

# A LOGICAL GUIDE TO Relay Ladder Logic Symbols

## Introduction

You've trail-blazed thousands of miles with your caravan. As the sun beats down on your back, heat blisters the size of softballs make your skin feel like it's boiling. You don't know if you guys will make it with the few supplies you have left.

To the relief of your entire clan, you spot a small limestone cave off in the distance.

As you and your comrades settle in for the best night of sleep you've had in weeks, a brilliant idea strikes you.

You have to write it down now. You can't risk forgetting something so important.

You reach for one of the exotic roots you've collected along your travels and touch it to the limestone.

In a stunning red dye, you leave a message for the next group traveling through:

*"If you've come this far and still have enough water, keep going on the old path. If you're running out of provisions, take this shortcut to the city..."*

Like ancient travelers following the path of [hieroglyphic messages](#), we use logic symbols to direct the flow of our production lines.

## Relay Logic

Relay logic, ladder logic, and ladder diagrams all refer to the [way we organize](#) an electrical circuit.

Relay logic's ladder diagrams used physical contacts, coils, switches, and lots of other devices for every single function on a machine.

As time went on, changes to logic programming and mechanical failures caused engineers to look for a digital alternative.

Early relay diagram symbols were literal interpretations of physical relay panels. Electricity is shown going through the left side of the ladder in order to energize the input and achieve the output.

Ladder logic looks almost identical to a ladder diagram except the contacts and coils are replaced with computer bits.

But we still need to illustrate what these bits represent, so [we use logic symbols](#). These symbols come straight from relay logic diagrams even if some of the components are digital now.

By using a [programmable logic controller](#) (PLC) we can set up our ladder logic programs and let the computer do the repetitive part.

This gives you more time to work on new PLC designs and programs rather than constantly having to monitor the state of your machines.

### Relay Ladder Logic Symbols

Contacts and coils are represented on the ladder just like they would be modeled in real life. For example, the symbol for contact often indicates some kind of input, like a light switch.

Relay logic even lets us install features like turn-off delays. If we wanted to give everyone 5 minutes to exit a room after turning the light switch off, we can build a delay into the diagram.

The same goes for turn-on delays. We can set a timer within our circuit to delay turning a machine on that may get turned on by accident. We'd still have several minutes to flip the switch off.

On the other end, we have our outputs, which aren't always as visible as lights turning on. Many outputs represent only a single bit in the PLC and act as the input for another rung on the ladder.

So instead of our lights going on right away, we could send the output over to another switch that requires a key. The output of the first switch is now the input for the second.

### Bit Logic Instructions

Building our logic symbols diagram starts with [bit logic instructions](#). Bit logic instructions run on the same principles as good old machine language, using nothing more than 0s and 1s to send signals.

The result of logical operations (RLO) is where we gain the freedom to set the outcomes of each rung of our logic symbols diagram.

When all our bit logic instructions and their corresponding RLOs are put together, we have the horizontal rungs of our logic symbols diagram held together by vertical power lines.

And just like a good book, we always read these diagrams from top to bottom and left to right.

### Scan Time

Scan time can mean the difference between meeting a deadline and having disappointed clients waiting on their orders.

PLCs are computers and need time to process what they read just like we do. Even though they can scan programs faster than we can blink, there is still room for improvement.

This is where function blocks become crucial to a profitable assembly line.

## Function Blocks

Function blocks look like literal blocks that squeezed their way into your logic symbols diagram like a pair of jeans they outgrew twenty pounds ago.

Some refer to them as control blocks, but in this article, we're going to use only the term function blocks to avoid confusion.

By breaking down our more complex tasks into function blocks, we can speed up the time it takes for our PLC to run through our circuit.

### Why Use Function Blocks?

Not only do function blocks improve scan time, they often [replace entire physical components](#) of your system.

They can help you avoid the need for hard-wired mechanical timers, saving you on both upfront costs and maintenance.

They can replace several lines of rungs in your ladder, making your diagram easier to implement and read. It provides a much clearer illustration of multiple inputs and their relationship to one another.

For example, if we were using a traditional ladder diagram we'd need two connect logic symbols in order to allow for more than one input.

### Introducing "AND" Logic

Function blocks allow us to condense rungs of the ladder by using AND logic. Using function blocks is like putting all your nuts and bolts into their own containers.

It's not that regular ladder logic is hard to follow, but complex systems can look messy and cluttered.

Even experienced techs shouldn't have to deal with messy PLC programs.

Function blocks can make it easier for a technician to diagnose problems with the PLC at a glance.

AND logic is perfect for designing security features into your production line.

Why?

Let's say you have an [industrial pressing machine](#) on your line that can apply enough pressure to compact a skyscraper into the size of a walnut.

To avoid any accidents where the press is turned on by accident, a function block using AND logic would force your technicians to press two or more buttons for the machine to start.

So one input is totally useless without the other when it comes to energizing the output.

### Introducing "OR" Logic

Since function blocks condense our lines of ladder logic, they always have a ladder logic counterpart.

