

Parameter Trees

Walter Katz

SiSoft

DAC IBIS Summit
San Francisco, CA
June 5, 2012

Overview

- What is a Parameter Tree
- Example Parameter Tree
- IBIS AMI Parameter Trees
- Leaves in a .ami File
- Syntax and Context Rules
- ISS Parameter Tree Context Rules

What is a Parameter Tree

From http://en.wikipedia.org/wiki/Tree_structure

A **tree structure** is a way of representing the hierarchical nature of a structure in a graphical form. It is named a "tree structure" because the classic representation resembles a tree, even though the chart is generally upside down compared to an actual tree, with the "root" at the top and the "leaves" at the bottom.

Every finite tree structure has a member that has no superior. This member is called the "root" or root node. It can be thought of as the starting node. The converse is not true: infinite tree structures may or may not have a root node.

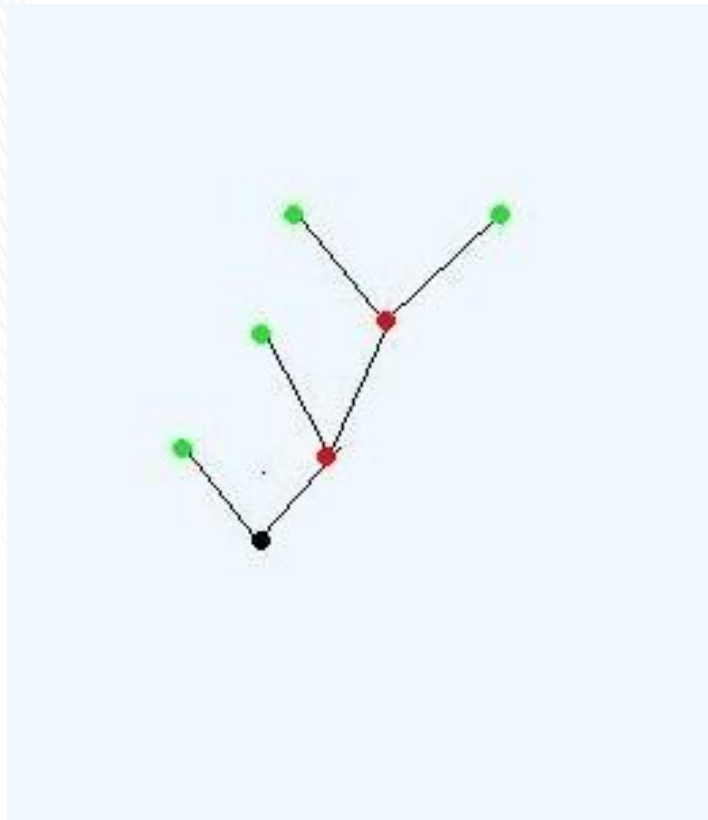
The lines connecting elements are called "branches", the elements themselves are called "nodes". Nodes without children are called leaf nodes, "end-nodes", or "leaves".

The names of relationships between nodes are modeled after family relations. The gender-neutral names "parent" and "child" have largely displaced the older "father" and "son" terminology, although the term "uncle" is still used for other nodes at the same level as the parent.

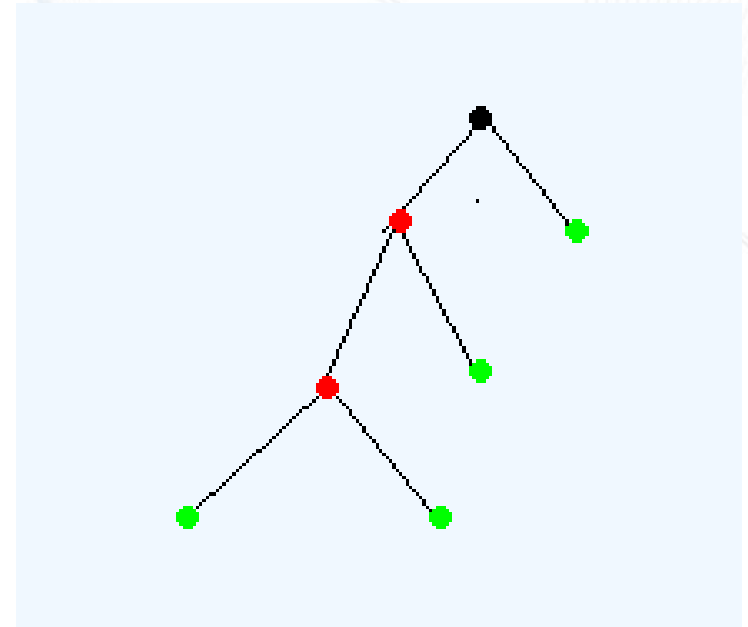
- A node's "parent" is a node one step higher in the hierarchy (i.e. closer to the root node) and lying on the same branch.
- "Sibling" ("brother" or "sister") nodes share the same parent node.
- A node's "uncles" are siblings of that node's parent.
- A node that is connected to all lower-level nodes is called an "ancestor". The connected lower-level nodes are "descendants" of the ancestor node.

