

## Tutoriel IO Checker



Même si HDL Works BV a testé le logiciel et révisé la documentation, HDL Works BV n'offre aucune garantie ou déclaration, expresse ou implicite, à l'égard de ce logiciel et de sa documentation, qualité, performance, qualité marchande ou d'adéquation à un usage particulier.

Aucune partie de ce document ne peut être copié ou reproduit sous quelque forme ou par quelque moyen sans l'autorisation écrite préalable de HDL Works BV. Les informations contenues dans ce document sont sujettes à modification sans préavis et ne doivent pas être interprétées comme un engagement de HDL Works BV. HDL Works BV n'assume aucune responsabilité pour les erreurs qui pourraient apparaître dans ce document. HDL Travaux BV n'offre aucune garantie d'aucune sorte à l'égard de ce matériel, y compris, mais sans s'y limiter, les garanties implicites de qualité marchande et d'adéquation à un usage particulier. Le logiciel décrit dans ce document est droits d'auteur par HDL Works BV, Ede, aux Pays-Bas.

Le logiciel décrit dans ce document est fourni sous licence et ne peut être utilisé ou copié que conformément aux termes d'un accord de licence du logiciel.

HDL Works BV reconnaît des marques commerciales ou des marques déposées d'autres organisations pour leurs produits et services respectifs.

Version du logiciel: IO Checker 4.0 Rev1.

## Table des matières

---



<b>1- Introduction</b>	<b>6</b>
<b>2- Démarrage de l'application</b>	<b>7</b>
<b>3- Mise en place des données</b>	<b>9</b>
<b>4- Fenêtre principale de IO checker</b>	<b>13</b>
<b>5- Inspection des premiers résultats</b>	<b>15</b>
<b>6- Se concentrer sur les issues potentielles</b>	<b>18</b>
<b>7- Ajouter une règle d'utilisateur</b>	<b>20</b>
<b>8- Incompatibilités utilisateur acceptées</b>	<b>22</b>
<b>9- Vérifications voltages et masses</b>	<b>24</b>
<b>10- Mise à jour du fichier de contraintes</b>	<b>26</b>
<b>11- Générateur HTML</b>	<b>28</b>
<b>12- Sauvegarder le projet</b>	<b>29</b>
<b>13- Informations contact</b>	<b>30</b>

## Introduction

Ce tutoriel vous montre comment vous pouvez utiliser IO Checker pour vérifier que le FPGA et PCB sont connectés correctement. Dans ce tutoriel, nous utilisons un Stratix-2 dispositif Altera et un package Mentor cross reference netlist mais les étapes décrites dans ce tutoriel sont indépendantes du fournisseur de FPGA et système de PCB utilisés.

### Sur Windows:

Les exemples IO Checker sont installés dans 'ProgramData\HDL Works\iochecker30rev...\iochecker\examples' répertoire sur Windows Vista / 7/8, lorsque le programme lui-même est installé dans le répertoire 'ProgramFiles'. Lorsque IO Checker n'est pas installé dans le répertoire 'ProgramFiles' les exemples sont à l'intérieur du répertoire d'installation 'iochecker\examples'.

### Sur Linux:

Sous Linux les exemples sont à l'intérieur du répertoire d'installation 'iochecker/examples'.

Nous allons d'abord créer le projet en précisant les données nécessaires. Après cela, nous allons examiner les résultats de la comparaison et générer les règles et accepter les inadéquations (correctes). Enfin, nous mettrons à jour le fichier de contraintes qui peut être utilisé pour régénérer un fichier de broche.

Outre la comparaison du fichier de broche de FPGA avec la netlist de PCB, nous allons aussi montrer le contrôle de la tension et la génération HTML de IO Checker.

Toutes les actions de l'utilisateur dans ce tutoriel sont précédées par le symbole



Si vous avez des questions s'il vous plaît appelez-nous ou envoyez-nous un e-mail. Nous serons heureux de vous aider !

