

ScoutCheckerBat 使い方

setupも辞めました。
ダウンロードしたらインストール手順も無くご使用頂けます。
パスを通す必要もありません。
中にはマニュアルもサンプルプロジェクトも入っています。



Ver0.3.1.0 以降の話です。

1 ダウンロード

以下でBatタイプの最新版をダウンロード
<http://download.scoutlabo.com/top/scoutchecker/>

Batファイルタイプは以下の違いがあります。
ScoutChecker「Bat」_VerX.X.X.X

zipファイルを解凍してご使用ください。

2 ディレクトリ構造

```
Top
├ doc
│   └ マニュアルを格納
├ lib
│   └ ライブラリファイル（テキスト）を格納
├ projects
│   └ sample1
│       └ サンプルプロジェクト
│           └ Sample.bat : サンプルプロジェクトbatファイル
│               └ (backup)
│                   └ バックアップファイル格納庫。各モジュールexe間のデータやりとりに使います。
│                       └ InputData
│                           └ サンプルプロジェクトの回路ファイル等格納庫。
│                               └ (Result)
│                                   └ 結果出力ディレクトリ。
│                                       └ (Data)
│                                           └ 結果ファイル格納庫。
│                                               └ (images)
│                                                   └ 結果ファイルが参照するbmpファイル格納庫。
├ ユーザープロジェクト
├ ユーザープロジェクト
└ ScoutCheckerBat
    └ 本体格納庫。exeファイルの集合体。
        └ bin
            └ 各モジュールexe格納庫。
                └ files
                    └ CSS
                        └ CSSファイル格納庫。
                            └ images_org
                                └ bmpファイル格納庫。
```

3 サンプルプロジェクト起動

Sample.bat : サンプルプロジェクトbatファイル をダブルクリックしてください。

結果ディレクトリが作成されてHTMLファイルが生成されます。

ブラウザで中を確認してください。



必要ファイル準備

ファイル種類

ScoutCheckerBatで回路図チェックを行う
対象回路に関するファイルの準備について説明します。

- Netリストファイル
 - ・ 回路図CADが吐き出すNetリストファイル。
 - ・ このファイルを使ってパターン設計などが行われる。
 - ・ テキストファイル。
 - ・ フォーマットはtelesis。

SC_N2UC.exeの引数でこのNetリストファイルを指定してexeを起動することで、UA／CAを生成します。

- BOMファイル
 - ・ 部品表ファイル。
 - ・ csvファイル。

SC_B2U.exeの引数でこのBomファイルを指定してexeを起動することで、UAに情報追加されます。

- Etcファイル
 - ・ 追加情報ファイル。
 - ・ 回路の周辺情報を追加するファイル。
 - ・ ユーザーが作成するテキストファイル。
 - ・ 基板間接続情報などもここに属する。
 - ・ ライブラリもこのフォーマットを用います。

SC_E2C.exe／SC_E2U.exeの引数でこのEtcファイルを指定してexeを起動することで、CAやUAに情報追加されます。

例

- ・ 外部からの電源入力情報。
- ・ 回路内の端子間スルー設定情報。
- ・ 各デバイスのpinアサイン情報。

- UOファイル
 - ・ UOはUserOutputの略。
 - ・ ユーザー独自の結果出力ファイルのための設定を保存している。
 - ・ テキストファイル。

SC_OutTXTForUser.exeの引数でこのUOファイルを指定してexeを起動することで、
CAやUA、バースリスト(PL)からUOファイルにしたがってユーザー独自ファイルを出力します。

● 接続追加 行タイプ

- Unit間接続の追加。CAに追加されます。
- この接続に方向性はありません。両方から素通りが可能になります。
- 全く同じConが存在していても、追加されます。現状は仕様。

	行タイプ		Unit1						Unit2						行末	説明
			Info1						Info1							
[&]	Act	キーリ	:	値	;		Act	キーリ	:	値	;			
例1	[&]	KeyA	:	MAIN.IC1	;		KeyB	:	MAIN.IC1.1	;			普通の接続。REF(MAIN.IC1)とPIN(MAIN.IC1.1)の接続例。	
例2	[&]	KeyA	:	MAIN.IC1.1	;		KeyB	:	MAIN.#0001	;			普通の接続。PIN(MAIN.IC1.1)とNET(MAIN.#0001)の接続例。	
例3	[&]	KeyA	:	MAIN.CN1.1	;		KeyB	:	IO.CN2.1	;			基板間接続。PIN(MAIN.CN1.1)とPIN(IO.CN2.1)の接続例。基板間接続時に使用する。	
例4	[&]	KeyA	:	MAIN.D1.CATHODE	;		KeyB	:	MAIN.D1.ANODE	;			スルー接続。PIN(MAIN.D1.CATHODE)とPIN(MAIN.D1.ANODE)の接続例。電源ラインにダイオードがシリアルに入っている場合、電源アサインをその先に到達させるために使用する。	

行タイプ	説明
[&]	Unit1とUnit2が繋がっていることを定義する。方向性無し。

使用キーリ

キーリ	値	説明
KeyA/KeyB	Keyを入れる	KeyAとKeyBは等値に扱われます。

【基板間接続コネクタアサイン】

- 基板同士のコネクタ接続をPIN同士を繋ぐアサインで実現しています。本来であればREFのコネクタ同士が間に挟まるのですが。扱いにくいのでPIN同士。
- Result表示では、間にコネクタのシンボルが自動で入ります。

例:CMOS.CN3.1とMAIN.CN2.1を接続

[&]KeyA:CMOS.CN3.1,KeyB:MAIN.CN1.1,;

【スルーアサイン】

- ダイオードやEMIフィルタは3pin以上のモノもあるためバスをスルーさせるためここでピン（端子）間にスルー設定をアサインします。
- ダイオードの端子名はネットリスト内の名前を書いてください。