Black – Root Red – Node Green – Leaf

USA Tree



Australian Tree



Example Parameter Tree

```
(Root
  (Leaf1 value1)
  (Leaf2 value2)
  (Node1
    (Leaf3 value3)
    (Node2
      (Leaf4 value4)
      (Leaf5 value5)
    )
    (Node3
      (Leaf6 value6)
    )
  )
)
```

All Nodes and Leaves have one Parent Branch

- IBIS 5.0 uses the word Branch to describe Nodes.
- IBIS 5.0 uses the word Leaf to describe Nodes that do not have Children

IBIS AMI Parameter Trees

- IBIS AMI uses a Parameter Tree to transfer information into and out of the AMI executable model.
- IBIS AMI also uses a Parameter Tree in the .ami file to define parameters that get passed into and out of the AMI executable and parameters that are informational to the EDA tool. AMI parameter may have one or more than one possible values.
- The User/EDA tool picks the value of each parameter using, for example, GUI selection.
- A leaf in the Parameter Tree that is passed into and out of the AMI executable is a node in the .ami file that contains leaves that describe the parameter.

Leaves in a .ami File

- Leaves in a .ami Parameter Tree describe the properties of their Parent Node (AMI Parameters) which are Leaves in the tree passed to the Executable Model.
- Leave in an .ami must be one of the following
 - Type
 - Usage
 - Description
 - Labels
 - Format Leaves
 - Value
 - List
 - Corner
 - Range
 - Steps
 - Increment
 - Table

Type

- Integer
 - 5
- Float
 - 5, 5.0, 0.5e1, 0.5E1
- UI (percentage of bit time)
 - .1
- String
 - “ABC”, “A B C”
- Boolean
 - True, False

Usage

- In
 - Parameter is passed into executable model
- Out
 - Parameter is passed out of executable model
- InOut
 - Parameter is passed into and out of executable model
- Info
 - Parameter is not passed into or out of executable model
 - Parameter is informational to either EDA tool or User

Description and Labels

- Informational to EDA tool and User. That can be used to describe the Root and any Node in the Parameter Tree.

Format Leaves

- Value
 - A single value
- List
 - A list of values
- Corner
 - IBIS Process Corner (typ, min, max)
- Range
 - Range between a minimum and maximum value
- Steps
 - Range between a minimum and maximum value divided into an integer number of steps
- Increment
 - Range between a minimum and maximum value divided into uniform steps of a specified size

Format Leaves (continued)

- Table
 - Tabular data consisting of rows and columns
 - Each row has the same number of columns
 - All data in each column must be the same Type

AMI Parameter Tree Example

```
(Rx_Root
  (Description "Example .ami File")
  (Reserved_Parameters
    (Ignore_Bits (Usage Info) (Type Integer) (Value 500000))
    (Max_Init_Aggressors (Usage Info) (Type Integer) (Value 25))
    (Init_Returns_Impulse (Usage Info) (Type Boolean) (Value True))
    (GetWave_Exists (Usage Info) (Type Boolean) (Value True))
    (Rx_Receiver_Sensitivity (Usage Info)(Value .05) (Type Float))
  )
  (Model_Specific
    (Framis (Usage In)(Corner .2 .1 .5) (Type Float) (Description "Controls value of the Rx Framis."))
    (DFE (Usage In)(Range .1 -.2 .2) (Type Float) (Description "Controls DFE equalization."))
  )
)
```

String passed to DLL when IBIS process corner is min (slow), and user sets DFE to .05

```
(Rx_Root (Framis .1) (DFE .05))
```

Parameter Tree Futures

- Dependency Table
- IBIS ISS Parameter Passing
- Converting .ibs to Parameter Tree Format

Dependency Table

- Dependency Tables currently under discussion (BIRD 150).
- Dependency Tables allow the User to select one or several “independent” Info parameter values, which would determine values of other “dependent” parameter values. An example of this using the parameter My_Corner with values “ExtremeSlow”, “Slow”, “Typ”, “Fast” and “ExtremeFast” to define five process corners. My_Corner would be an independent parameter in the Dependency Table which would in turn determine the values of dependent Info and In parameters. Info parameters such as Touchstone File, impedance, and capacitance would be passed to the EDA tool. In parameters such as Peak-to-Peak voltage swing, would be passed into the Executable Model.

