I_{SP}. At the end of the time period output *O* turns back "Off".

- I_{Tr} = Input that triggers turning on Output "*O*". I_{Tr} must turn "Off" and then turn back "On" to trigger a new cycle (digital/binary).
- I_{SP} = Input that specifies how long Output "*O*" will remain on after I_{Tr} turns "On".
- I_{Max} = Large movie theater with very high quality images and surround sound; actually, this limits the maximum value of I_{SP}.
- *O* = Output (digital/binary).
- Unit = Unit of measure associated with the value that is input to I_{Sp} such as hours, minutes, etc.

Back to Contents

DDC Logic Function Blocks

Time Clock



Time Clock – Input I is connected to Output O based on a schedule. The schedule shall:

- Provide for a different schedule for each day of the week
- Provide for a different schedule for a holiday.
- Provide for up to 4 different "On-Off" events per day or holiday
- Provide for all standard U.S. holidays plus 10 additional user defined holidays.

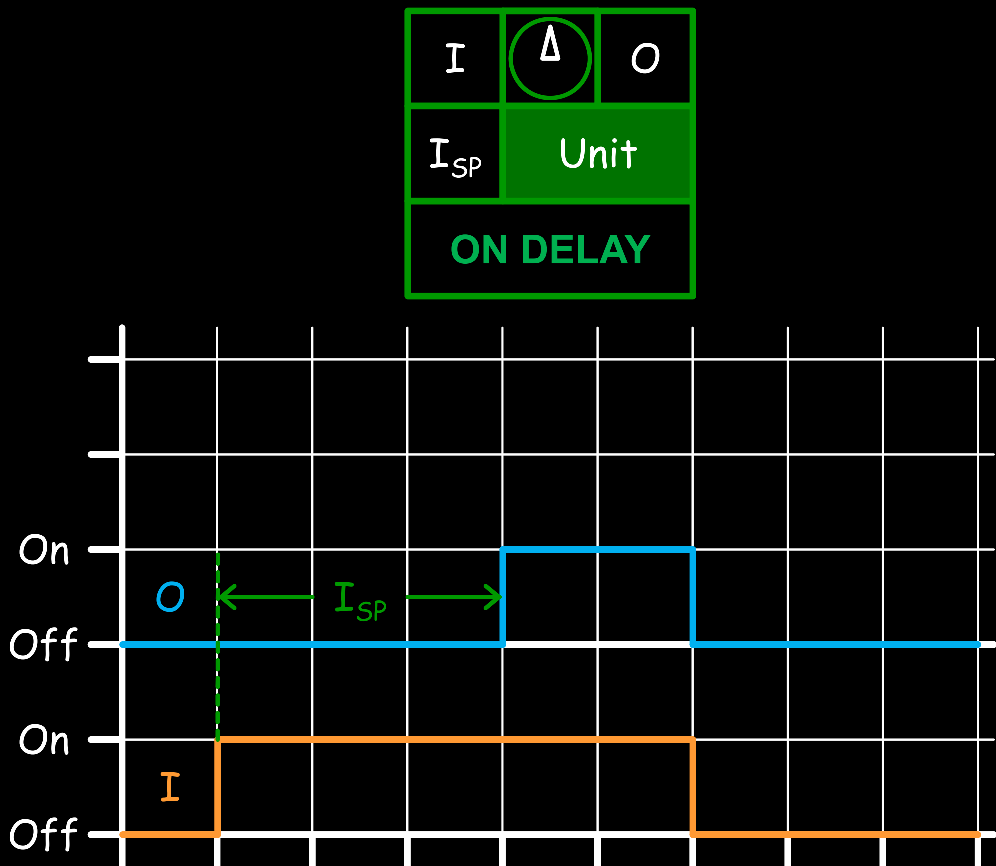
Functions are as follows:

- I = Input from the schedule database as described above.
- O = Output (digital/binary); "On" if the schedule is occupied or if I_{Or} is "On".
- I_{Or} = Overide input; turning this on turns output O on even if the schedule is unoccupied.