例1:CMOS.D3.AとMAIN.D3.Cを接続

[&]KeyA:CMOS.D3.A,KeyB:CMOS.D3.C,;

例2:IO.D1.CATHODEとIO.D1.ANODEを接続

[&]KeyA:IO.D1.CATHODE,KeyB:IO.D1.ANODE,;

● パラメータ追加 行タイプ

UAの個々のUnitに対して該当条件を満たした場合パラメータ追加。UAの子供のUnitに情報が付与されます。

行タイプ	Unit1															...			Unit2					改行	
	Info1															...			Info1						
	[]	Act	カテゴリ	:	値	,	Act	カテゴリ	:	値	,	...	Act	カテゴリ	:	値	,	...	Act	カテゴリ	:	値	,	
例	[>		TYPE	:	Pin	,	<AST>	MODEL	:	EP3C*484*	,	...		PIN	:	1	,	...		IOTYPE	:	In	,	

行タイプ	説明
[>]	Unit1の情報と合致するUnitをUAから検索しUnit2を付与

Unit1 : 該当条件

- Unit1の中の全Infoと合致するUnitが情報追加対象になります。
- 該当Unitが複数存在する場合、全部に追加されます。

Unit2 : 追加情報

- 紙面の関係上上記例ではInfoが1個ですが、複数Info並べて書くことも可能です。
- 該当Unitが既に同じカテゴリのInfoを持っていた場合、値を上書きします。

使用カテゴリ カテゴリは他にもたくさんあります。ここでは使用しそうな一部を載せています。

カテゴリ	値	説明
KEY	基板シンボル、配置番号、ピン（端子）番号	回路全体の中でUnitを特定するための名前
cSIG	POW/SIG	この端子の正極SIGタイプを入れる。
clOGIC	L/H	この端子の正極ロジックを入れる。
cVOLT	数字V	この端子の正極電圧を入れる。
jIO	I/O/IO	実際の所のIOタイプを入れる。
jSIG	POW/SIG	実際の所のSIGタイプを入れる。
jLOGIC	L/H	実際の所のロジックを入れる。
jVOLT	数字V	実際の所の電圧を入れる。

【内部電源アサイン】

ScoutChecker?が回路内の電源ICを認識出来なかった場合(未対応の場合)、手動でそのデバイスの出力端子に電源アサインを付与して電源バスを実現します。
(電源バスは回路内全て実現しないとチェックは有効な結果を吐きません)

例:GigE.U6.2に3.7Vをアサイン

[>]KEY:GigE.U6.2,;IO:Out,SIG:POW,LOGIC:H,VOLT:3.7V,jIO:Out,jSIG:POW,jLOGIC:H,jVOLT:3.7V,;

【外部電源アサイン（必須）】

この回路群に外部から供給する電源をアサインします。

例:IO.CN2.5にGNDをアサイン

[>]KEY:IO.CN2.5,;IO:Out,SIG:POW,LOGIC:L,VOLT:GND,jIO:Out,jSIG:POW,jLOGIC:L,jVOLT:GND,;

例:IO.CN2.10に電源 1 2 Vをアサイン

[>]KEY:IO.CN2.10,;IO:Out,SIG:POW,LOGIC:H,VOLT:12V,jIO:Out,jSIG:POW,jLOGIC:H,jVOLT:12V,;

● Act欄

Cat/Valだけでは、パラメータを扱う上で不便が出てきたので追加。
意味合いとしては動詞や前置詞的な使い方を予定。

<AST>

Act欄に<AST>を入れるとVALUE部分に*が使える。
これは該当条件部分に限る。

Aster(*)由来で決めた。

現状使用可能なのはパラメータ追加 行タイプの該当条件Unitのみ。
Info内のみで有効。

使用上のルール

- ・ *と書かれた部分にはどんな文字が入っても合致と判断する。
- ・ 文字数を問わない。0も可。
- ・ 数字/文字を問わない。スペースも可。
- ・ 文字列最初の*の前や、文字列最後の*の後などは厳密に比較を行う。
- ・ 以下の場合EP3Cまでは一字一句違うと合致にならない。
[>]<AST>MODEL:EP3C*484*,TYPE:Pin,PIN:1, . . .

【例1】 例えばAlteraのサイクロン3のFPGAでこう書くと484pinタイプ共通のライブラリを作ることが出来る。

[>]<AST>MODEL:EP3C*484*,TYPE:Pin,PIN:1, . . .

<NAST>

Act欄に<NAST>を入れるとVALUE部分に*が使える。この場合は*一つで1文字のみ。
これは該当条件部分に限る。

Number of aster(*) 由来で決めた。

Info内のみで有効。

使用上のルール

- ・ *と書かれた部分にはどんな文字が入っても合致と判断する。ただし1文字のみ
- ・ 数字/文字を問わない。スペースも可。

【例1】 例えばなんだか分からんが6文字目から484の文字が入った型番のモノに情報を追加出来る。

[>]<AST>MODEL:*****484*,TYPE:Pin,PIN:1, . . .

