

# Resolving Component Polarity Problems

Jack C. Olson, CID+

One manufacturing issue discussed on a regular basis is component polarity. The intent of this article is to describe the process for translating component placement information from the computer-aided design (CAD) software to the component pick-and-place machine and to suggest ways in which the process might be improved.

I have heard complaints about backwards parts on assemblies and drawings that are unclear, and the response from engineering has been to “use the data.” This solution is not thoughtless—when software has been debugged properly, computers don’t make mistakes, and humans often do. In an ideal situation, a good set of CAD-generated data files will provide enough information to mount components on a bare board correctly. Theoretically, a board should be able to be assembled using computer-generated data alone, without silkscreen, without polarity marks and without the need for visual inspection to a drawing.

If the data provided by the CAD system is not reliable, though, the assembler is forced to add manual processes using various types of drawings and visual inspection to ensure the assembly is correct. Even automated optical inspection (AOI) equipment is ineffectual if it has not been loaded with good data. Attempts to resolve problems caused by CAD-generated data files are time consuming and error prone; therefore, examining ways to improve accuracy and usefulness is in our best interest.

## First, a Little Background

To better understand where the data comes from, a brief explanation of the design process is required. One of the ways a CAD system aids the circuit board designer is a process called *packaging*. When packaging a design, the software looks at the schematic diagrams created by an engineer and maps the schematic symbols into actual components. After the packaging is complete, each component is given a unique identifier called a *reference designator*. At this point, a bill of materials (BOM) can be generated since the system now knows how many of each type of component is required to package the design.

The components are also shown on the designer’s computer screen in their default orientation. In other words, the components are loaded into the design database in the same orientation in which they were created by the CAD librarian. If the designer were to create data files at this point, all of the components would be described as having a rotation of zero degrees, since none of them have been moved or rotated yet. The bare board designer’s responsibility is to rearrange the components by moving and rotating them into logical groups and connecting

them together using copper traces in such a way that the design can be manufactured efficiently and the finished circuit board performs reliably under all conditions.

## Understanding the Pick-and-Place File

The one factor that we are addressing here is the fact that, during the design process, components are rotated in the database to make a better layout. The system remembers these rotations and reports them in a file called a *pick-and-place file* (Table 1). The reason it is called a *pick-and-place file* is because a special piece of equipment on the manufacturing floor uses this data, and, for each component described, it will “pick” the correct part from a reel and “place” it on the board in the proper location with the correct rotation.

**TABLE 1:** A sample pick-and-place file

C1	148-6881	c7343	1.0750	0.7250	1	270
C2	534-0034	c7343	1.4000	0.3500	1	90
J1	123-4567	conn8	1.2000	1.0000	2	180
Q1	879-6151	sot23	0.7500	0.6500	1	0
R1	141-4901	r0805	0.4750	0.3750	1	90
R2	117-2850	r0805	0.6750	0.3750	1	90
R3	233-3440	r1206	0.8500	0.3750	1	180

The format of this file is easy to understand. Each line of the file represents one component, so this design contains two capacitors, a connector, a transistor and three resistors (Table 1). For each component we are given the reference designator, the part number, the location (x-y coordinate), which side of the board the component is located on and the rotation. For example, notice that the connector J1 will be placed on the back of the board (side “2”) at location 1.2000, 1.0000 and at a rotation of 180°. The rest of the components are placed on the top of the board (side “1”).

Understand that this one file contains every part in the design, but manufacturing engineers will be evaluating the build and may decide to split the data into multiple parts. The topside components will be separated from the backside because they are run in different passes; some groups may be run on different pick-and-place machines, some components may be installed manually, etc. The point is that they will decide the best way to *use* the data, but the original file contains *all* the data.

Now turn your attention to just the first line of the file, which gives us information about the capacitor labeled C1. This example shows how a potential error can be introduced into the

manufacturing system. This component is a polarized capacitor, which means that it must be installed correctly. If it is mounted backwards, the circuit will not work properly. The data reports 270°, and in the absence of any other documentation such as an assembly drawing, that rotation has to be taken at face value. But what does that mean? To answer that question, let us go back to the beginning and see where the value of “270” came from.

## Begin at the Beginning

Before a new component can be used in a schematic, it has to be created and added to the CAD library. After it is in the library, it can be called by all subsequent designs that require it. To add a new component to the CAD library, a schematic symbol will be mapped to a generic footprint of the appropriate size.

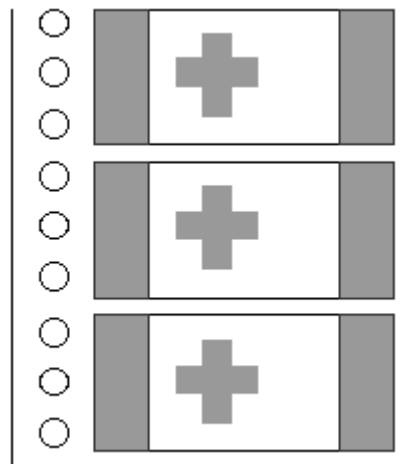
In the example of the polarized capacitor mentioned previously, let’s assume the CAD librarian created a footprint that looked like this picture.

From now on, whenever an engineer selects a part that can use this size footprint, this footprint will be automatically loaded into the design database at this default orientation, and it will have a rotation of zero degrees.

During board layout, the designer might have to rotate the part as shown in this picture, which would result in the final rotation being listed in the pick-and-place file as being 270°, as we saw in the sample file above.

Now let’s look at the situation from the manufacturer’s point of view. He is buying these parts on reels, and the orientation shown in this diagram is generally thought of as being zero degrees.

