The Use of Temperature in Simulations

This application note discusses how to create projects that use temperature as a variable in Microwave Office (MWO) and Analog Office (AO) simulations. This document includes:

- A brief discussion of how the built-in variables _TEMP and _TEMPK are used.
- References to a project used to illustrate the main points of this application note. Click the button below to open the project.

Open Pro

Introduction

Temperature controls the noise generation processes and the static operating point of nonlinear components and their dynamic behavior. With the necessary models for these components it is possible to calculate the DC operating point, noise figure, and both AC small and large signal properties of components as a function of temperature.

The AWR Design Environment uses two built-in variables _TEMP and _TEMPK, and functions such as ctok(x) and ktoc(x) to assist a designer in setting up projects to perform temperature sensitive simulations.

TEMP is a built-in variable in the AWR Design Environment intended for models that have a specific temperature parameter. _TEMP uses the units specified as global units (choose Options > Project Options and click the Global Units tab to select degree Kelvin, degree Celsius or degree Fahrenheit as the global units. The default value of this variable can be overwritten using an equation; for example the equation "_TEMP = 30"sets this global variable to 30 degrees Celsius if these are the set units.

_TEMPK is in degrees Kelvin and retains these units regardless of the global units setting. This variable is used to adjust the temperature for models without a temperature parameter. This variable only affects noise simulation of linear elements.

You can use both _TEMP and _TEMPK in equations to define the operating temperature of components. You can use equations to tie both linear and nonlinear temperature to the same value, and to assign one temperature value to all elements through hierarchy. Also, you can assign different temperatures to passive circuits, small signal amplifiers, Power Amplifier drivers and Power Amplifiers with a global temperature being used to define the base-plate or housing temperature, and equations added to define the unique temperatures of the high temperature components using dissipation and thermal resistance calculations.

Displaying Temperature Values Used in Simulation

You can view the current value of any variable by using the "variable:" notation. To return or expose the value of _TEMP, create the following equation inthe Global Definitions or any schematic window: _TEMP: and then simulate. You see that the variable _TEMP is set to 25. This is interpreted as 25 degrees Celsius. Now change the global units for temperature to degree Kelvin and resimulate. You see that _TEMP is set to 298.1 degree Kelvin. As previously explained, you can define the value of _TEMP by using an equation _TEMP = 30. To see the default value, overwrite this value with a user-specified value, and confirm the new value, create the following equations:

```
_TEMP:
_TEMP=30
_TEMP:
```

and simulate. The value of _TEMPK always returns the current value in degrees Kelvin. If no value is explicitly set, the value is 290 degrees Kelvin. These variables are displayed in the "A: Displaying Temperature Values" schematic.

Click to V

Different Temperature Situations in AWR

How these components use the built-in variables for temperature and derived temperatures depends upon the origin of the model parameters for the component. There are four different situations to be aware of when setting temperature in simulation:

- passive elements.
- nonlinear elements without _TEMP assigned.
- nonlinear elements with _TEMP assigned.
- netlists.

Temperature and Noise for Passive Structures

The AWR Design Environment supports both implicit and explicit methods when using temperature in simulations. The implicit method uses the built-in variable _TEMPK; the use of this approach is illustrated in the following schematic diagram. The schematic consists of an attenuator whose shunt and series elements are designed using the standard equations for a PI attenuator. This particular building block is chosen for this example because the NF equals the insertion loss when the physical temperature of a matched attenuator is set to the reference temperature. Here you can see that the resistors responsible for loss and noise generation do not have an explicit temperature defined. The built-in global variable _TEMPK controls the temperature of all such elements that possess loss, and therefore can generate noise.

The simulation uses a local copy of the built-in variable _TEMPK which is controlled by the Sweep Variable element. The temperature is swept between 0 and 400 degrees K. The X axis of the graph is set to use the sweep variable, _TEMPK. For the use of the SWPVAR see the MWO/AO Element Catalog Help documentation. In the example project, see the "B: Implicit Use of _TEMPK" schematic.