Batファイルの作成

概要

ScoutCheckerBatのモジュール群のexeを逐次実行するBatファイルを作って回路図の検査を行う。

Ver0.2ではGUIがありましたが、OSバージョンアップと共に見直す必要が出てくるので、OSバージョン依存ないようにBatファイルにしました。

Batファイルとは

パッチファイルを読む。

中身はテキストファイル。

拡張子を.batするとBatファイルになる。

アイコンをダブルクリックするだけで、中に書かれた順にコマンドを実行してくれる便利なファイル。

Windows限定。

フォーマット

Batファイルのルールに従う。

Batファイルに関する情報は世の中にたくさんあるのでここでは割愛。

モジュール[半角スペース]第1引数[半角スペース]第2引数([半角スペース]第3引数([半角スペース]第4引数([半角スペース]第5引数[半角スペース])))

半角スペースはタブに置き換えてもOKなようです。

【例】

```
SC_B2U.exe %BackupPath%backup_00.txt %BackupPath%backup_00.txt InputData\SAMPLE.csv TOP "0 0 3 4 5 6"
```

各モジュールの仕様

モジュール	引数番号					説明
	1	2	3	4	5	
SC_N2UC.exe			ファイル名 (NetWork)	文字列 (Symbol)		NetファイルからBackupファイルのUA/CAを探索。情報を生成。Symbol：回路群内の基板を特定する名前
SC_B2U.exe			ファイル名 (Bom)	文字列 (Symbol)	文字列 (Format)	BomファイルからBackupファイルのUAにデータを入れ込む。配置番号で判断してMODEL/MAKER/IMP(負数)を該当Unitに付与。Format：ファイル内の7桁目指定
SC_E2C.exe			ファイル名 (Etc)			EtcファイルからBackupファイルのCAにデータを入れ込む。Conを追加。
SC_E2U.exe			ファイル名 (Etc)			EtcファイルからBackupファイルのUAにデータを入れ込む。該当Unitに情報を追加。
SC_AddInfoRef2Pin.exe						UAのRefTypeUnitの情報をPinTypeUnitにコピーする。(BomFile情報はRefTypeUnitにしか追加しないので)
SC_PinCountMax2Unit.exe						UAのRefTypeUnitのPin数をカウントしてRefTypeUnitに設定する。(これがないとDB、HTMLで新しいデバイスは何ピンから分らない。そのPinSearchの3pin以上ステップが効かない。)
SC_SearchNet2OnePin2Net.exe						Netを全部見て、Netに対して接続が1個だけの場合、そのNetにNCをアサインする。
SC_PathSearchGND.exe						Net探索。GNDアサインのある端子(LOGIC-H)を基点に探索。探索先にGNDアサイン。
SC_PathSearchVCC.exe						Net探索。電源アサインのある端子(LOGIC-H)を基点に探索。探索先にVCCアサイン。
SC_PathSearchSig.exe						Net探索。未アサイン端子を基点に探索。探索先にSigアサイン。
SC.PinPin2PinDevPin4PathList.exe						PinPin接続時にDevKeyのUnitを挿入する。
SC_CompressPath.exe						間接パスを除外。
SC_HyokaObPow.exe						Object単位で電源の判定。
SC_OutTXTNetList.exe			ファイル名 (result)			TypeがNetのUnitについてHTMLファイル出力。
SC_OutTXTPinList.exe			ファイル名 (result)			TypeがPinのUnitについてHTMLファイル出力。
SC_OutTXTRefList.exe			ファイル名 (result)			TypeがRefのUnitについてHTMLファイル出力。

現状、環境に合わせて変更が必要な箇所

☐ : 変更を推奨しない箇所

Bomファイルの引数

SC_B2U.exeモジュールの第5引数のFormatとは
Bomファイル内のデータの並び情報です。

並びのルールは以下になります。

1文字目	2文字目	3文字目	4文字目	5文字目	6文字目
スタート行	半角スペース	配置番号の列	半角スペース	VALUEの列	半角スペース
				MODEL(型番)の列	半角スペース
				メーカーの列	半角スペース
					員数の列

注意事項

同じ列を指定しても不都合はありません。同じ情報が複数箇所にアサインされるだけです。
ダブルクォーテーションで括弧に入れてください。
0スタートです。

バックアップファイル

ScoutCheckerBatで対象にしているプロジェクト情報の全てを格納しています。
基本的に全部同じファイルを指定して上書きで構いません。

全モジュールが入力バックアップファイルの中身を踏襲するので
開始時のファイルだけは空っぽファイルを指定してスタートした方がいいです。

The screenshot shows a Windows command prompt window with the title "Sample.bat". The window displays the contents of a batch file. Red arrows point to various sections of the code, each with a Japanese annotation explaining its function.

Annotations and corresponding code sections:

- 変数宣言 (Variable Declaration):** Points to lines 1-10, which set various paths like `bin`, `css`, `original`, `img`, `library`, `lib`, `backup`, `src`, and `dst`.
- カレントディレクトリ設定 (Current Directory Setting):** Points to line 21, which sets the current directory to the project directory.
- バックアップディレクトリ作成 (Backup Directory Creation):** Points to lines 22-24, which create the backup directory and set the backup path.
- バックアップファイル (開始空っぽファイル) 作成 (Backup File (Start empty file) Creation):** Points to lines 25-27, which create the backup file.
- Netリストファイル取り込み (Net List File Inclusion):** Points to line 32, which copies the net list file.
- Bomファイル取り込み (Bom File Inclusion):** Points to line 36, which copies the bom file.
- 外部コネクタ情報取り込み (External Connector Information Inclusion):** Points to lines 40-42, which copy the external connector information file.
- ユーザー設定ライブラリ (テキストファイル) の取り込み (Inclusion of User Setting Library (Text File)):** Points to lines 44-46, which copy the user setting library file.
- AddInfoRef2Pinを間に挟むのは Refに追加した情報をPinにも展開しないと Pinに情報を追加出来ないため (Adding AddInfoRef2Pin between Ref and Pin is necessary because if the information added to Ref is not expanded to Pin, the information cannot be added to Pin):** Points to lines 48-50, which copy the AddInfoRef2Pin file.
- この辺は触らない (Do not touch this area):** Points to lines 52-54, which copy the AddInfoRef2Pin file.