Example Dependency Table

```
(Rx_Root
  (Reserved_Parameters
    ...
  )
  (Model_Specific
    ( My_Corner (Usage Info)(List "ExtremeSlow" "Slow" "Typ" "Fast" "ExtremeFast" ) (Type String)
      (Description "This model has 5 process corners."))
    (Framis (Usage In)(Range 0. 0. 1.) (Type Float) (Description "Controls value of the Rx Framis."))
    (Framis_Corners (Usage Dependency_Table) (Type String Float)
      (Table
        (Labels "My_Corner In" "Framis Out_Match")
        ( "ExtremeSlow"      0.0      )
        ( "Slow"             0.2      )
        ( "Typ"              0.3      )
        ( "Fast"             0.4      )
        ( "ExtremeFast"      1.0      )
      )
    )
  )
)
```

IBIS 5.0 file Currently

[Model] Rx

Model_type Input

C_comp 0.0p 0.0p 0.0p

Vinh = 0.55

Vinl = 0.45

[Algorithmic Model]

Executable Windows_VisualStudio7.1.3088_32 Rx.dll Rx.ami

[End Algorithmic Model]

[Temperature_Range] 25 100 0

[Voltage Range] 1.0 1.0 1.0

[GND Clamp]

| | | | |
|------------|------------|------------|------------|
| -2.500E+00 | -5.000E-02 | -5.555E-02 | -4.545E-02 |
| 0.000E+00 | 0.000E-02 | 0.000E-02 | 0.000E-02 |
| 2.500E+00 | 5.000E-02 | 5.555E-02 | 4.545E-02 |

[End]

IBIS 7.0 file (Parameter Tree)

(Model

(Model_Name "Rx") (Model_type "Input")

(C_comp (Corner 0.0p 0.0p 0.0p)) (Vinh 0.55) (Vinl 0.45)

(Algorithmic_Model

(Executable

(Table

(Labels "Compiler"

"Executable"

"AMI_File")

("Windows_VisualStudio7.1.3088_32" "Rx.dll"

"Rx.ami")

)

)

)

(Temperature_Range (Corner 25 100 0))

(Voltage_Range (Corner 1.0 1.0 1.0))

(GND_Clamp

(Table

(Labels "Voltage" "I Typ" "I Min" "I Max")

(-2.500E+0 -5.000E-2 -5.555E-2 -4.545E-2)

(0.000E+0 0.000E-2 0.000E-2 0.000E-2)

(2.500E+0 5.000E-2 5.555E-2 4.545E-2)

)

)

)

Equation Based GND_Clamp

(GND_Clamp

(Equation " $I=a+b*V+c*V^2+d*V^3$ ")

(Table

| (Labels | "Corner" | "a" | "b" | "c" | "d" |
|---------|----------|-----|-------|-----|------|
| (| "Typ" | 0.0 | 0.020 | 0.0 | 0.0) |
| (| "Min" | 0.0 | 0.022 | 0.0 | 0.0) |
| (| "Max" | 0.0 | 0.018 | 0.0 | 0.0) |

)

)

Syntax and Context Rules

- The Syntax of an IBIS Parameter Tree is well defined and stable.
- The .ami Parameter Tree file has specific context rules.
 - Root must have only two branches to nodes Reserved_Parameters and Model_Specific
 - Reserved_Parameters must only contain nodes specified in the IBIS specification.
 - Usage has specific meanings
- The Parameter Tree file for .ibs files or wrapping ISS subckts in [External Model] will have different context rules

ISS Subckts Parameter Tree

Context Rules – Preliminary

- There will not be Model_Specific and Reserved_Parameter branches.
- Usage will only have the allowed values of Info and In.
 - In
 - » The parameter will be passed to the ISS subckt call as an ISS parameter on the subckt instance line.
 - » Parameters names must comply with the ISS parameter naming rules
 - Info
 - » Controls setting up the values of In parameters, for example used in dependency tables.

Conclusion

- Parameter Trees are supported by all EDA vendors
- Parameter Trees are easily extended to handle new parameters
- Parameter Trees are easily extended to handle new parameter data types
- Parameter Tree can be used for other IBIS models such as ISS subckts in [External_Model].