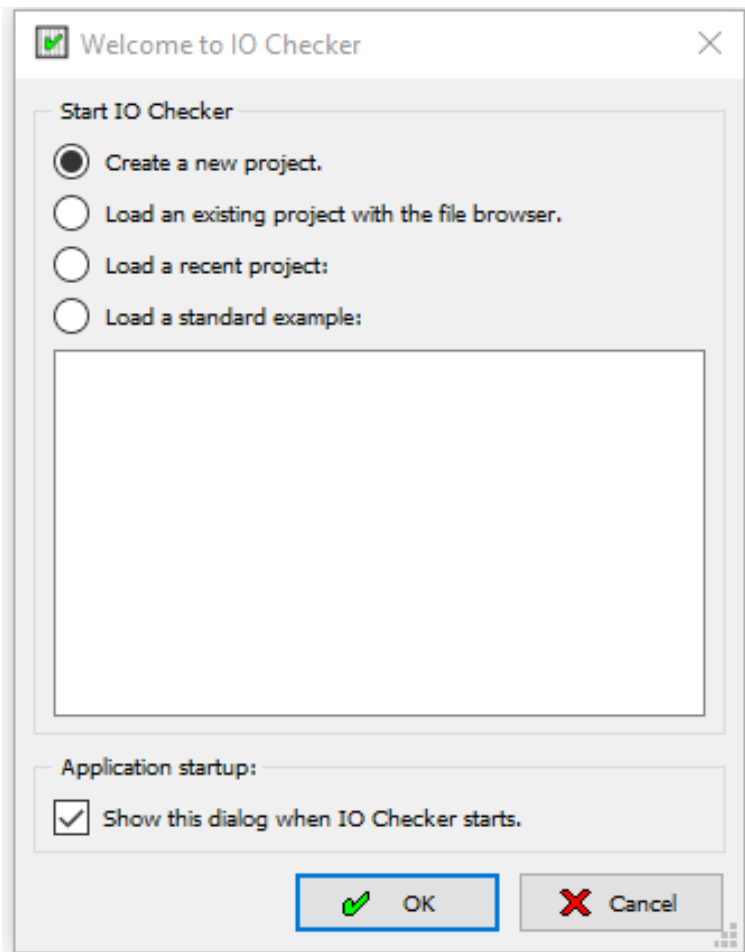
### Autres tutoriels:

Le didacticiel Création du fichier de contrainte vous montre comment générer un fichier de contraintes d'un fichier de HDL et PCB netlist.

## Démarrage de l'application

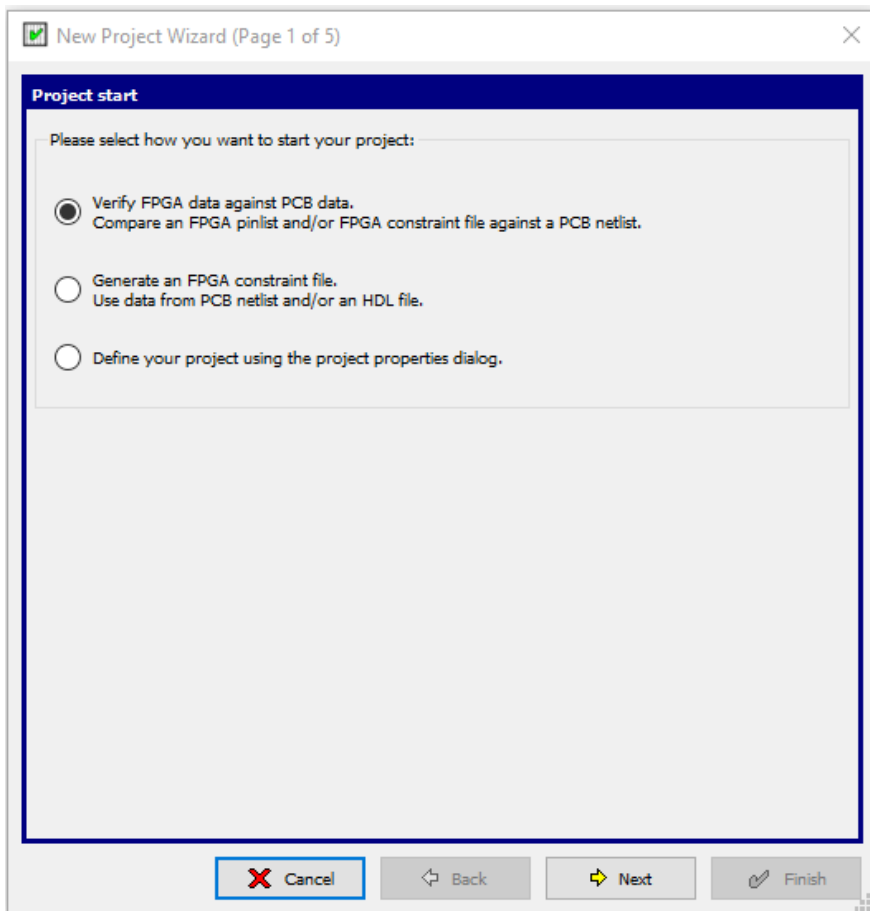


Lancer **IO Checker** et sélectionner **Créer un nouveau projet** dans la boîte de dialogue **Welcome to IO Checker** et appuyez sur le bouton OK.




*Démarrage de dialogue*

Maintenant l'**Assistant Nouveau projet** apparaît. Cet assistant vous aide en recueillant les données nécessaires à votre projet.



