

JavaScriptを利用したPcell作成

はじめに

JavaScriptは多くのユーザが使用する、現在WEB上でもっともポピュラーなスクリプト言語の一つです。Expertではカスタマイズ機能およびPcell(Parameterized cell)を作成するためにJavaScriptを利用することが可能です。

このアプリケーションノートではJavaScriptを使ってPcellを作成する際のいくつかのヒントを紹介します。

JavaScript Pcellの優位性

JavaScript PcellはこれまでのLISA(Language for Interfacing Silvaco Application) Pcellに比べいくつかの利点があります。

- 書きやすさ**
JavaScriptはLISAよりも多くの関数を持っています。
- WEB上の例題**
JavaScriptはWEB上で大変人気の高いスクリプト言語であり、JavaScriptの使い方はネット上から容易に検索可能です。
- コールバック機能の追加**
Expert ではJavaScriptを使用することでPcellのコールバック機能を利用できるようになりました。この機能によりPcellのパラメータの値を簡単にコントロールすることが可能です。

JavaScriptのための3つのマニュアル

- JavaScriptのコア部分のリファレンスマニュアルは以下から参照できます。
<https://developer.mozilla.org/ja/javascript>
ExpertではMozillaプロジェクトのJavaScriptエンジンを内蔵しており、このマニュアルがJavaScript言語の正式なマニュアルとなります。
- ExpertのJavaScript API マニュアル
ExpertのメニューからJavaScript API のマニュアルを参照できます。
ヘルプ >> JavaScript API reference
このマニュアルにはレイアウト編集にかかわる全てのJavaScript APIの情報が載っています。
- JavaScript とLISA のコマンド比較表
“インストールディレクトリ”/lib/expert/”バージョン名”/docs/lisa_javascript_table.pdf
上記のマニュアルにはJavaScriptとLISAのコマンドを比較した表が載っています。すでにLISA言語について熟知しているユーザには、とても参考になる資料です。

JavaScriptを使用する準備

ExpertでJavaScriptを利用するには、最初にExpertのメニューからJavaScriptをスクリプトエンジンとして選択する必要があります。

ツール>>スクリプト>>JAVAスクリプト

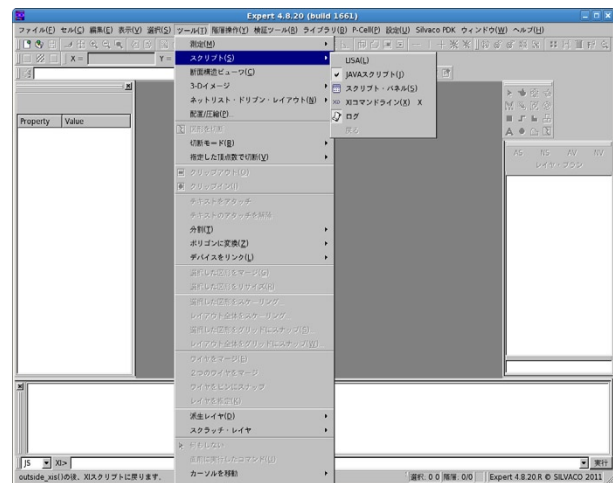


図 1 JavaScriptの設定

ログウィンドウの確認

Expertのログウィンドウには実行した全てのコマンドが表示されます。そのため特にマニュアルを参照しなくても、GUIから入力したコマンドのJavaScript APIをこのログから確認することができます。このログウィンドウはXI(Expert Interface)コマンドをすぐに確認するのにとても有用です。



図 2 ログウィンドウ

コールバック機能

ExpertではJavaScriptを使ったPcellにのみ、Pcellパラメータのコールバック機能が利用可能です。（LISAではサポートされていません）

シルバコのWEBサイトの例題集の中に詳細なコールバック機能の例題が載っています。詳しい内容はこちらを参照してください。

<http://www.silvaco.com/examples/expert/section3/example4/index.html>

外部のJavaScriptファイルを利用したPcellのコンパイル

Expertが内蔵するXISクリプトパネルはPcellを作成するのに優れたエディタですが、時にはCVSなどを利用して履歴管理をするために外部ファイルとしてPcellのスク립トを管理するケースがあります。

