

Paramssets: Candidate CMC Standard Model File Format

Geoffrey Coram

The World Leader in High-Performance Signal Processing Solutions





Paramsets

- ◆ **Verilog-A replacement for .model cards:**
 - Collects overrides for module parameters
 - Differentiates between “model” and “instance” parameters

- ◆ **Added to Verilog-AMS LRM 2.2 as part of compact modeling extensions**

- ◆ **Significantly more flexible than .model**
 - May be “chained”
 - More flexible automatic selection (binning)



Paramset syntax

```
module psp101(d, g, s, b);  
    ...  
endmodule  
  
paramset nch psp101;  
    parameter l = 90n from [90n:150n);  
    parameter w = 100n from [50n:1u];  
    .l = l;  
    .w = w;  
    .tox = 10n;  
    .phi = 0.6;  
endparamset
```

Paramset syntax (2)

```
module psp101(d, g, s, b);  
    ...  
endmodule
```

} Underlying module
definition contains
all behavioral code

```
paramset nch psp101;  
    parameter l = 90n from [90n:150n);  
    parameter w = 100n from [50n:1u];  
    .l = l;  
    .w = w;  
    .tox = 10n;  
    .phi = 0.6;  
endparamset
```

l(g,d) <+ ddt(...);

Paramset syntax (3)

```
module psp101(d, g, s, b);  
    ...  
endmodule  
  
paramset nch psp101;  
    parameter l = 90n from [90n:150n);  
    parameter w = 100n from [50n:1u];  
    .l = l;  
    .w = w;  
    .tox = 10n;  
    .phi = 0.6;  
endparamset
```

Diagram illustrating the paramset syntax. The text "paramset nch psp101;" is shown. A red circle highlights "nch", with a red arrow pointing to the label "paramset name". A blue circle highlights "psp101", with a blue arrow pointing to the label "module name".

Paramset syntax (4)

```
module psp101(d, g, s, b);
```

```
...
```

```
endmodule
```

```
paramset nch psp101;
```

```
parameter l = 90n from [90n:150n);
```

```
parameter w = 100n from [50n:1u];
```

```
.l = l;
```

```
.w = w;
```

```
.tox = 10n;
```

```
.phi = 0.6;
```

```
endparamset
```

Parameters
of paramset
are “instance
parameters”

Paramset syntax (5)

```
module psp101(d, g, s, b);
```

```
...
```

```
endmodule
```

```
paramset nch psp101;
```

```
parameter l = 90n from [90n:150n);
```

```
parameter w = 100n from [50n:1u];
```

```
.l = l;
```

```
.w = w;
```

```
.tox = 10n;
```

```
.phi = 0.6;
```

```
endparamset
```

Module parameters are
set with an initial “.”

Paramset syntax (6)

```
module psp101(d, g, s, b);
```

```
...
```

```
endmodule
```

```
paramset nch psp101;
```

```
parameter l = 90n from [90n:150n);
```

```
parameter w = 100n from [50n:1u];
```

```
.l = l;
```

```
.w = w;
```

```
.tox = 10n;
```

```
.phi = 0.6;
```

```
endparamset
```

} Other module parameters are
“model parameters”
(can't be set for instance)



Paramset chaining

- ◆ Paramset may reference another paramset
(.model card must reference a built-in model)

```
paramset nch_dogbone nch;  
    parameter l = 90n from [90n:150n);  
    parameter w = 50n from [50n:100n);  
    .absource = (w * 100n) + 120n*120n;  
    ...  
endparamset
```

Local variables

- ◆ Local variables may be declared and used

```
paramset nch_dogbone nch;  
  parameter l = 90n from [90n:150n);  
  parameter w = 50n from [50n:100n];  
  real weff = (w - 10n);  
  .w = weff;  
  .absource = (weff * 100n) + 120n*120n;  
  ...  
endparamset
```

Paramset overloading

- ◆ Paramsets may be overloaded (several paramsets with the same “name”)
- ◆ Trivial to do L/W MIN/MAX model selection:

```
paramset nch psp101;
```