[Back to Contents]

DDC Logic Function Blocks

On Delay Timer



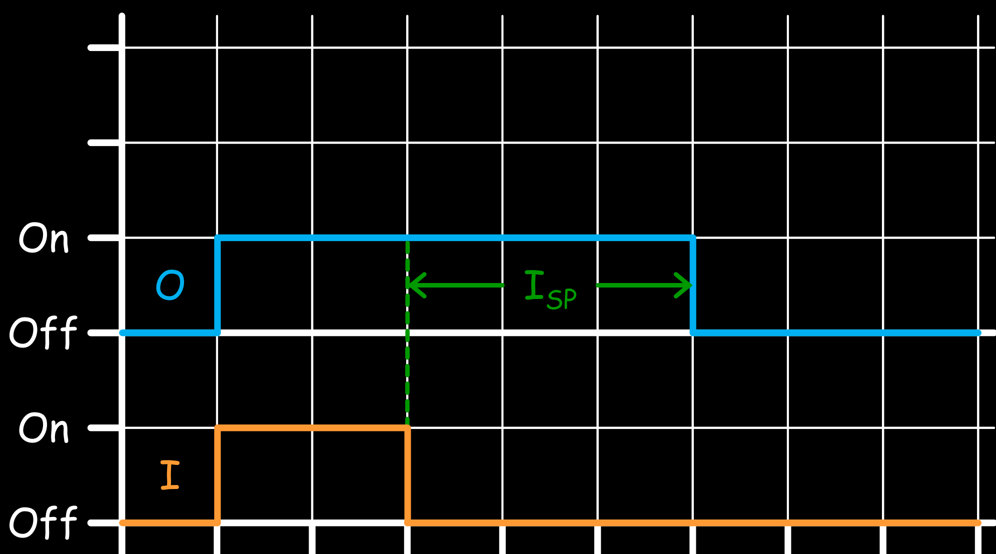
On Delay Timer – When input I turns "On", output O turns "On" after the delay specified by I_{SP}. When input I turns off, output O turns "Off".

- I = Input (digital/binary)
- I_{SP} = Input that specifies how long Output O will remain "Off" after I_{Tr} turns "On".
- O = Output (digital/binary)
- Unit = Unit of measure associated with the value that is input to I_{Sp} such as hours, minutes, etc.

Back to Contents

DDC Logic Function Blocks

Off Delay Timer



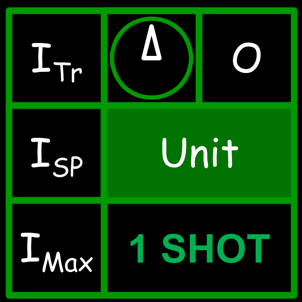
Off Delay Timer – When input I turns "On", output O turns "On". When input I turns off, output O turns "Off" after the delay specified by I_{SP}.