このような場合、definePCell() 関数を利用することで外部のPcellスク립トファイルをコンパイルすることができます。

XIコマンドライン上にて**definePCell("external file name")**を実行します。

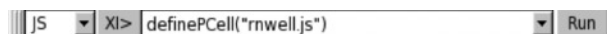


図3 XIコマンドライン上のdefinePCell

Expert 4.8.17.R以降では、definePCell()を利用して複数のPcellを一つのファイルで一括してコンパイル可能です。

例)

“compile.js”の記述

```
definePCell("rnewell.js");
```

```
definePCell("rnpoly.js");
```

```
definePCell("rppoly.js");
```

XIコマンドラインでの実行 **run("compile.js")**

JavaScript Pcellの例

以下のURLのシルバコのリソースセンターからアナログデモ用PDKとして、コールバック関数を用いたPcellのサンプルがいくつかダウンロード可能です。

<https://dynamic.silvaco.com/dynamicweb/silen/downloadFiles/nonProprietaryPDKs/13727-ppdk-silvaco-analog-demo-1-1-4-R.ssu>

これらのPcellはJavaScript Pcellを知る上で非常によいサンプルになっています。

まとめ

JavaScript Pcellは全てのユーザにとってとても書きやすいものです。JavaScriptは占有的な言語ではないため、WEB上から様々な情報が入手可能です。そのため複雑なマニュアルを調べる必要はありません。さらに作成を始める際にいつでもサンプルのPcellがダウンロード可能です。

またExpertのJavaScriptはPcellだけにとどまらず、ユーザによるレイアウト機能のカスタマイズ言語としてもその優位性を発揮します。

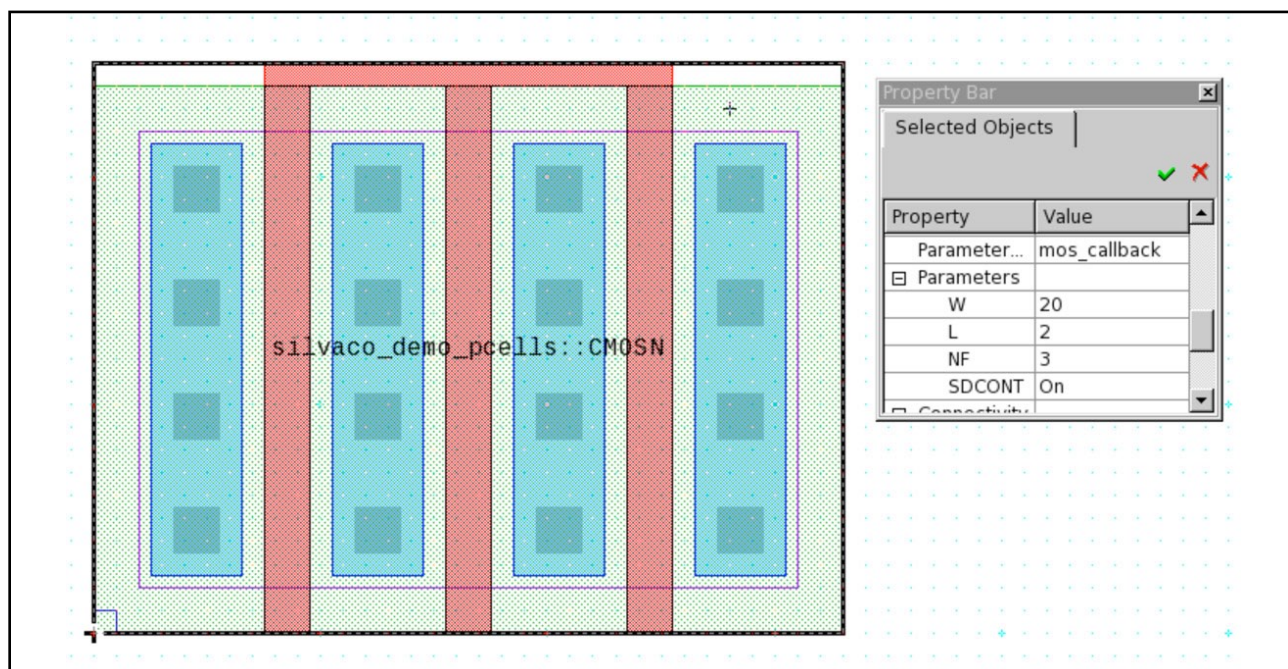


図4 デモ用PDKに含まれるPcell