APPLICATION NOTE

Using newWaveX—Applying Shorts

How to apply a shorting link or zero ohm load in IEEE1641 using newWaveX

Ashley Hulme

Application Note 1003/ Issue 1, January 2010

Introduction

This note explains how to create and use a short or other specified impedance within newWaveX. See reference 1 for a general discussion on using Loads in IEEE 1641.

A short may be implemented in two different ways; the first being the application of a constant source impedance of zero ohms, and the second being to addition of a Load on the output of two pins.

Constant source of type impedance

In this example, a TSF is created using a Constant BSC is used as a source of impedance. This may be set to 0 Ohm and applied (as with any other source) to a pair of UUT input pins. Figure 1 shows a pictorial representation of the TSF as produced by newWaveX. Figure 2 shows the information entered for each of the BSCs (upper and lower panels on the left) and the overall TSF properties (on the right).

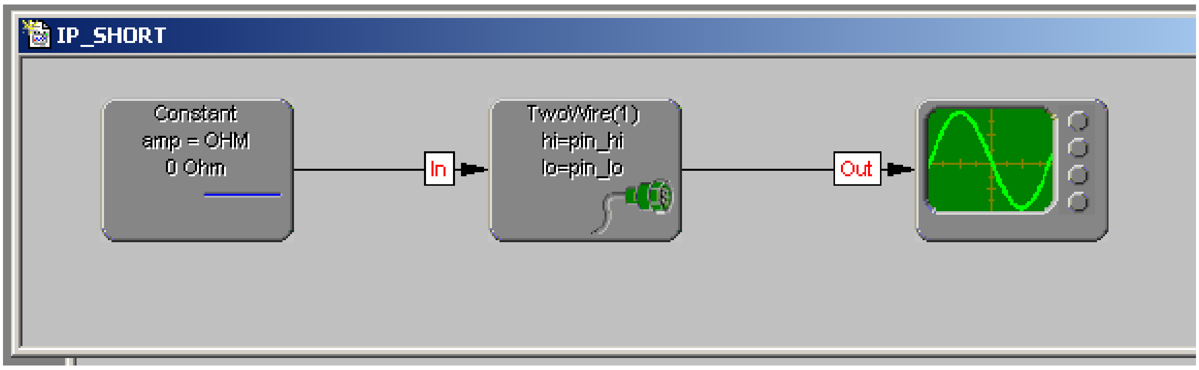

Figure 1—TSF for an input Short

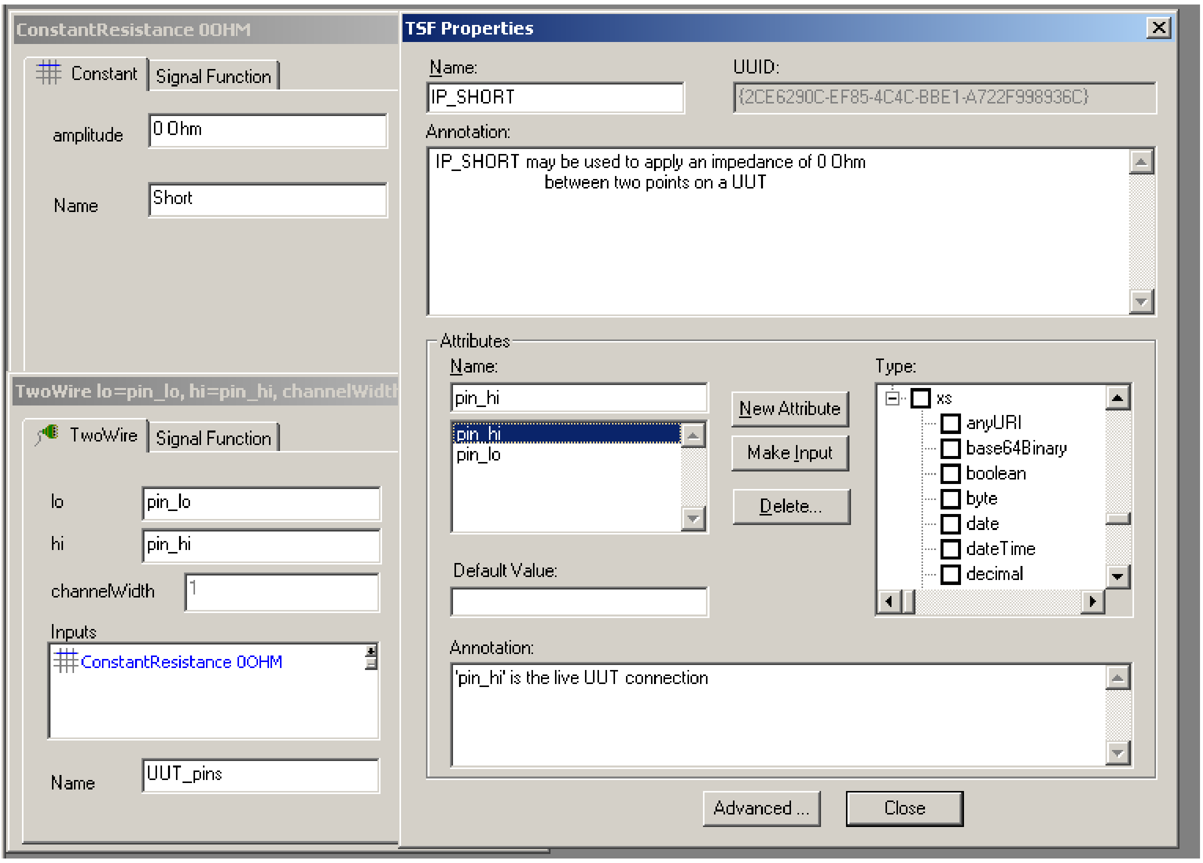

Figure 2—TSF data input panels

The TSF in XML format

The XML for the TSF would be as follows:

```xml
<?xml version="1.0" ?>
<!-- generated with newWaveX v3.1.8 (http://www.newWaveX.com) -->
<tsf:TSF name="IP_SHORT"
  uuid="{2CE6290C-EF85-4C4C-BBE1-A722F998936C}"
  xmlns:tsf="STDTSF" xmlns:std="STDBSC" xmlns="STDBSC">
  <tsf:interface>
    <xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
      elementFormDefault="qualified">
      <xs:element name="IP_SHORT">
        <xs:annotation>
          <xs:documentation>
            IP_SHORT may be used to apply an impedance of 0 Ohm
            between two points on a UUT
          </xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="SignalFunctionType">
              <xs:attribute name="pin_hi" type="string">
                <xs:annotation>
                  <xs:documentation>
                    'pin_hi' is the live UUT connection
                  </xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="pin_lo" type="string">
                <xs:annotation>
                  <xs:documentation>
                    'pin_lo' is the return UUT connection
                  </xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </tsf:interface>
  <tsf:model>
    <Signal Out="UUT_pins">
      <Constant type="Resistance" name="Short" amplitude="0 Ohm" />
      <TwoWire type="Resistance" name="UUT_pins" lo="pin_lo"
        hi="pin_hi" channelWidth="1" In="Short" />
    </Signal>
  </tsf:model>
</tsf:TSF>
```

The model does, of course, simulate correctly within newWaveX as it is a source signal. Although the simulation just shows a horizontal line with an amplitude of zero.

To use the TSF, it is selected as a project TSF icon and the UUT pin names added. This is illustrated in figure 3.