View B: Ir

When a specific temperature of an element needs to be defined other than that defined by the built-in global variable _TEMPK, you can create local variables and use elements with an explicit temperature parameter. In the following schematic you see that the REST element is used in place of the RES element. The REST element has a parameter for the temperature of the resistor. The **localTemp** variable is used to control the temperature of the element and uses the global units setting, which in this example is in degrees Celsius. Since the sweep is still in degrees Kelvin, a conversion is used for the temperature setting for each resistor, **T=ktoc(localTemp)**. The **ktoc** function converts from Kelvin to Celsius. Note that you could use the built-in _TEMP variable, or alternatively, you could not use the ktoc conversion and sweep the **localTemp** variable in degrees Celsius instead of degrees Kelvin. In every other respect, this design is identical to the previous design. The simulation results for these schematics are identical. In the example project, see the "C: Explicit Use of Temperature" schematic. Notice the changes shown in bold on the schematic.

View C: E

Temperature and Active Device Operation

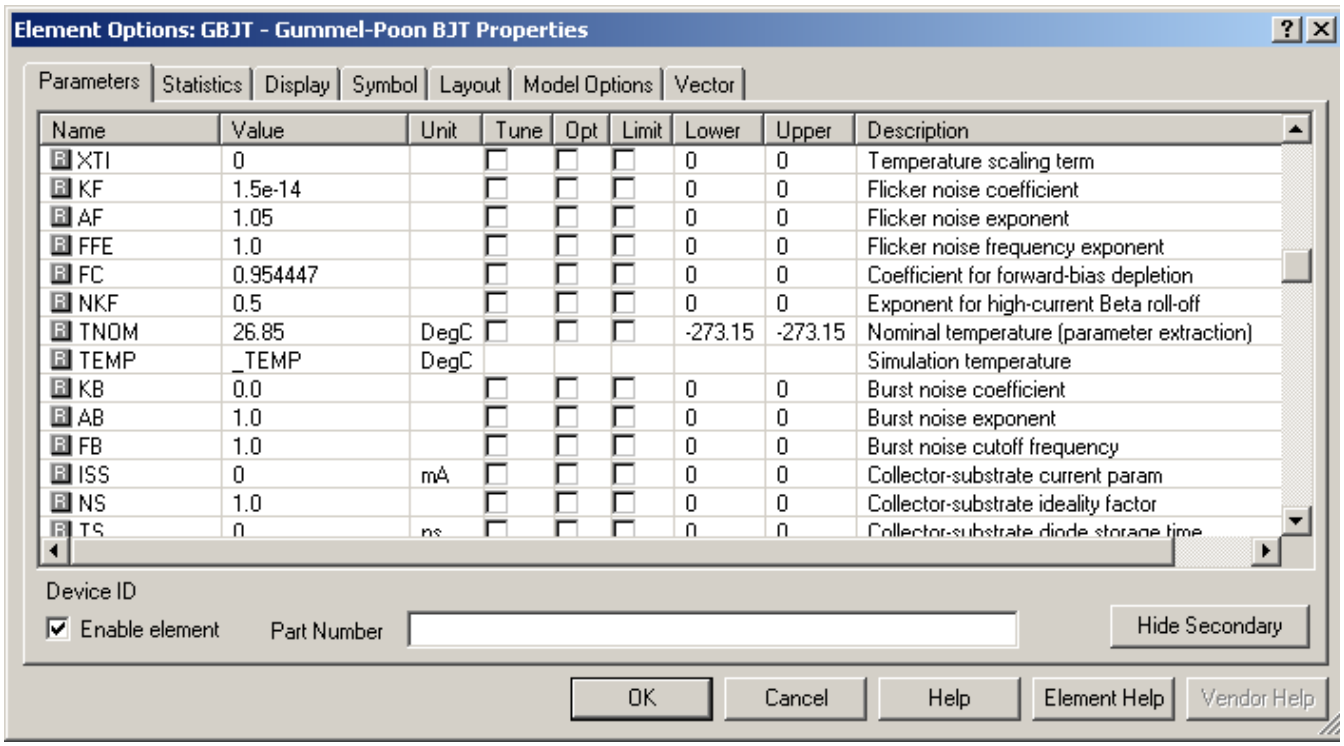
The principle of defining the temperature of circuit elements can be extended to active devices. The following is a schematic for a bipolar amplifier used later in the system noise figure calculations. The device is modeled using the nonlinear model and several elements to define the package parasitics. The schematic that defines the device model is shown following the test schematic. The top level schematic has _TEMP defined as a variable, and then the swept variable block is used to sweep the temperature from -270 to 100 degrees C in steps of 10 degrees C. The variable is set to be passed down through hierarchy designated by the solid red line surrounding the variable. (You can toggle this setting by using the Tune Tool and holding down the **Shift** key). In the example project, see the "E: Nonlinear Temperature_Schematic" schematic.