The batch file content is as follows:

```

1 rem ##### For Project #####
2 rem [bin(exe) path]
3 set bin=..%ScoutCheckerBatBin%
4
5 rem [css path]
6 set css=..%ScoutCheckerBatFilesYoss%
7
8 rem [original] images path
9 set img=..%ScoutCheckerBatFilesImagesOrg%
10
11 rem [library path]
12 set lib=..%lib%
13
14 rem [backup path]
15 set backup=..%lib%
16 rem [backup path + file]
17 set backup=..%lib%
18 rem [backup path + file]
19 set dst=..%lib%
20
21 rem ##### For Project #####
22 set ResultPath=Result
23 set ResultPath=Data
24 set ResultPath=Images
25
26 cd ..
27
28 mkdir %dst%
29
30 echo > %dst%
31
32
33 %bin% %css% %img% %lib% %backup% %dst%
34 %bin% %css% %img% %lib% %backup% %dst%
35
36 %bin% %css% %img% %lib% %backup% %dst%
37 %bin% %css% %img% %lib% %backup% %dst%
38
39 %bin% %css% %img% %lib% %backup% %dst%
40 %bin% %css% %img% %lib% %backup% %dst%
41 %bin% %css% %img% %lib% %backup% %dst%
42
43 %bin% %css% %img% %lib% %backup% %dst%
44 %bin% %css% %img% %lib% %backup% %dst%
45 %bin% %css% %img% %lib% %backup% %dst%
46 %bin% %css% %img% %lib% %backup% %dst%
47 %bin% %css% %img% %lib% %backup% %dst%
48 %bin% %css% %img% %lib% %backup% %dst%
49 %bin% %css% %img% %lib% %backup% %dst%
50 %bin% %css% %img% %lib% %backup% %dst%
51 %bin% %css% %img% %lib% %backup% %dst%
52 %bin% %css% %img% %lib% %backup% %dst%
53
54 %bin% %css% %img% %lib% %backup% %dst%
55 %bin% %css% %img% %lib% %backup% %dst%
56 %bin% %css% %img% %lib% %backup% %dst%
57 %bin% %css% %img% %lib% %backup% %dst%
58 %bin% %css% %img% %lib% %backup% %dst%
59 %bin% %css% %img% %lib% %backup% %dst%
60 %bin% %css% %img% %lib% %backup% %dst%
61 %bin% %css% %img% %lib% %backup% %dst%
62 %bin% %css% %img% %lib% %backup% %dst%
63 %bin% %css% %img% %lib% %backup% %dst%
64
65 mkdir %dst%
66 mkdir %dst%
67 mkdir %dst%
68
69 copy %css% %dst%
70
71 copy %css% %dst%
72
73 %bin% %css% %img% %lib% %backup% %dst%
74 %bin% %css% %img% %lib% %backup% %dst%
75 %bin% %css% %img% %lib% %backup% %dst%
76
77 set /p USER_INPUT_STA=":"
78

```


基板間接続情報作成

複数基板構成の回路図群について
基板間接続情報が必要になるので

ここでは基板間接続情報ファイルの作成手順を示す。

基板間接続情報ファイル

ScoutCheckerBatではEtcファイルの一部として扱う。

Etcファイルの接続追加 行タイプ

例を上げると

```
[&]KeyA:CMOS.CN3.1,KeyB:MAIN.CN1.1,;  
[&]KeyA:CMOS.CN3.2,KeyB:MAIN.CN1.2,;  
. . .  
[&]KeyA:CMOS.CN3.20,KeyB:MAIN.CN1.20,;
```

このようなモノを接続コネクタの端子本数分作って
テキストファイルにする。

あと、コネクタは基本的に終端として扱っているので
終端しないことを設定するために、接続するコネクタ２個にそれぞれ以下の設定を入れる。

```
[>]KEY:CMOS.CN3,TermEach:NotTerm,;  
[>]KEY:MAIN.CN1,TermEach:NotTerm,;
```

ライブラリファイル作成

ここではライブラリファイルの作り方を説明します。

ScoutCheckerBatでのライブラリファイルは

- ・特定のデバイス固有の情報を入れたファイル。
- ・デバイスの型番などで該当するUnitに情報を付与するための情報。
- ・テキストファイルである。
- ・基本的にユーザーが作るもの。
- ・「必要ファイル準備」のファイル種類で言うとEtcファイルに該当します。

チェックのルールに当たる部分でもあるので

ここは設計者の管理の元生成されるべきという考え方です。

● 作成手順概略

テキストファイルを作る。

↓

任意のディレクトリに入れる。

(ただのEtcファイルなのでユーザーが置き場所を変えることも出来る)

Top

└─ lib

↓

batファイルでSC_E2U.exeからパスで指定。

● 作成ルール

実はただのEtcファイルなので、ルールはEtcファイルに従います。

要は、Alteraのサイクロン3なら型番の頭が「EP3C40U484」なら1番ピンはIOピン。

などのように、「必要ファイル準備 → パラメータ追加 行タイプ」を

応用して実現します。

● ライブラリファイル例 説明

- * ライブラリファイル (Cap_murata.txt) を開いてください。

Top

└─ lib

Cap_murata.txt

これはムラタのセラコン情報を付与するライブラリファイルです。

データシートを元にムラタの型番ルールに則って、パッケージ、耐圧、容量値を付与しています。

- ・ [>]TYPE:Ref;RTYPE:C;MAKER:MURATA,<NAST>MODEL:***02*;;PACKAGE:0402;;
 - ・ [>]はEtcファイルの「パラメータ追加 行タイプ」ルール。
 - ・ TYPEはUnitの種類としてRef/Pin/Netがあり、そのどれかを指定している。
 - ・ RTYPEはデバイスの種類、Cならセラコンを指定している。
 - ・ MAKERでムラタを指定して、ムラタの型番に絞っている。
(自分のBOMがムラタを村田やMURATAと表記している場合はそれに合わせるか、表記ゆれ対策を行ってください)
 - ・ <NAST>は*の使い方を指定している。
「必要ファイル準備 → Act欄」参照
この場合、4 - 5文字目が02の場合ヒットします。
 - ・ MODELは型番を指す。BOMファイルの情報に基づきます。
 - ・ PACKAGEは付与情報で、ユーザーが定義するモノです。