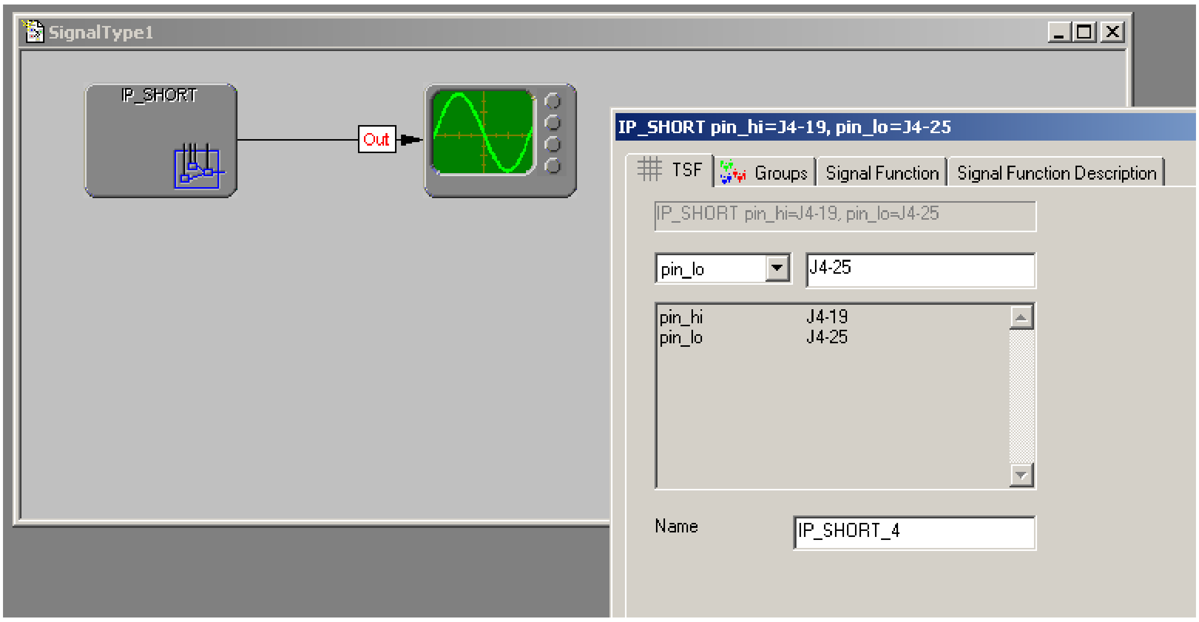

Figure 3—Using the IP_SHORT TSF

Shown below is the XML code showing an instance of the IP_SHORT TSF (named IP_Short_4) connected to two UUT pins (J4-19 and J4-25).

```xml
<?xml version="1.0" ?>
<!-- generated with newWaveX v3.1.8 (http://www.newWaveX.com)  -->
<Signal Out="IP_Short_4" xmlns="STDBSC" xmlns:this="MyTSFLib"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="MyTSFLib MyTSFLib.xsd">
  <this:IP_SHORT name="IP_Short_4" pin_hi="J4-19" pin_lo="J4-25" />
</Signal>
```

It would also be acceptable in many cases, depending upon the application, to apply zero volts across two UUT pins. Although the end result may be the same, it may imply that a voltage source capable of being programmed to 0 V is required, whereas a resistance of 0 Ohm may be considered to imply a wire link. Each requirement should be considered on its own merits and the appropriate solution selected. However, it is important to remember that the 1641 standard does not specify the solution, only the signal required.

Use of the Load BSC on the output of a UUT.

The next model shows how the Load BSC may be used. It assumes an output from the UUT to which a load is attached. This is acceptable in many cases as a short across two pins may actually be a current sink, depending on the circuitry within the UUT. This is illustrated in figure 4.