View E: N

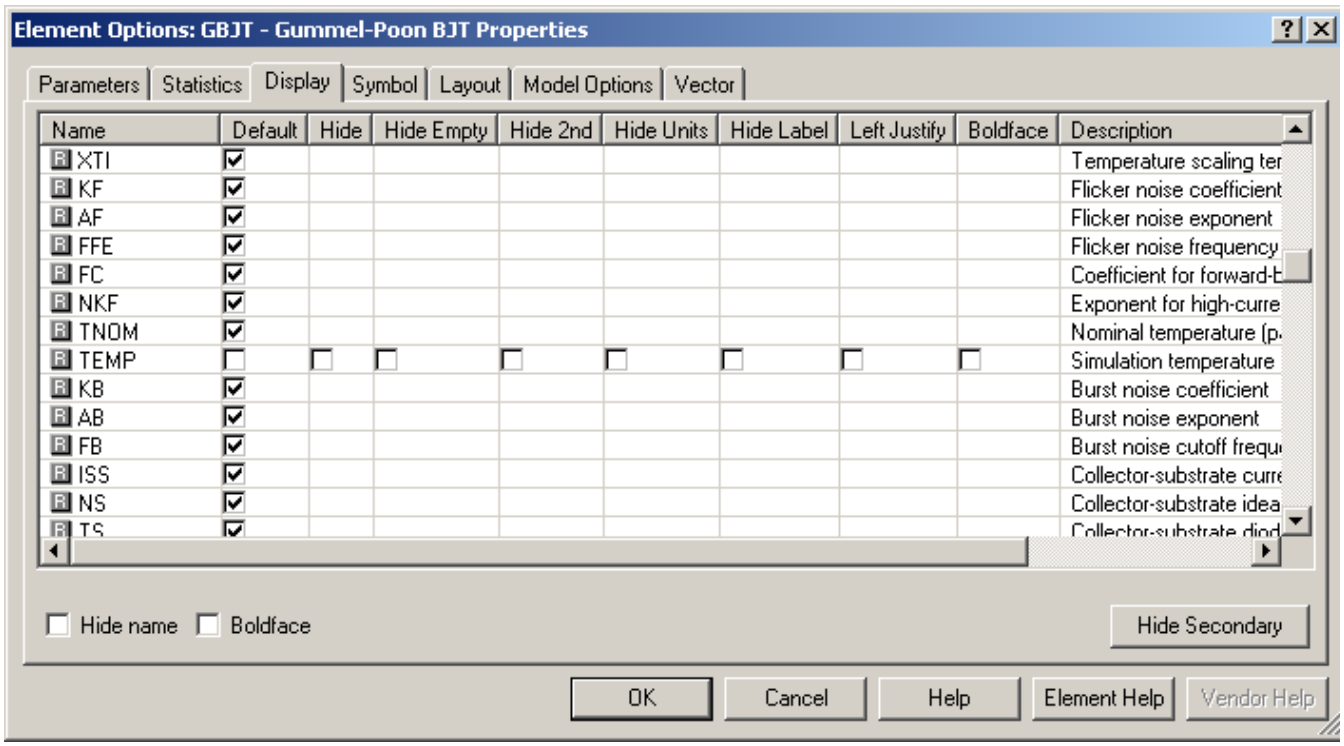
The schematic "BFR360F" defines the transistor parasitic elements.