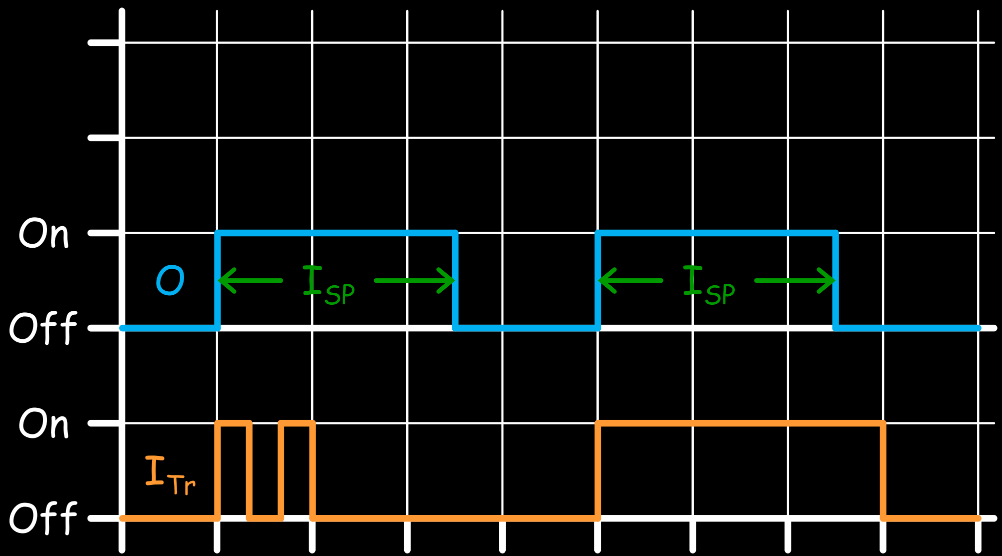
- I = Input (digital/binary)
- I_{SP} = Input that specifies how long Output O will remain "On" after I_{Tr} turns "Off".
- O = Output (digital/binary)
- Unit = Unit of measure associated with the value that is input to I_{Sp} such as hours, minutes, etc.

Back to Contents

DDC Logic Function Blocks

One Shot Timer





One Shot Relay – When I_{Tr} is turned "On", output *O* turns "On" and remains "On" for the time specified by I_{SP}. At the end of the time period output *O* turns back "Off". During the time delay, transitions of I_{Tr} are ignored.

- I_{Tr} = Input that triggers turning on Output "*O*". I_{Tr} must turn "Off" and/or be off after the time delay and then turn back "On" to trigger a new cycle (digital/binary).
- I_{SP} = Input that specifies how long Output "*O*" will remain on after I_{Tr} turns "On".
- I_{Max} = This limits the maximum value of I_{SP}.
- *O* = Output (digital/binary).
- Unit = Unit of measure associated with the value that is input to I_{Sp} such as hours, minutes, etc.

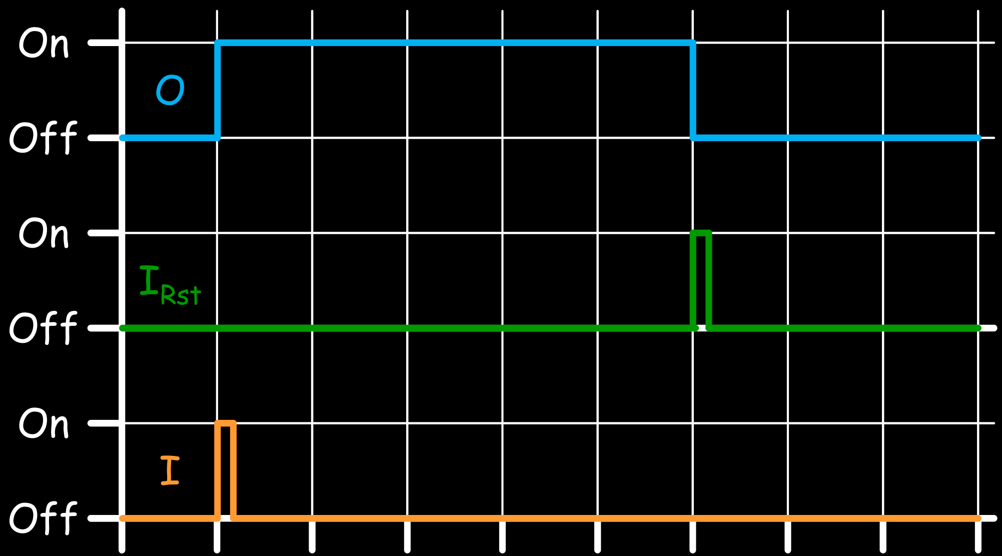
[Back to Contents](#)

DDC Logic Function Blocks

Latch On

Latch On – Output O turns "On" when input I turns "On" momentarily and remains on until input I_{Rst} turns "On" momentarily.