ここでは、PACKAGE情報を付与していますが

PACKAGEという項目自体、ここで初めて定義されるもので

ScoutCheckerの予約語などでは無いです。

使い方としては

ここで自分の考える情報を付与して

出力ファイルでこの項目を指定して

比較したり、目視チェックに使用します。

具体的な例としてはAlteraのサイクロン3の該当ピンに

BANK番号を付与して、後からBANKごとにまとめて見て

一つのBANK内で異なる電源が供給されてないか

などを確認したりします。

(同一BANK内で違う電源電圧を入れてはいけないルールをチェック)

ちなみに、出力ファイルもユーザー独自に作ることができます。

- * ライブラリファイル (EP3C10E144_pin.txt) を開いてください。

```
Top
└─ lib
    EP3C10E144_pin.txt
```

これはAlteraのサイクロン3 144ピンパッケージの
ピン情報をピンに付与するものです。

- ・ [>]TYPE:Ref,RTYPE:IC,<AST>MODEL:EP3C10E144*;;cPinCount:144,MAKER:Altera;;

- ・ TYPEがRefなのでIC本体のUnitを指定しています。
- ・ RTYPEがICを対象にしています。もしネットリストで配置番号の頭文字がICじゃない場合はICをするかこのライブラリファイルの修正が必要です。
- ・ <AST>は*の使い方を指定している。
「必要ファイル準備 → Act欄」参照
この場合はMODELの先頭がEP3C10E144で有る場合ヒットすることになります。
- ・ ヒットしたUnitに対してcPinCountという項目でデバイスの正解ピン数を付与しています。
- ・ ヒットしたUnitに対して、メーカー名をAlteraと入れて、メーカー名の統一を図っています。(表記ゆれ対策)

- ・ [>]TYPE:Pin,RTYPE:IC,<AST>MODEL:EP3C10E144*,PIN:5,;IOTYPE:in,cSIG:POW,NAME:VCCINT,cLOGIC:H,cVOLT:1.2V,RecommendVolt:1.15-1.25V,MaxVolt:-0.5-1.8V,BANK:B1,VREFB:VREFB1N0,;

- ・ IOTYPEはそのピンの方向を入れます。
- ・ cSIGは正解の信号タイプ。
- ・ cLOGIC/cVOLTも正解の情報を付与しています。あとで、この正解情報を元に判定を行っています。

- * 表記ゆれ対策

どうしても表記ゆれが出ます。
誰かが書いたBOMをベースとして引き継いだり
同じ人の中でも
書いた時期によっても起きる。

具体例として
コンデンサメーカー名でムラタとMURATAと村田、村田製作所など。

ここでは、それらの表記ゆれを一つに統一するためにコンバートを行う書き方を説明します。

- ・ 書き方

簡単に言ってしまうと
表記がゆれに該当したら、まとめた表記に変換するだけ。

例

```
[>]MAKER:ムラタ;;MAKER:MURATA,;
```

サンプル回路にも入れましたが
BOMの一部に表記ゆれを作ってあり
それがコンバートされて
問題無く出力されていることが分かります。

参照

```
SAMPLE_ExtCN.txt
[>]MAKER:ムラタ;;MAKER:MURATA,;
```

```
SAMPLE.csv
3行目
C3,C,3,0,1U,GRM155B31E104KA87D,ムラタ,1
```

他のセラコンのメーカー名はMURATAになっていますが
C3だけカタカナになっており、この対策がSAMPLE_ExtCN.txtに入れてあります。

このコンバートはScoutChecker内でのコンバートなので
大本のBOMを修正するものではありません。

ライブラリの一部に入れてもいいですし、プロジェクトごとに設定してもよろしいかと。

配置番号のRTYPEでICとUなどで表記ゆれがある場合にも使用できると思います。

ユーザー出力ファイル作成

ユーザー独自の結果出力ファイルを作成することができます。

出力フォーマット、抽出ルールなどを書いたUO (UserOutput) ファイルを作り
ユーザー出力モジュール (SC_OutTXTForUser.exe) でそれを指定することでユーザー出力ファイルが生成されます。

● 作成手順概略

UOファイルを作る。

↓

任意のディレクトリに入れる。

(ユーザーが置き場所を変えることも出来る)

Top
└─ lib
↓
batファイルでSC_OutTXTForUser.exeからパスで指定。

● UOファイル作成ルール

● UO 行タイプ

- ・ 1行でファイル内の1項目を表します。
- ・ 1項目にはアイコンリストや表などの種類があります。
- ・ 複数行書くことで、1個のファイルに情報を詰め込むことが出来ます。
- ・ Infoは順不同。

行タイプ	Unit1															Unit2															実行				
	Info1					Info2					Info3					Info1					Info2					Info3									
	Act	975*9	:	値	:	Act	975*9	:	値	:	Act	975*9	:	値	:	Act	975*9	:	値	:	Act	975*9	:	値	:	Act	975*9	:	値	:					
例	[UO]		UO_MTYPE	:	TABLE_UA	:		UO_TITLE	:	EP3C_BANK_B8	:		UO_COLFIL	:	KEYKEY PIN/PIN NAME/NAME	:		<AST>	MODEL	:	EP3C*	:		TYPE	:	Pin	:		BANK	:	B7	:	
例	[UO]		UO_MTYPE	:	IPATH_UA	:		UO_TITLE	:	EP3C_BANK_B8_ipath	:			:		:		<AST>	MODEL	:	EP3C*	:		TYPE	:	Pin	:		BANK	:	B7	:	