So the end result of this example is that the designer really wants the part to be placed in the same orientation as the part is loaded on the reel, but the data is showing a rotation of 270°! This discrepancy cannot be resolved by merely saying, “Build it to the data.”



Now take into account that the design may have hundreds of these same part types in all four rotations on front and back. And then realize that, although this component is always 270° off, other polarized component types may only be 90° off, some others 180° and some may accidentally match. To the casual observer looking through the data file, the rotations would appear completely random! At this point, the rotations portion of the data file is virtually unusable.

The immediate question that comes to mind is: “Why not build the library footprints with the same orientation as they are loaded on the reels?” If we were to start building a *new* CAD library, this idea would be an easy way to resolve most of the polarity problems. Unfortunately, some companies are using well-established libraries that have been used on hundreds of designs. If the footprints were to be modified with different rotations, new errors could be introduced the next time one of these designs is revised. We should consider the proper orientation for any *new* component types, even though mixing good rotations with bad still leaves about the same level of difficulty.

The situation is not hopeless. One consistent fact that we can use to our advantage is the knowledge that the rotations for each component type are always the same amount off throughout the entire design. Even though hundreds of different part numbers may use the same footprint, all of the pick-and-place data entries that list that footprint will always have consistent rotations.

For example, look once again at the sample pick-and-place file that we have been using, and notice that, even though the second component C2 has a different part number, it is still using the same c7343 footprint. If the first c7343 footprint is 270° off, then the second will also be 270° off, as well as every other component in the design that uses a c7343. Adding a post-processing step that parsed through the entire data file and added 270° degrees to every c7343 component would be fairly simple.

Performing two simple tasks may make the CAD data files more useful to manufacturing:

1. Create a table of all the polarized component types that are consistently problematic. This file would be an ASCII file that could be as simple as listing the footprint name and the rotation adjustment that is desired, for example:

C7343	270
BGA272	90
SOT23	180

2. Write a short program to:

- read in each line from the original CAD-generated pick-and-place file
- compare the footprint name to the ones listed in the look-up table
- if found, apply a rotation adjustment to it
- write it out to a new file.

This program could be run on the pick-and-place file for every design released, and, although it would not solve every problem, the program would provide a reliable baseline from which manufacturing could start.

Note that, in special cases where a particular part number is delivered on reels in a different orientation than expected, the same process could be used to modify those rotations by searching for the *part number* instead of the footprint name.

## Documenting Polarity

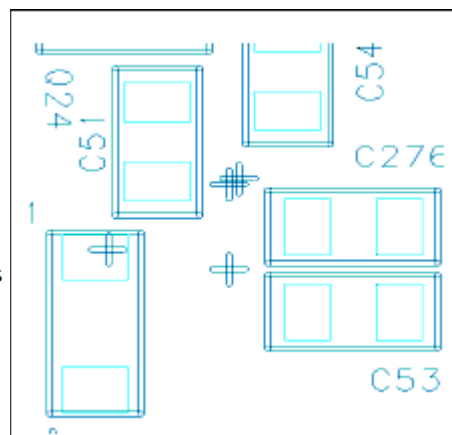
So far, we have only covered issues related to the data, but, since no process is perfect, verifying correctness is mandatory, which is the value of good documentation.

### Silkscreen

As has been noted, the silkscreen on the bare board has limited value. Designs are getting denser, and we are packing smaller parts into smaller enclosures; sometimes, there is not enough room to arrange the reference designators in any logical manner. Many times, they are simply removed if they look confusing and might cause more harm than good. If this trend continues, the end result may eventually be no silkscreen at all or maybe just silkscreen for testing and interface purposes. Retaining the polarity marks even if the component outlines are turned off may be beneficial, if the CAD library is structured correctly.

Pictured right is a fairly typical example of a recent design. Notice that four polarity marks are in close proximity. After studying the arrangement, determining the proper way to place the components on the board is possible, but not ideal. There is too much room for misinterpretation.

The bare board vendor will remove one of these marks because it will interfere with soldering, and the two overlapping marks will look like a blob of white ink on the end product. The designer could make the design clearer for everyone by moving the polarity marks.



Another point that might be helpful to manufacturers would be to have the *shape* of the polarity mark be consistent. For example, I have seen filled circles, hollow circles, diagonal lines across corners, double lines and the number "1" on integrated circuits (ICs). Plus signs, curved lines, bold lines and the letter "C" or "K" have been on various types of diodes, many of which cannot be seen after the component is installed. Using marks consistently will help everyone involved.

### Polarity Identification in Copper

Another idea is to build the footprint in such a way that the copper pad for pin one is a different shape than the rest. Polarity could be identified in the absence of silkscreen. This solution seems simple as long as the difference is visible after the component is installed and as long as the solderable surface of the pad (solder mask-defined) stays the same.

### Assembly Drawing

Often, the assembly drawing is of limited value as far as polarity checking goes. Sometimes, fast-turn prototypes are being built before final documentation is complete, and machines are being programmed from data files before the drawings have even been started. Even with a released drawing the information can be misinterpreted, especially if the drawing is little more than a picture of the silkscreen. Consider turning off the silkscreen layer and showing the reference designator centered in a simple outline. This practice will free up a lot of area and make the polarity marks more apparent.

## Summary

Cleaning up an existing library of footprints may require a significant investment in time by the CAD librarian. Ensuring that polarized components are installed correctly requires an extra effort by the board designer during layout and when creating documentation. The benefits seen in manufacturing may not be readily apparent, but the cost in machine down time, testing, repair, schedule slippage and frustration often far outweighs the cost of the time spent upfront. Hopefully, these practical techniques will help your company reduce the need for tedious and error-prone manual processes.

Jack Olson can be reached at [pcbjack@gmail.com](mailto:pcbjack@gmail.com)