AND logic is the same as a series circuit and OR logic is equivalent to a parallel circuit. So we can use OR logic within our function blocks to require one input or the other.

OR logic is great for machines with multiple control panels. A large machine may have an on/off switch at both ends for the sake of convenience. With OR logic using either panel causes the same effect.

## Most Common Logic Systems

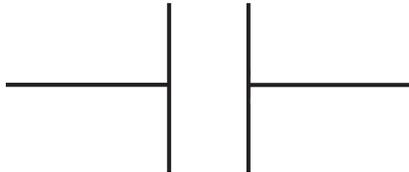
Ladder logic includes an entire language of symbols, but for this article, we are only interested in the ones [you'll see in diagrams](#).

### Contact Symbols

The logic symbol for a normally open contact looks like a break in one of the ladder's rungs. These symbols come in two types, the first being the "Normally Open" type, and the "Normally Closed" type.

The following explanation should clarify their differences and similarities:

#### Normally Open (NO) Contact



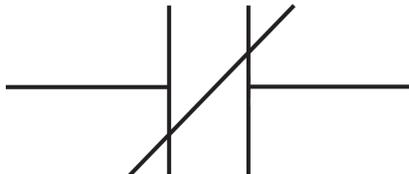
Using our light switch example from earlier we know this contact would represent the light switch. Unless someone flips the light switch on the circuit stays off.

Some of the most common uses for NO contact symbols are for:

- Power buttons
- Light switches
- Internal programming

If you're still unclear on the function of this logic symbol, an explanation of its polar opposite should help...

#### Normally Closed (NC) Contact



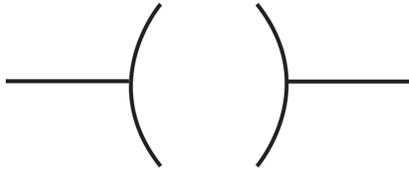
The polar opposite of a NO contact looks just like a NO contact but with a slash going through it. A circuit is active until input tells the NC contact to shut things down.

The most common uses for NC contacts are:

- [Stop buttons](#)
- Fail-safe features
- Heat monitoring

Notice the contrast between the uses of the Open and Closed versions of the contact logic symbols. This contrast helps many new programmers understand both types better.

## Output Symbol

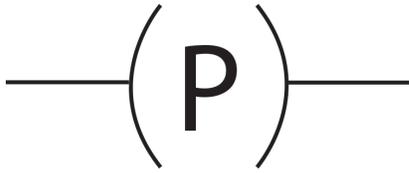


Common uses of the output symbol are:

- Warning signs
- Indicator lights
- [Motor contactors](#)

The output symbol looks like it swallowed the NO contact symbol. When the corresponding input becomes energized the output is on.

## One Shot Positive Edge Connection

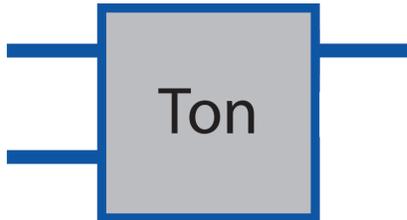


They are best used in:

- Counters
- Math commands
- Data transfer

These look like the output signal with a capital P in the middle. If a condition changes from false to true over the time it takes the PLC to go through one scan, the output is turned on.

## Timer Delay-On (TON)

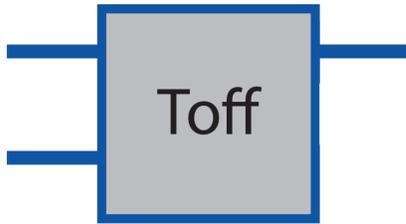


If we energize the input of our PLC program, the timer to turn the application on begins. This works great in:

- Delaying siren sounds
- Sequence start delays

Think of this in terms of starting your car and letting it idle on a cold morning, before venturing out onto the road. You're giving your engine time to "warm up." Likewise, "TON" gives a machine time to perform startup functions.

### Timer Delay-Off (Toff)



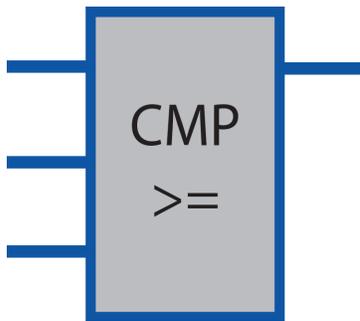
Just like a NC contact but with a timer. If the input is energized the timer begins and the application will turn off after. The best uses of "Toff" are:

- Automatic displays
- Machinery that may accidentally get turned off

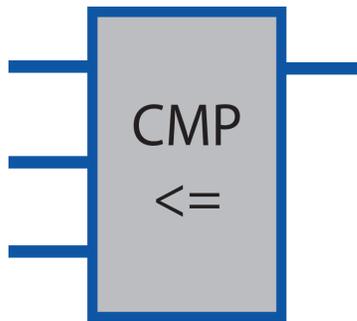
Think of this as the opposite of the "warm-up" time analogy above. Instead of turning off immediately, "Toff" puts a delay between the shut-off and the machine actually shutting down.