View BFR

The following is the model of the transistor. The _TEMP variable has replaced the default value of the device temperature (TEMP = _TEMP).



The assignment of the temperature is displayed (it is normally hidden) by changing the display options of the parameter on the **Display** tab. You can change the default (hidden) by clearing the appropriate box as shown in the following figure.



Many of the device models for MMIC, RFIC and hybrid circuit design use this form when defining temperature. The model variable TEMP is the universal variable used by models within SPICE and other EDA simulations.

The AWR Design Environment supports the simulation of linearized noise figure, whereby the temperature and bias point dependent operating point is calculated before the small signal noise properties are calculated. The "E: Nonlinear Temperature_Schematic" has the graph with the noise and gain for the previous transistor circuit.

View E: N

In this example, the Gummel Poon model was pulled from the AWR Design Environment library of elements under Nonlinear > BJT. The appropriate model parameters were entered to model this part and _TEMP was assigned to the TEMP parameter. In the PDKs available for AWR, the temperature parameter is set to _TEMP by default.

In this example you can set the temperature to any variable name. AWR recommends that you use the _TEMP variable to maintain consistency between designs. If you want to assign different temperatures to different elements in the design, you will want to set a different variable name, however.

Temperature and Active Device Operation with a Device Defined by a SPICE Netlist

Often the device model is found from a component manufacturer's web site in the form of a SPICE compliant netlist. The following is one such netlist that has been imported into AWR. Note that on import, the syntax of the file is modified to conform to the AWR Design Environment netlist standard. The netlist contains both a circuit description of the transistor and an inline device model section with the normal BJT parameters. In SPICE syntax, the units are always assumed to be in base units (Farads, Amps, Henrys, etc). The base unit for temperature in spice is Celsius. The units for each type of variable are shown below. In the example project, see the "F: Nonlinear Temperature_Netlist" schematic.

View F: N

Then view the netlist in the "q2SC3357_v13" netlist.

View q2S

Typically the model does not explicitly define the temperature of the device. As previously mentioned, the SPICE standard temperature variable is TEMP. You must add this to the netlist to simulate at temperatures other than the default. Also, to enable this variable to be passed into the netlist from the parent schematic that owns the netlist, you must add an additional equation to the line that specifies the element node numbers. These additions are shown in the "qNE52418_v161" netlist or click the below to see this netlist. find the lines that start with !NOTICE to find lines below that are different.

View qNE

The transistor is defined by a subcircuit (a netlist in this example) and the deviceTemp variable has in turn been equated to a _TEMP variable.

The "F: Nonlinear Temperature_Netlist" schematic shows the results of the gain and NF simulations.

View F: N

In this case, the project units for temperature were degrees C so assigning _TEMP to the deviceTemp parameter set the correct temperature in the netlist. If the project units for temperature were other than degrees C, the temperature passed to the netlist needs to be converted to Celsius. For example, the same results can be achieved when the global unit for temperature is set to degrees K with the schematic as follows. Click below to see the same schematic with these settings.

View Cha

The differences to note are displayed in **bold** and are that the swept values are now from 73 to 313 in steps of 20 since the units are degrees K. The deviceTemp value passed to the netlist is now converted to degrees C using the ktoc function. So deviceTemp =ktoc(_TEMP) converts _TEMP from Kelvin to Celsius before the value is passed to the netlist.

Again, you can use any variable value to pass temperature, however _TEMP should be used to maintain consistency between designs.

How to Sweep Passive and Active Temperature Through Hierarchy

In IC design, the temperature of each component is normally set to the same value. In this situation, each nonlinear model typically has its explicit temperature set to _TEMP in the PDK model set. The best approach is to assign the temperature at the top level schematic and pass the values down through the hierarchy. Assign _TEMP and _TEMPK to use the same swept values. Because _TEMP uses the temperature units set for the project and _TEMPK is always in Kelvin, built-in equations are used to make the two temperatures the same. In the example project, see the "G: Linear and Nonlinear Temperature Schematic" schematic.