行タイプ	説明
[UO]	UserOutputの略。ユーザー独自の結果出力ファイル作成ルール。

Unit	
Unitタイプ	説明
Unit1	項目情報
Unit2	データ抽出フィルタ。

カテゴリ	値	説明
UO_MTYPE	テキスト	UOタイプ、項目ごとのアイコンリストや表などのタイプを指定する。
	IPATH_UA	UAに対してフィルタ(データ抽出)を掛けて アイコンバツTree にして出力する。
	TABLE_UA	UAに対してフィルタ(データ抽出)を掛けて 表 にして出力する。
UO_TITLE	テキスト	項目のタイトル。
UO_COLFIL	テキスト	項目の表の列のヘッダのリスト。半角空欄区切り。[半角縦線]で左側が結果での列タイトル。右側が抽出項目名

● U0ファイル例 説明

- * UOファイル (Alterra_EP3C_BANK.txt) を開いてください。

Top
└─ lib
 Alterra_EP3C_BANK.txt

これはAlteraのFPGAのbank別の接続と状況確認情報を出力するUOファイルです。

[UO]UO_MTYPE:TABLE_UA,UO_TITLE:EP3C_BANK_B1,UO_COLFIL:KEY|KEY
PIN|PIN NAME|NAME BANK|BANK st|LOGIC st|POW|stPOW j|SIG|結果

j|SIG j|LOGIC|結果
j|LOGIC j|VOLT|結果
j|VOLT Linked|Linked
.;MAKER:Altera,TYPE:Pin,RTYPE:IC,<AST>MODEL:EP3C*.BANK:B1.;

- ・ [UO]はこの行がUOファイルの設定ルールであることを指します。
- ・ 1行で1出力項目を表現します。
- ・ UO_MTYPEはこの表示項目がどのタイプであるかを示します。

TABLE_UA	UAの中から抽出して、テーブル(表)出力する。
IPATH_UA	UAの中から抽出して、該当Unitを左側にしたAIコンパスを表示する。
- ・ UO_TITLEは表示項目の題。
- ・ UO_COLFILEはテーブル(表)表示のヘッダ行に表示される内容を定義します。

A	
カテゴリ	テーブルに表示させる文字列 改行の 使用可。

↑ ユーザーが設定したカテゴリまたは既にあるカテゴリを指定。

Aで1列分を表現し、Aを半角空欄を挟んで並べる。

Aの一例としてここを見ると

これはjLOGICというカテゴリをこの列に表示させ、ヘッダ行には「結果（改行）jLOGIC」と表示させる。

- ・Unit2はMAKERがAlteraでタイプがPinでRTYPEがIC、MODELがEP3C*に該当しBANKがB1のものを抽出しています。
内容的に無くても同じ結果になりそうな条件もありそうですが。