### Comparison Logic Symbols

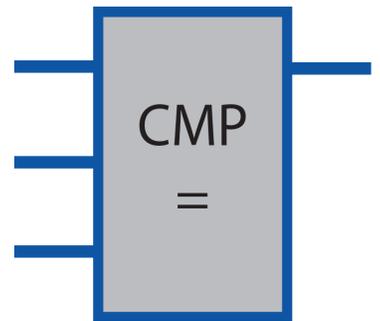
#### Greater Than or Equal To



#### Less Than or Equal To



#### Equal To

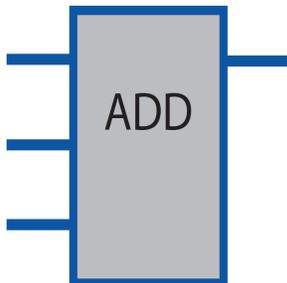


Using class math symbols, comparison logic symbols look like equal to, greater than, and less than symbols.

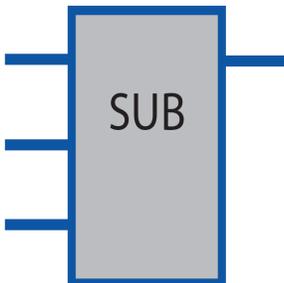
Batching systems use comparisons to make sure the contents of every single package are filled to the proper weight before moving on.

### Math Instructions

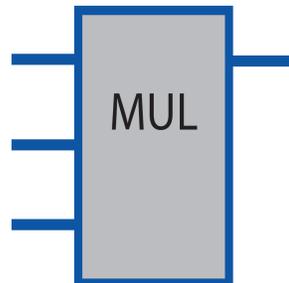
#### Addition



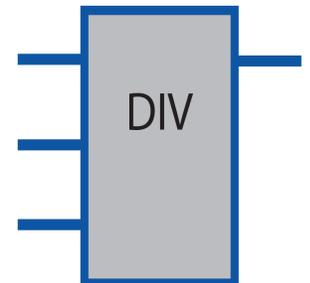
#### Subtraction



#### Multiplication



#### Division



Logic symbols also allow our ladder logic program to execute most mathematical operations.

We need these operator symbols for calculating things like motor speed.

## Variations in Logic Symbols

As you may know from frustrating experiences, different brands of PLCs use slight variations of the standard logic symbols.

But the IEC works hard to put out standards to make things easier for everyone to understand.

So, are makers of PLCs throwing caution to the wind when using variations of standardized logic symbols?

Not exactly. See, the early graphical interfaces used by manufacturers wasn't much more than a dressed-up command line.

Any characters had to be represented by ASCII characters, basically what's on your keyboard.

The IEC knows slight variations in line thickness and all sorts of other minor details can make logic symbols look a little different. So while they put out a standard logic symbols table, they don't describe any specific characters that must be used.

And it doesn't really matter because they all function the same way. Variations in symbols are superficial and while they might make you do a double take; they look very similar to the standard set of logic symbols.

You don't need to worry that your new PLC software has logic symbols you've never seen or heard of.

## c3 Makes Logic Symbols Funner Than Memes

It's 6:30 at night after a long day of work. You're packing up for the day when you hear it.

Your production line stops without any warning and the deafening silence pours around you like cement, stopping any plans you had to leave work on time.

You can almost hear the collective groan from everyone that knows this means over time. You're probably thinking, "this is what I signed up for with this job."

But it doesn't have to be that way.

[Most of the time](#), faulty wiring and other defects are to blame for PLC failures. With outsourcing being so commonplace in the electronics industry, it's no wonder so many companies put an emphasis on quantity over quality.

And hey, can you blame them? They're just trying to meet their bottom line, right?

Wrong. Just like the output of a function can be the input for another operation, what you get from your electronics manufacturer directly effects your business.

When the Taylor family decided to buy their own equipment rather than outsource, they made a commitment to quality the way only a vertically integrated manufacturer can.

They have total control of the product they deliver to you from start to finish. No dealing with sketchy suppliers halfway across the world out to make a quick buck.

## Take Your Logic Symbols to the Next Level

Partnering with c3 gives you over 40 years of industry expertise, all with the fast and friendly service of a family business.

No sitting through hours of automated messages only to be directed to another department. Work with experts. Work with family.

Our equipment turns boring logic symbols into the assembly line of your dreams.

[Check out our inventory now!](#)

### **Disclaimer:**

The content provided in this white paper is intended solely for general information purposes and is provided with the understanding that the authors and publishers are not herein engaged in rendering engineering or other professional advice or services. The practice of engineering is driven by site-specific circumstances unique to each project. Consequently, any use of this information should be done only in consultation with a qualified and licensed professional who can take into account all relevant factors and desired outcomes. The information in this white paper was posted with reasonable care and attention. However, it is possible that some information in these white papers is incomplete, incorrect, or inapplicable to particular circumstances or conditions. We do not accept liability for direct or indirect losses resulting from using, relying or acting upon information in this white paper.