- I = Input (digital/binary, momentary)
- I_{Rst} = Resetting input (digital/binary, momentary)
- O = Output (digital/binary)



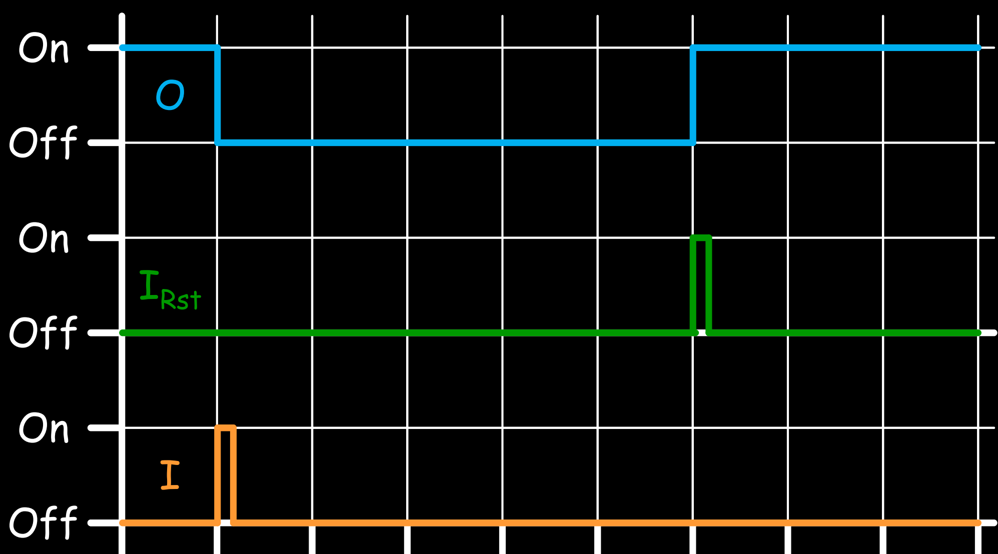
[Back to Contents](#)

DDC Logic Function Blocks

Latch Off

Latch Off – Output O turns "Off" when input I turns "On" momentarily and remains off until input I_{Rst} turns "On" momentarily.

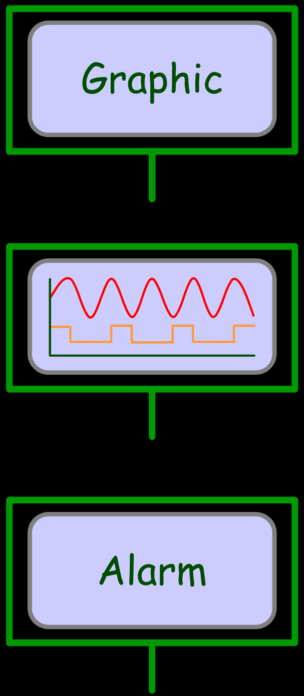
- I = Input (digital/binary, momentary)
- I_{Rst} = Resetting input (digital/binary, momentary)
- O = Output (digital/binary)



[Back to Contents](#)

DDC Logic Function Blocks

Graphic Interface, Trend, Alarm



__Graphic Interface__ – Value appears on a graphic screen or screens. See specification and drawing requirements for graphics for additional details

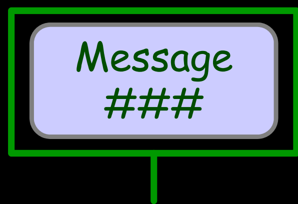
__Trend__ – Establishes a trend for the value including archiving the data. See the point list and specifications for additional details.

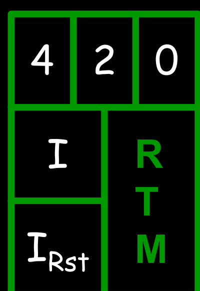
__Alarm__ – Generate an alarm based on information on the point list associated with the point. See the point list and specifications for additional details.