8オブジェクト

TOPCH1
TOPCH2
TOPCH3
TOPCH4
TOPCH5
TOPCH6
TOPCH7
TOPCH8
TOPCH9
TOPCH10
TOPCH11
TOPCH12
TOPCH13
TOPCH14
TOPCH15
TOPCH16
TOPCH17
TOPCH18
TOPCH19
TOPCH20
TOPCH21
TOPCH22
TOPCH23
TOPCH24
TOPCH25
TOPCH26
TOPCH27
TOPCH28
TOPCH29
TOPCH30
TOPCH31
TOPCH32
TOPCH33
TOPCH34
TOPCH35
TOPCH36
TOPCH37
TOPCH38
TOPCH39
TOPCH40
TOPCH41
TOPCH42
TOPCH43
TOPCH44
TOPCH45
TOPCH46
TOPCH47
TOPCH48
TOPCH49
TOPCH50
TOPCH51
TOPCH52
TOPCH53
TOPCH54
TOPCH55
TOPCH56
TOPCH57
TOPCH58
TOPCH59
TOPCH60
TOPCH61
TOPCH62
TOPCH63
TOPCH64
TOPCH65
TOPCH66
TOPCH67
TOPCH68
TOPCH69
TOPCH70
TOPCH71
TOPCH72
TOPCH73
TOPCH74
TOPCH75
TOPCH76
TOPCH77
TOPCH78
TOPCH79
TOPCH80
TOPCH81
TOPCH82
TOPCH83
TOPCH84
TOPCH85
TOPCH86
TOPCH87
TOPCH88
TOPCH89
TOPCH90
TOPCH91
TOPCH92
TOPCH93
TOPCH94
TOPCH95
TOPCH96
TOPCH97
TOPCH98
TOPCH99
TOPCH100
TOPCH101
TOPCH102
TOPCH103
TOPCH104
TOPCH105
TOPCH106
TOPCH107
TOPCH108
TOPCH109
TOPCH110
TOPCH111
TOPCH112
TOPCH113
TOPCH114
TOPCH115
TOPCH116
TOPCH117
TOPCH118
TOPCH119
TOPCH120
TOPCH121
TOPCH122
TOPCH123
TOPCH124
TOPCH125
TOPCH126
TOPCH127
TOPCH128
TOPCH129
TOPCH130
TOPCH131
TOPCH132
TOPCH133
TOPCH134
TOPCH135
TOPCH136
TOPCH137
TOPCH138
TOPCH139
TOPCH140
TOPCH141
TOPCH142
TOPCH143
TOPCH144
TOPCH145
TOPCH146
TOPCH147
TOPCH148
TOPCH149
TOPCH150
TOPCH151
TOPCH152
TOPCH153
TOPCH154
TOPCH155
TOPCH156
TOPCH157
TOPCH158
TOPCH159
TOPCH160
TOPCH161
TOPCH162
TOPCH163
TOPCH164
TOPCH165
TOPCH166
TOPCH167
TOPCH168
TOPCH169
TOPCH170
TOPCH171
TOPCH172
TOPCH173
TOPCH174
TOPCH175
TOPCH176
TOPCH177
TOPCH178
TOPCH179
TOPCH180
TOPCH181
TOPCH182
TOPCH183
TOPCH184
TOPCH185
TOPCH186
TOPCH187
TOPCH188
TOPCH189
TOPCH190
TOPCH191
TOPCH192
TOPCH193
TOPCH194
TOPCH195
TOPCH196
TOPCH197
TOPCH198
TOPCH199
TOPCH200
TOPCH201
TOPCH202
TOPCH203
TOPCH204
TOPCH205
TOPCH206
TOPCH207
TOPCH208
TOPCH209
TOPCH210
TOPCH211
TOPCH212
TOPCH213
TOPCH214
TOPCH215
TOPCH216
TOPCH217
TOPCH218
TOPCH219
TOPCH220
TOPCH221
TOPCH222
TOPCH223
TOPCH224
TOPCH225
TOPCH226
TOPCH227
TOPCH228
TOPCH229
TOPCH230
TOPCH231
TOPCH232
TOPCH233
TOPCH234
TOPCH235
TOPCH236
TOPCH237
TOPCH238
TOPCH239
TOPCH240
TOPCH241
TOPCH242
TOPCH243
TOPCH244
TOPCH245
TOPCH246
TOPCH247
TOPCH248
TOPCH249
TOPCH250
TOPCH251
TOPCH252
TOPCH253
TOPCH254
TOPCH255
TOPCH256
TOPCH257
TOPCH258
TOPCH259
TOPCH260
TOPCH261
TOPCH262
TOPCH263
TOPCH264
TOPCH265
TOPCH266
TOPCH267
TOPCH268
TOPCH269
TOPCH270
TOPCH271
TOPCH272
TOPCH273
TOPCH274
TOPCH275
TOPCH276
TOPCH277
TOPCH278
TOPCH279
TOPCH280
TOPCH281
TOPCH282
TOPCH283
TOPCH284
TOPCH285
TOPCH286
TOPCH287
TOPCH288
TOPCH289
TOPCH290
TOPCH291
TOPCH292
TOPCH293
TOPCH294
TOPCH295
TOPCH296
TOPCH297
TOPCH298
TOPCH299
TOPCH300
TOPCH301
TOPCH302
TOPCH303
TOPCH304
TOPCH305
TOPCH306
TOPCH307
TOPCH308
TOPCH309
TOPCH310
TOPCH311
TOPCH312
TOPCH313
TOPCH314
TOPCH315
TOPCH316
TOPCH317
TOPCH318
TOPCH319
TOPCH320
TOPCH321
TOPCH322
TOPCH323
TOPCH324
TOPCH325
TOPCH326
TOPCH327
TOPCH328
TOPCH329
TOPCH330
TOPCH331
TOPCH332
TOPCH333
TOPCH334
TOPCH335
TOPCH336
TOPCH337
TOPCH338
TOPCH339
TOPCH340
TOPCH341
TOPCH342
TOPCH343
TOPCH344
TOPCH345
TOPCH346
TOPCH347
TOPCH348
TOPCH349
TOPCH350
TOPCH351
TOPCH352
TOPCH353
TOPCH354
TOPCH355
TOPCH356
TOPCH357
TOPCH358
TOPCH359
TOPCH360
TOPCH361
TOPCH362
TOPCH363
TOPCH364
TOPCH365
TOPCH366
TOPCH367
TOPCH368
TOPCH369
TOPCH370
TOPCH371
TOPCH372
TOPCH373
TOPCH374
TOPCH375
TOPCH376
TOPCH377
TOPCH378
TOPCH379
TOPCH380
TOPCH381
TOPCH382
TOPCH383
TOPCH384
TOPCH385
TOPCH386
TOPCH387
TOPCH388
TOPCH389
TOPCH390
TOPCH391
TOPCH392
TOPCH393
TOPCH394
TOPCH395
TOPCH396
TOPCH397
TOPCH398
TOPCH399
TOPCH400
TOPCH401
TOPCH402
TOPCH403
TOPCH404
TOPCH405
TOPCH406
TOPCH407
TOPCH408
TOPCH409
TOPCH410
TOPCH411
TOPCH412
TOPCH413
TOPCH414
TOPCH415
TOPCH416
TOPCH417
TOPCH418
TOPCH419
TOPCH420
TOPCH421
TOPCH422
TOPCH423
TOPCH424
TOPCH425
TOPCH426
TOPCH427
TOPCH428
TOPCH429
TOPCH430
TOPCH431
TOPCH432
TOPCH433
TOPCH434
TOPCH435
TOPCH436
TOPCH437
TOPCH438
TOPCH439
TOPCH440
TOPCH441
TOPCH442
TOPCH443
TOPCH444
TOPCH445
TOPCH446
TOPCH447
TOPCH448
TOPCH449
TOPCH450
TOPCH451
TOPCH452
TOPCH453
TOPCH454
TOPCH455
TOPCH456
TOPCH457
TOPCH458
TOPCH459
TOPCH460
TOPCH461
TOPCH462
TOPCH463
TOPCH464
TOPCH465
TOPCH466
TOPCH467
TOPCH468
TOPCH469
TOPCH470
TOPCH471
TOPCH472
TOPCH473
TOPCH474
TOPCH475
TOPCH476
TOPCH477
TOPCH478
TOPCH479
TOPCH480
TOPCH481
TOPCH482
TOPCH483
TOPCH484
TOPCH485
TOPCH486
TOPCH487
TOPCH488
TOPCH489
TOPCH490
TOPCH491
TOPCH492
TOPCH493
TOPCH494
TOPCH495
TOPCH496
TOPCH497
TOPCH498
TOPCH499
TOPCH500
TOPCH501
TOPCH502
TOPCH503
TOPCH504
TOPCH505
TOPCH506
TOPCH507
TOPCH508
TOPCH509
TOPCH510
TOPCH511
TOPCH512
TOPCH513
TOPCH514
TOPCH515
TOPCH516
TOPCH517
TOPCH518
TOPCH519
TOPCH520
TOPCH521
TOPCH522
TOPCH523
TOPCH524
TOPCH525
TOPCH526
TOPCH527
TOPCH528
TOPCH529
TOPCH530
TOPCH531
TOPCH532
TOPCH533
TOPCH534
TOPCH535
TOPCH536
TOPCH537
TOPCH538
TOPCH539
TOPCH540
TOPCH541
TOPCH542
TOPCH543
TOPCH544
TOPCH545
TOPCH546
TOPCH547
TOPCH548
TOPCH549
TOPCH550
TOPCH551
TOPCH552
TOPCH553
TOPCH554
TOPCH555
TOPCH556
TOPCH557
TOPCH558
TOPCH559
TOPCH560
TOPCH561
TOPCH562
TOPCH563
TOPCH564
TOPCH565
TOPCH566
TOPCH567
TOPCH568
TOPCH569
TOPCH570
TOPCH571
TOPCH572
TOPCH573
TOPCH574
TOPCH575
TOPCH576
TOPCH577
TOPCH578
TOPCH579
TOPCH580
TOPCH581
TOPCH582
TOPCH583
TOPCH584
TOPCH585
TOPCH586
TOPCH587
TOPCH588
TOPCH589
TOPCH590
TOPCH591
TOPCH592
TOPCH593
TOPCH594
TOPCH59