View G: L

In this example, the unit for temperature is set to degrees Celsius. In the top level schematic, the value for _TEMP is set to sweep its value and pass down its value. This is done by using the SWPVAR block and assigning the _TEMP variable to pass down its value. Again, the global temperature unit is Celsius so the swept values are in Celsius. _TEMPK is assigned to follow the _TEMP value, but it must be converted from degrees Celsius to degrees Kelvin using the equation "_TEMPK = ctok(_TEMP)". Note that the temperature conversion depends on what the global unit setting is for temperature.

The schematic "BFR360F W Resistor" shows the lower level in the hierarchy where a resistor is added at the input to show the effects of _TEMPK.

View BFR

The "G: Linear and Nonlinear Temperature Schematic" schematic shows the swept Noise Figure versus temperature.

View G: L

Note, that different results are achieved if the different temperature settings are not passed down through the hierarchy. When neither _TEMP nor _TEMPK are passed down, the result is shown on the "G: Linear and Nonlinear Temperature Schematic". Notice the red box around both equations is now removed.

View Res

The noise figure is flat because the swept temperature is not passed, and so the default values of 25 degrees C for _TEMP and 290 degrees for _TEMPK are used in simulation.

When just _TEMP is passed through hierarchy, the result is shown on the following graph. Notice only the _TEMP equation has the red box around the equation.

View Res

When just _TEMPK is passed through hierarchy, the result is shown on the following graph. Notice only the _TEMPK equation has the red box around the equation.

View Res

Group Design Issues

When sharing designs between different designers or different projects in the AWR Design Environment, units can be an issue. If you don't use any variables anywhere in a design, there are no issues. However, variables do not use unit scaling and so it is possible for designs to be passed between AWR projects that are not identical. The following are some possible techniques to address this issue:

- Set up a common set of global units to use in all projects.
- Agree to use base units for all projects.
- Agree to select the **Dependent parameters use base units** setting prior to doing any design

The following example demonstrates:

Assume a designer uses capacitance units of nF and a variable assigned to a capacitance of 0.01. If this schematic is exported and then imported to a project with units of pF, the software doesn't know how to scale the value of a variable (variables can be combination of equations, etc. so this cannot be done in a general sense). When this schematic is imported the variable still has a value of 0.01. However, now that units are pF, the value is off by three orders of magnitude. Using base units for dependent parameters (Farads) in this example allows for easy design sharing.

Dependent parameters use base units can be set globally by choosing Options > Project Options</emphasis> and clicking the **Schematics /Diagrams tab**, or it can be set locally for each schematic. **This option is not set by default**. This setting will set any parameter using a variable value to base units. Base units are any unit type without a modifier, such as Farads, Henrys, and Amps, instead of the specified units in the global units setting. When designers share designs, the software does not know what the global units setting are in the original design. Always using base units for any parameter using a variable ensures that values are set appropriately any time a design is shared.

If you are using the **Dependent parameters use base units** setting, the units for temperature would display as degrees Kelvin, and you should sweep temperature values in Kelvin. If using a hierarchical design, it is preferable to make this setting for the entire project, if not you will need to make the setting for any schematic in the hierarchy with a temperature setting. If using this setting in this example, _TEMP will always be in degrees Kelvin and should use those values to sweep. This setting would also keep _TEMP and _TEMPK in the same units and no conversion is necessary.

To demonstrate the Dependent parameters use base units option, the linear simulation was repeated with the explicit temperature model. However, now this option is set locally (right-click on the schematic in the Project Browser, choose Options and click the Schematic tab), overriding the default setting. Now the temperature units are displayed in base units and the equation to convert from Celsius to Kelvin is not required. Again, the simulation results are identical. In the example project, see the "D: Explicit Use of Temperature Base Units" schematic.

View D: E

Conclusions

The AWR Design Environment supports several methods to define and manipulate the temperature of electronic devices, and therefore to enable comprehensive simulation of noise processes, device operating point parameters such as gain and full nonlinear behavior.

This application note has illustrated the methods that can be employed to perform noise and nonlinear simulations when the operating temperature needs to be accounted for.