Back to Contents

DDC Logic Function Blocks

Message, Run Time Monitor





Message – Send a message; ### is the message number from the message list associated with the operating sequence logic for the system. See the message list for the specific message and triggering event. See the drawings and specifications for additional information.

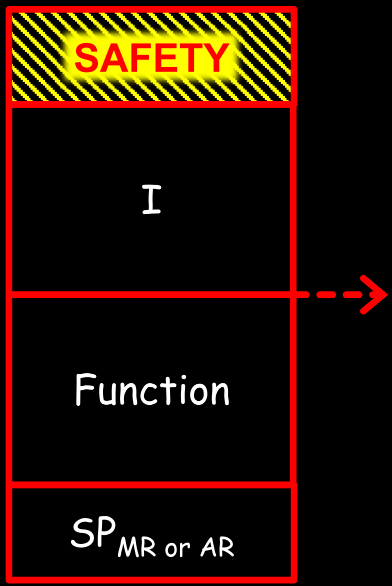
Run Time Monitor – Accumulates runtime based on the status of a digital input and converts it to hours. Accumulates hours for the day, for the month to date and for the year. Archives daily hours.

- I = Digital/binary input associated with the device being monitored. If the input is “On” the RTM accumulates time.
- I_{Rst} = Resetting input; turning the input on momentarily resets the counter to “0”. If there is no resetting input, the counter rolls over to “0” when it reaches its maximum value.

Back to Contents

DDC Logic Function Blocks

Hardwired Safety



Hard Wired Safety – This logic block is an accessory logic block that is attached to an output to indicate that there are hard wired (non-DDC logic) controls associated with the output that are arranged to over-ride the action of the DDC system to protect the equipment, building, or its occupants and processes from damage due to an unusual event or a situation under-which the system should not be operated.

Examples include freezestats, firestats, smoke detectors, fire alarm shut downs, high and low static pressure limits, pressure relief doors, motor overloads, limit switches, etc.

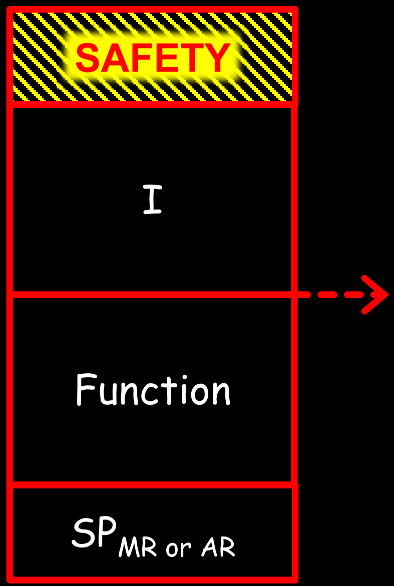
The control functions associated with a safety are accomplished outside of the DDC logic by some physical device that is independent of the DDC logic.

(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Hardwired Safety



Hard Wired Safety (Continued)

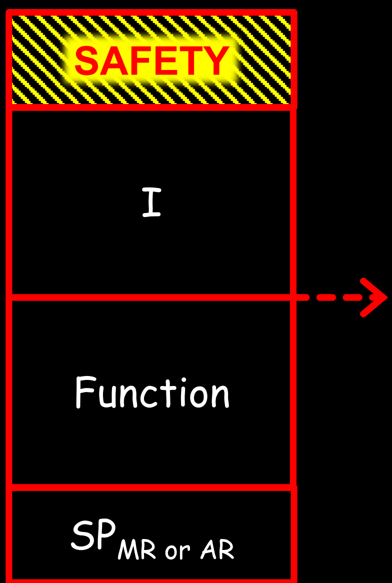
- I = Physical parameter being monitored; for example mixed air temperature or duct static pressure or motor current.
- Function = Function of the device; for example Freezestat or Hi Limit or Pressure Relief
- $SP_{MR\ or\ AR}$ = Set Point; An "MR" subscript indicates a device that must be manually reset. An "AR" subscript indicates a device that automatically resets.