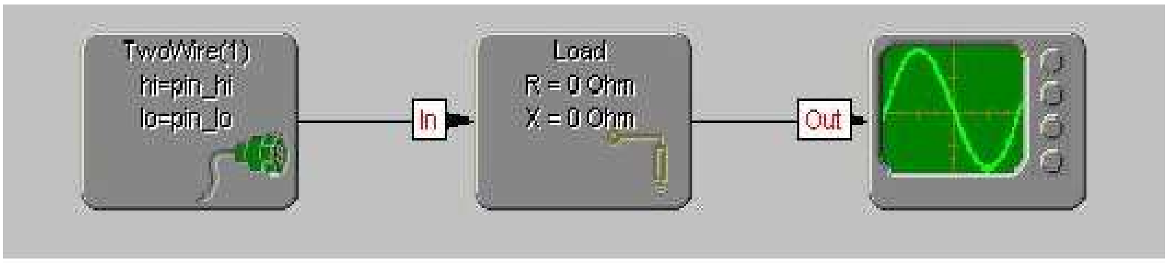

Figure 4—TSF with Load BSC

This shows the TwoWire connection (which is used to connect to the UUT) and the following Load BSC. The Load is given an impedance of 0 Ohm. The Load BSC does not require an output if no measurement is being taken or the signal is not required to be passed to another element, both of which would be pointless after a 0 Ohm load.

The TSF in XML format

The XML for such a TSF would be as follows:

```
<?xml version="1.0" ?>
<!-- generated with newWaveX v3.1.8 (http://www.newWaveX.com) -->
<tsf:TSF name="OP_SHORT"
    uuid="{F919E376-CFD5-43CC-9C1F-C3DA497CAB84}"
    xmlns:tsf="STDTSF" xmlns:std="STDBSC" xmlns="STDBSC">
  <tsf:interface>
    <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">
      <xs:element name="OP_SHORT">
        <xs:annotation>
          <xs:documentation>
            OP_SHORT may be used to apply an impedance of 0 Ohm
            on the output of two pins on a UUT
          </xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:complexContent>
            <xs:extension base="SignalFunctionType">
              <xs:attribute name="pin_hi" type="string">
                <xs:annotation>
                  <xs:documentation>
                    'pin_lo' is the live UUT connection
                  </xs:documentation>
                </xs:annotation>
              </xs:attribute>
              <xs:attribute name="pin_lo" type="string">
                <xs:annotation>
                  <xs:documentation>
                    'pin_lo' is the return UUT connection
                  </xs:documentation>
                </xs:annotation>
              </xs:attribute>
            </xs:extension>
          </xs:complexContent>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </tsf:interface>
  <tsf:model>
    <Signal Out=" Load_Short ">
      <TwoWire name="UUT_Op_pins" lo="pin_lo" hi="pin_hi"
        channelWidth="1" />
      <Load name="Load_Short" In="UUT_Op_pins" />
    </Signal>
  </tsf:model>
</tsf:TSF>
```

This model does not simulate as it does not have an input signal attached to it.

In use, the TSF is selected as a project TSF icon and the UUT pin names added in a similar way to the IP_SHORT TSF. This is illustrated in figure 5.