語句説明

あいうえお、アイウエオ、アルファベット順

終端

バスを終わらせること。
バスを終わらせるデバイス。

追加情報

追加情報ファイル。
ETCファイル
ユーザーが設計に基いて、必要な情報を付与するためのファイル。
テキストファイル。手動で編集ができる。

アサイン

端子やデバイス、NETなどに対して情報を付与すること。

スルー

3pin以上のデバイスでバスを通過させたい場合。スルー設定を入れます。

チェック

回路図チェック

DRC (Design rule check)
ScoutChecker?で回路の検査を行うこと。

デバイス

部品のこと。
回路上の部品全てがこれに当てはまります。
デバイス本体と端子。
Unitで言うとREF+PIN。

バス

ここではデバイス間の接続をバスと呼ぶことにします。端子から終端条件のあるデバイスまたはNETまでの接続を1つのバスとします。バスはNETを共有出来ます。
(例：VCC端子には電源ICの出力端子、バスコンなどが繋がっていますが、VCC端子-電源出力端子/VCC端子-バスコン それぞれが一つのバスになります。)

アイコンバス

アイコンで表現されたバス

BOM

BOMファイル

Bill of materialの略。部品構成表。回路図のシンボルに対して詳細情報を記載したものの。

CA

ConArrayの略。Con配列。プロジェクト内の全てのデバイス本体-ピン（端子）／ネット-ピン（端子）接続の情報が、この中に入っています。

Con

デバイス本体-端子／ネット-端子接続を個々にConという単位で扱います。(全て1:1接続です)

DEVKEY

端子が属するデバイス本体のKey。端子が自分の親デバイスを把握するために持つ。
デバイス本体のUnitも持っている。

DRC

Design Rule Checkの略。
一般的に回路図CADに付属で付いている場合が多い。
内容についてはCADによってまちまちで、具体的に何を指すかは決められていない。

ETCファイル

追加情報ファイル。
ユーザーが設計に基いて、必要な情報を付与するためのファイル。
テキストファイル。手動で編集ができる。

Key

KEY

このプロジェクト内でUnit(デバイス本体/ピン（端子）/NET)を特定するために付ける名前。
配置番号のようなもの。

フォーマットは

基板シンボル.部品配置番号.ピン（端子）番号

例：CMOS.CN3.1

NET

NETリストを最小単位Unitにバラした時の基板のパターン部分。

NETリスト

ネットリスト。回路図CADが吐き出す結線情報。

PIN

NETリストを最小単位Unitにバラした時の端子の部分。デバイスの端子。デバイスは必ず端子でNETに接続されていることを前提にしています。

REF

NETリストを最小単位Unitにバラした時のデバイス（部品）本体の部分。
回路上の部品全ての本体部分(端子除く)がこれに当てはまります。

RTYPE

部品タイプ。REFの種類。部品配置番号の頭のアルファベット。このツールではこのアルファベットでデバイスの種類を把握しています。

RTYPE	定義
R	抵抗
C	コンデンサ
L	インダクタ
F	ヒューズ
Z	EMIフィルタ、パリスタ
CN	コネクタ
X	発振器、発信子
D	ダイオード、LED
Q	トランジスタ
IC	半導体
U	自分ではあまり使わないですが、使う人が多いので半導体に近い何かとして定義

SYMBOL(基板シンボル(テキスト))

基板を特定するシンボル名。
Batファイルで定義します。
主に基板間の配置番号の被り対策に使います。

UA

UnitArrayの略。Unit配列。プロジェクト内の全てのUnit(デバイス本体／ネット／端子) 情報が、この中に入っています。

Unit

REF/NET/PINの単位、総称。
NETリストをこの最小単位Unit(REF/NET/PIN)に分解した時の呼び名。
デバイス本体／ネット／端子を個々に(REF/NET/PIN)というUnit単位で扱います。

ライブラリ

ライブラリファイル

特定のデバイス固有の情報を付与するために準備したもの。
テキストファイル。
「4. 必要ファイル準備」のファイル種類で言うEtcファイルに該当します。
ディレクトリ構成で言うと以下の場所に保管されている。（Etcファイルなのでユーザー編集可能）
Top
└─ lib
基本的にユーザーが作るもの。

UO

UserOutputの略。
ユーザー独自ファイル出力機能。