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Hardwired Safety



Hard Wired Safety (Continued) – Most of the time, the device accomplishes it's function via a contact that is wired into the motor control circuit of the equipment that it is associated with and the wiring is arranged so that the device will shut down the machine regardless of the position of any Hand-Off-Auto switch or Invertor-Bypass switch.

A relatively common exception to this are duct pressure relief doors, which are spring loaded panels installed in duct systems that will blow open if there is an air hammer event. Air hammer events can occur in fractions of a second, so pressure based switches interlocked with the fan motor will not provide protection from such an event.

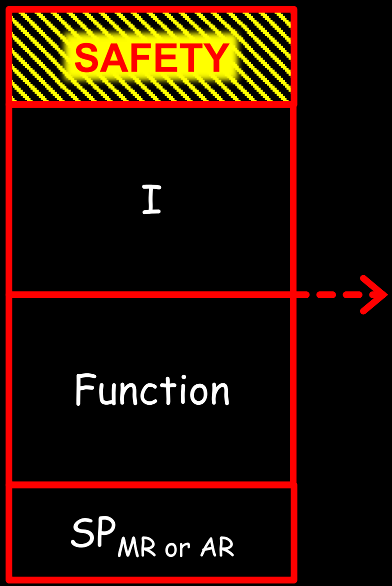
Most safety interlocks require manual reset, meaning if they trip, the mechanism is arranged so that when the triggering condition goes away, the

(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Hardwired Safety



Hard Wired Safety (Continued) – Most safety interlocks require manual reset, meaning if they trip, the mechanism is arranged so that when the triggering condition goes away, the contact does not close until someone comes and takes some sort of action, usually pushing a button on the device that allows the contact to close again.

Limit switches are usually an exception to this. For example a whisker switch that is sensing blade position on a damper to prevent a fan from starting until the damper has opened to a certain position will usually close when the damper reaches the desired position and open back up again if the damper closes back down below the set point.

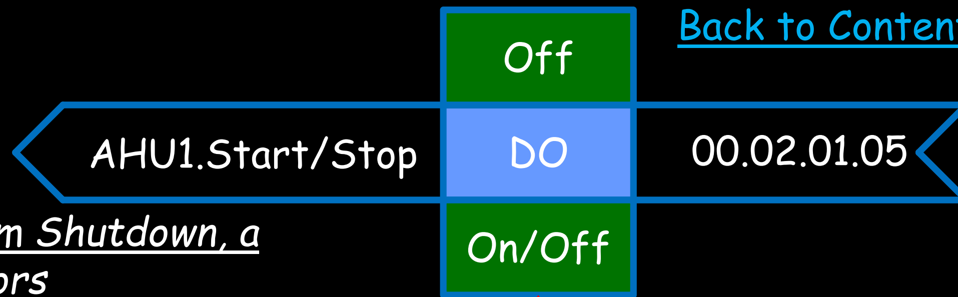
(Continued on the next slide)

Back to Contents

DDC Logic Function Blocks

Hardwired Safety - Example

Fan with a Freezestat, Motor Overloads, a Fire Alarm Shutdown, a High Static Pressure Safety and Pressure Relief Doors



Note 1 – Set by electrical contractor per NEC and project requirements

Note 2 – Relay output from the fire alarm system programmed per project fire alarm system requirements