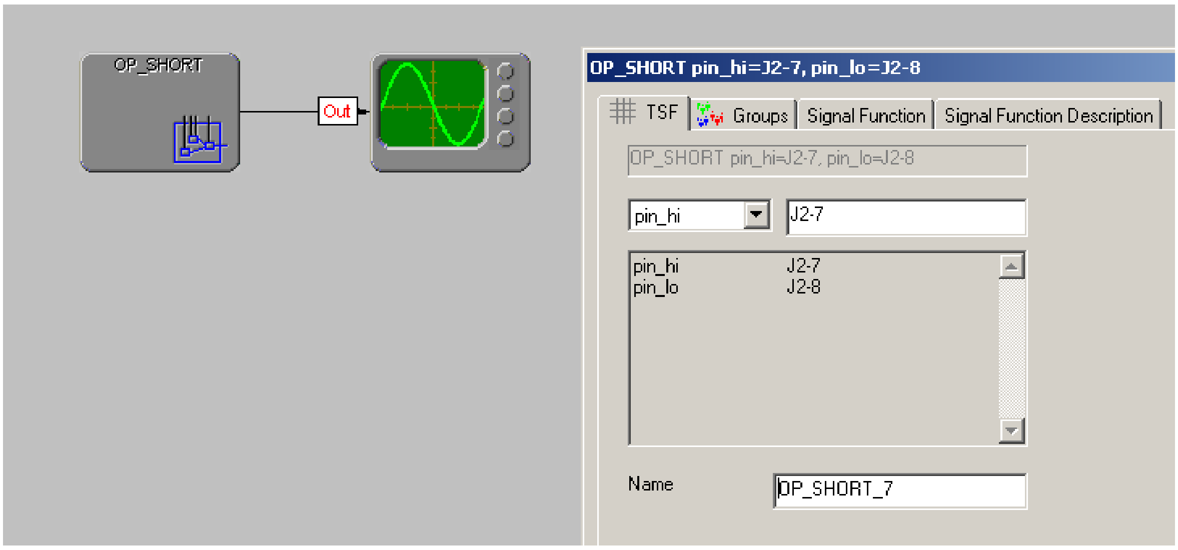

Figure 5—Using the OP_SHORT TSF

Shown below is the XML code showing an instance of the OP_SHORT TSF (named OP_Short_7) connected to two UUT output pins (J2-7 and J2-8).

```
<?xml version="1.0" ?>
<!-- generated with newWaveX v3.1.8 (http://www.newWaveX.com) -->
<Signal Out="OP_SHORT_7" xmlns="STDBSC" xmlns:this="MyTSFLib"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="MyTSFLib MyTSFLib.xsd">
  <this:OP_SHORT name="OP_SHORT_7" pin_hi="J2-7" pin_lo="J2-8" />
</Signal>
```

It would be simple to modify either of these TSFs to allow for impedances other than 0 Ohm. This would involve changing the fixed (0 Ohm) values for variable attributes. The TSFs would have additional attribute entries within the Interface section to allow the user to input different resistance values. An example of such an attribute would be as follows:

```
<xs:attribute name="res_value" type="Impedance">
  <xs:annotation>
    <xs:documentation>
      'res_value' is the value of the resistance required for the load
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
```

The data entry panel for the TSF would then show the additional attributes and require values to be added, as shown in figure 6.

