IBIS-AMI parameters in Matlab/Octave

Adge Hawes, IBM

IBIS summit, Jan 2013

IBM multi-dimensional parameters

- IBM IBIS-AMI models use multi-dimensional parameters
- Parameters may vary with
  - Corner (EC, WC, NC, BC)
  - VTR/VTT (1.20,1.05)
  - Gain (0, 5, 9, 12, 15, 20, 25)
  - PEAKF (00, 01, 10, 11)
  - etc.
- Spec sheet shows tables

RJ	**EC**	**WC**	**NC**	**BC**
1.20	230	220	180	150
1.05	240	230	190	160

- Need N-dimensional storage for processing

Source If Applicable

Why not use the (Format List) construction?

- The Format List construction has been used to select user parameters
- User must know which parameter to use
  - Data sheet as reference
- List is one-dimensional
- `(Format List 150 160 180 190 220 230 240)`

N-dimensional Matlab/Octave variable types

- Structures
  - Named fields
  - Mixed types
  - Similar to C `struct`
  - e.g. `param.name="txic";`
- Matrices
  - N-dimensional arrays
  - Numeric only
  - Generated with [ ]
  - Indexed with ( )
    -
```
value(3,4,1)=3.14;
val=[
  230,220,180,150 ;
  240,230,190,160
];
```
- Character arrays
  - Vector of character strings
  - Strings are equal length (2D array of chars)
    -
```
corner(1)="Nominal";
corner(2)="Worst  ";
```
- Cells
  - N-dimensional arrays
  - Any type, mixed
  - Generated and Indexed with { }
    -
```
var{1}=3.14;
var{2}="tx_ic.s4p";
```

Example parameter

```
corners={"EC","WC","NC","BC"};
CORNERDIM=struct('var','{rxcorner}', 'mode', 'match' , 'case',
{corners});

RJRX=++n;
param(n).name='rjrx';
param(n).dim=CORNERDIM;
param(n).attr.Description='Random Jitter, estimated, fs' ;
param(n).attr.units=”s”;
param(n).attr.scale=”f”;
param(n).value=[
%  'EC', 'WC', 'NC', 'BC'
   264 , 264 , 215 , 194
];
```

Parameter structure fields

- `name`
  - Parameter name in config
  - Mandatory
- `attr`
  - Parameter attributes
  - Further subfields:
    - AMI: `Usage`, `Type`, etc.
    - General: `units`, `scale`, etc.
- `comment`
  - Extra info that may be added to config
  - Note: not the same as “Description”
- `dim`
  - Dimension descriptors
- `value`
  - N-dimensional matrix or cell of values

The .dim field

- Describes each dimension of the value matrix
- Is an N-element vector (N= number of dimensions)
- Each element is a structure
  - `var`: The selection variable
  - `mode`: The method of matching selection variable
    - `match`
    - `ladder`
    - `step`
    - `pwl`
  - `case`: Values that are to be matched to the selection variable

Selection variables

```
corners={"EC","WC","NC","BC"}; CORNERSL=length(corners);
        [ EC , WC , NC , BC ]= enum;

param(n).name="txcorner";
param(n).attr.Description="User selected Tx corner";
param(n).attr.Type="String";
param(n).attr.Format="List";
param(n).attr.Usage="In";
param(n).attr.List=corners;

rates=  [ 28.05e9, 25.78125e9, 14.025e9, 10.3125e9, 8.5e9, 6.25e9];
RATESL=length(rates);
        [   G28  ,      G25  ,    G14  ,     G10  ,  G8,     G6  ] = enum;
ratestr={ "28G"  , "25G"     , "14G"   , "10G"    , "8G" , "6G"  };

AVTRs  =[  1.2   , 1.05  ];
AVTRstr={"1.20V", "1.05V"};AVTRSL=length(AVTRs);
        [ V120  ,  V105  ]=enum;
```

More complicated example

```
AVTRDIM = struct('var', '{AVTR}', ...
                 'mode','ladder', ...
                 'case',[1.05 1.20] );
GAINDIM=struct( 'var','{gain}', ...
               'mode','match',  ...
               'case', {gains} );


param(4).name='RxDCD';
param(4).dim=[AVTRDIM,CORNERDIM,GAINDIM];
param(4).value = cat(3,  ...
  [ 1  0  2  3 ;        ...
    3  2  4  5 ],       ...
                        ...
  [ 0  0  1  1 ;        ...
    1  1  2  2 ],       ...
                        ...
  [ 1  0  2  3 ;        ...
    3  2  4  5 ] );
```

Parameter processing

- Octave(/Matlab) scripts both generate and process parameters
- Fully defined parameter can be converted to
  - IBM Internal simulator configuration (HSSCDR - `.hss`)
  - IBIS-AMI Dependency Tables (BIRD 150 - `.ami`)
  - Proprietary conditional expressions (BIRD 155 - `.ami`)
  - XML format
  - Documentation