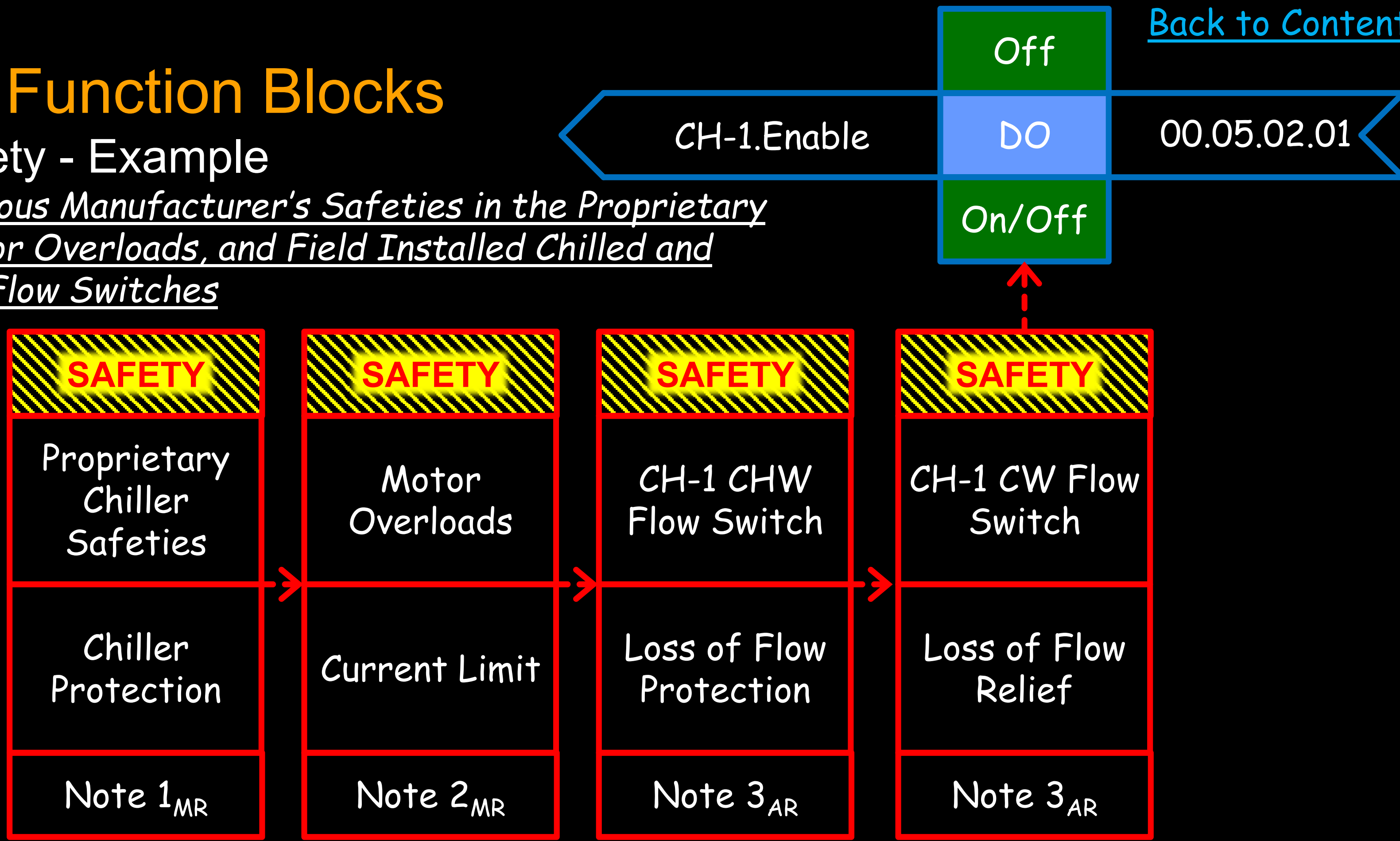
(Continued on the next slide)

[Back to Contents](#)

DDC Logic Function Blocks

Hardwired Safety - Example

Chiller with Numerous Manufacturer’s Safeties in the Proprietary Control Panel, Motor Overloads, and Field Installed Chilled and Condenser Water Flow Switches



Note 1 – Refer to the chiller I&OM for additional information.

Note 2 - Set by electrical contractor per NEC and project requirements.

Note 3 – Coordinate with the chiller start-up technician to adjust for the appropriate set point