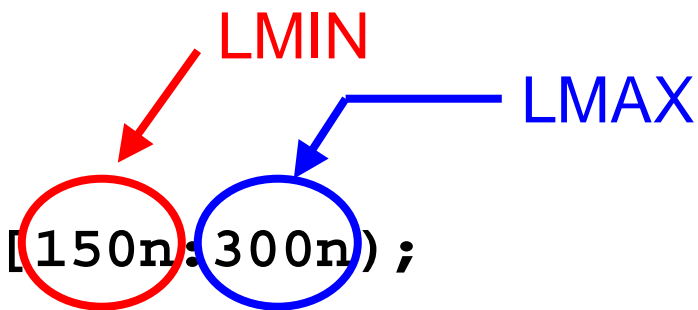
```
parameter l = 90n from [90n:150n);
```

```
...
```

```
paramset nch psp101;
```

```
parameter l = 150n from [150n:300n);
```

```
...
```



Paramset overloading (2)

- ◆ More advanced automatic model selection:
 - Not just L and W, but any paramset parameter
 - Also localparam

```
paramset nch psp101;
  parameter l=90n, w=100n;
  parameter absource = l*w from [10f:100f);
  localparam lssource = w+360n from [400n:500n);
  ...
paramset nch psp101;
  parameter l=90n, w=100n;
  parameter absource = l*w from [100f:200f);
  localparam lssource = w+360n from [500n:600n);
  ...
```



Paramset overloading (3)

◆ Paramset selection rules:

- Parameters specified for an instance must be parameters of the paramset (not of the module)
- Parameter values must be within the allowed ranges for the paramset
- Localparam values must be within the allowed ranges

◆ Secondary rules:

- Paramset with fewer unspecified parameters shall be preferred
- Paramset with more localparams (with ranges) shall be preferred

◆ Error if:

- Paramset specifies out-of-range value for module parameter
- More than one paramset satisfies above rules

◆ See Verilog-AMS LRM 2.2, section 7.3.2 for exact rules

Paramset overloading (4)

◆ Pre-layout and post-layout paramsets:

```
paramset nch psp101; // pre-layout
parameter l=90n, w=100n;
.l=l; .w=w; .absource = l*w;
```

...

If absource is specified, must use post-layout

```
paramset nch psp101; // post-layout
parameter l=90n, w=100n;
parameter absource = l*w from [100f:200f);
.l=l; .w=w; .absource=absource;
```

...



Statistics

- ◆ **Global and local mismatch supported per LRM**
- ◆ **LRM specifies distribution functions:**
 - `$rdist_normal`, `$rdist_uniform`, ...
- ◆ **Correlation possible with local variables**
- ◆ **Some ambiguity in LRM specification needs to be addressed**



Operating-point values

- ◆ Paramset variables can be made “operating-point values” just like module variables

```
(* desc="transconductance" *) real gm;
```

- ◆ May be used to re-name or hide op-pt variables of underlying module



Subcircuits

- ◆ **Verilog-A modules may contain other modules**
 - e.g., early definition of PSP included JUNCAP2 diodes
- ◆ **This provides ability to make “supermodels” and probe currents and voltages**



Model File Format Requirements

◆ Per e-mail from Marq Kohl:

- ✓ **MUST use IEEE 754 floating-point format**
- ✓ **SHOULD understand unit prefixes**
- ? **MUST support hierarchy**
- ? **MUST support scope (reference global variables)**
- ✓ **SHOULD be compatible with V-AMS paramsets**
- ☞ **MAY use international characters**
- ? **SHOULD allow multiple model sets (SiP, PPT)**
- ? **SHOULD allow integration in Doxygen**
- ✓ **SHOULD be case-sensitive**
- ✓ **MAY have a preprocessor**
- ? **MUST provide one model card as a subset of another**
- ✓ **SHOULD support local variables**
- ✓ **MAY allow simulator-specific convergence parameters**
- ? **MUST handle statistical data separate from the parameter sets**
- ✓ **SHOULD be able to correlate statistical parameters**



Verilog Parameters

- ◆ **A significant potential problem with Verilog is that PARAMETERS are considered CONSTANTS per IEEE 1364 (digital) Verilog**
 - **Can't sweep a parameter**
 - **Can't use out-of-module references to set parameters, because value may be "elaboration-order dependent"**
- ◆ **But ... 1364 Verilog doesn't know what a "dc sweep" is; we should be able to specify extensions for analog simulators**



CMC Standard?

- ◆ **Is the Verilog-A paramset a candidate for the CMC's standard model file format?**
 - **Has many useful features**
 - **Uses simulator-independent syntax**
 - **Ties nicely to Verilog-A as compact modeling language**

- ◆ **What other features are needed?**