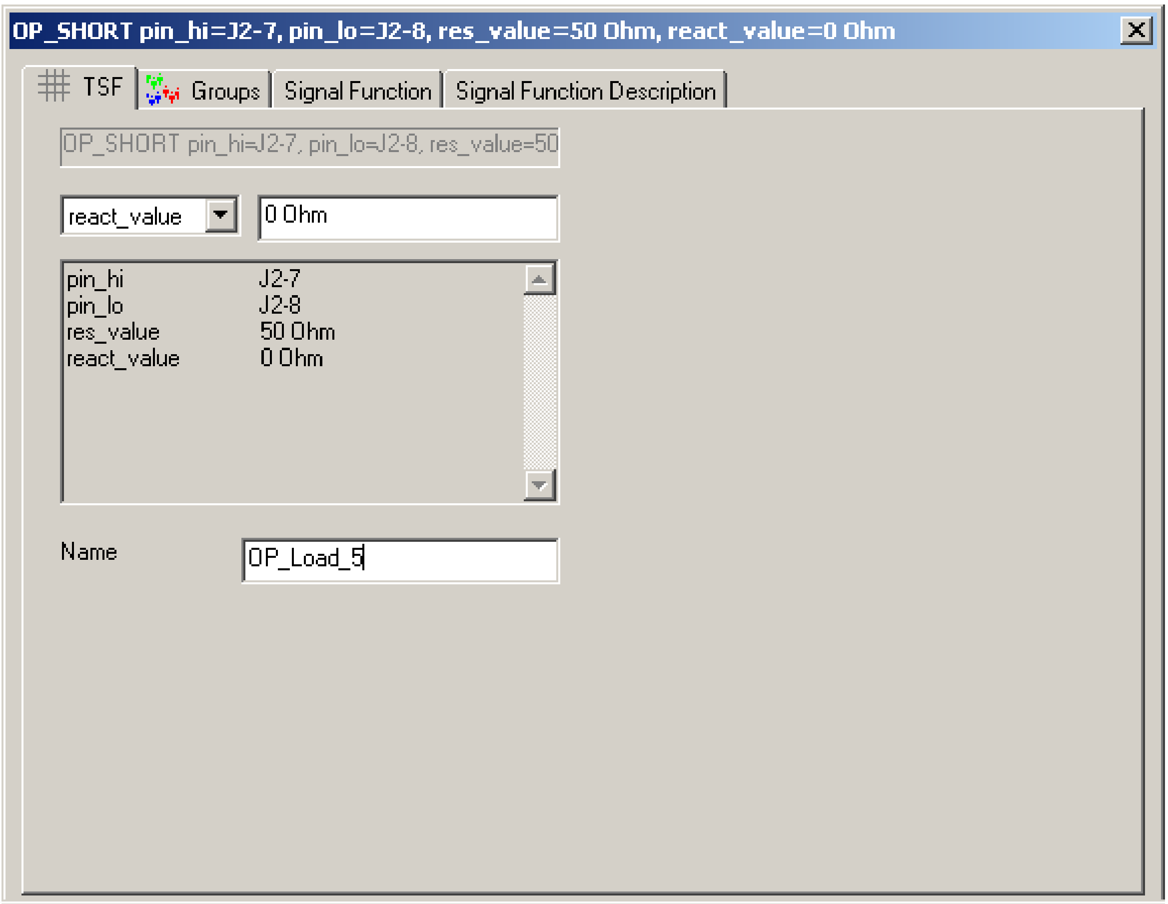

Figure 6—TSF data entry panel showing additional attribute values

This example shows a load in which both the resistance and reactance may be specified.

Why the Load BSC cannot be used as an input short to a UUT

The Load BSC is a transformation BSC and as such expects an input signal, therefore it would not be correct to use a Load BSC without a signal on its input. This can be seen in figure 7.

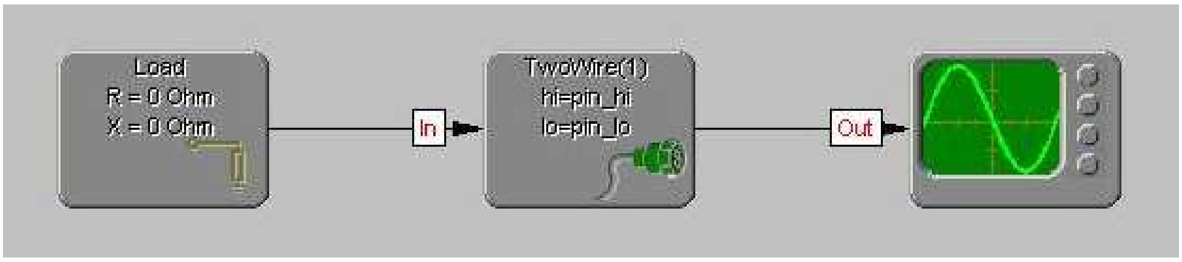

Figure 7—Using the Short BSC as an input (invalid use)

Initially, the model looks OK, but closer examination shows that it requires an input signal. This is even more obvious when the TSF XML is examined. The model section of the TSF is shown below, and in can be seen that the Load BSC (BadLoad) has an unsatisfied "In" parameter.

```
  <tsf:model>
    <Signal Out="UUT_Conn_4119" In="???">
      <Load type="Resistance" name="BadLoad" In="???" />
      <TwoWire type="Resistance" name="UUT_Conn" lo="pin_lo"
        hi="pin_hi" channelWidth="1" In="BadLoad" />
    </Signal>
  </tsf:model>
</tsf:TSF>
```

Summary

A short may be represented by two different BSCs, the Constant (of type Impedance) for an input and the Load BSC on the output of a UUT. The principles shown here, implemented within the newWaveX toolset, may be applied to any IEEE 1641 compliant tools.

Reference

Understanding IEEE 1641—Using Loads, A discussion on how loads and impedances are represented in IEEE 1641, AN1002 January 2010, EADS TES