*Assistant de création de projet*

Vous pouvez également démarrer un nouveau projet en sélectionnant **File >> New project** ou en appuyant sur le bouton **Create a new project** 

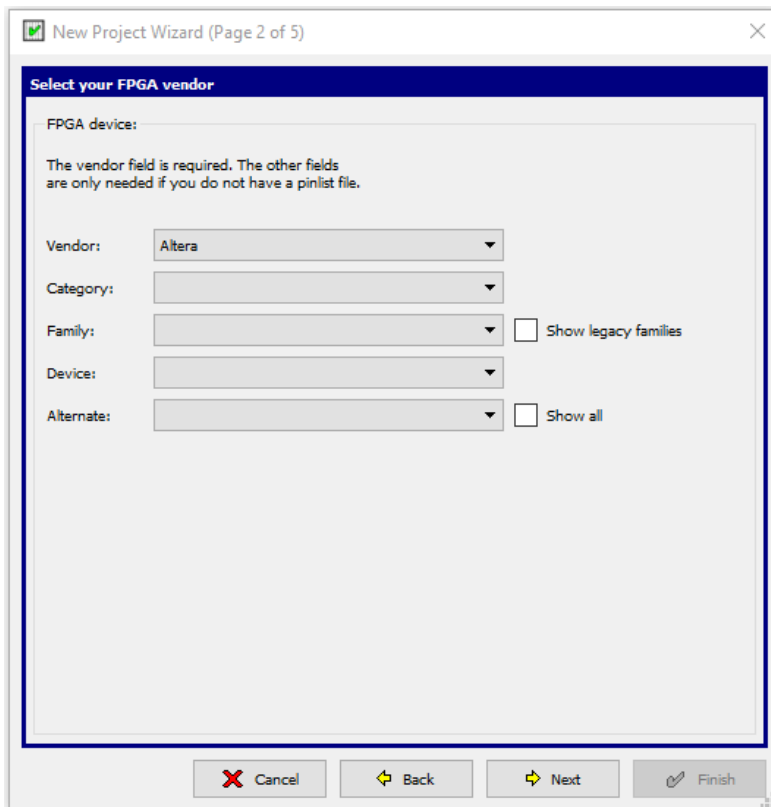


Sélectionner **Compare a FPGA Pinlist against a PCB netlist** dans l'**Assistant Nouveau projet** et appuyez sur le bouton **Suivant**.

La page du vendeur FPGA apparaît.

## Mise en place des données

Sur cette page, vous n'avez qu'à sélectionner le FPGA vendeur. Comme nous avons un fichier de broche Altera, le vendeur doit être mis à Altera. Vous pouvez laisser les entrées de la famille et de l'appareil vide, car elles peuvent être extraites du fichier de broche.




*Sélection du fournisseur de FPGA*



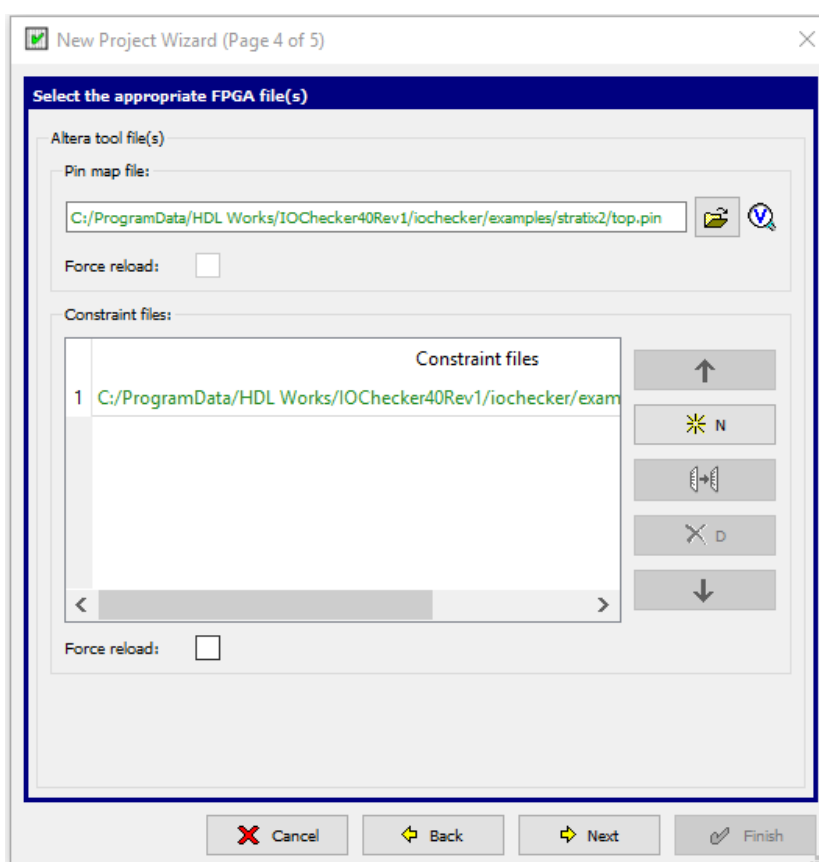
Sélectionnez **Altera** dans le champ du vendeur et appuyez sur le bouton **Suivant**

La page des fichiers FPGA apparaît. Sur cette page, vous devez sélectionner le fichier de broche. Le fichier de contrainte est facultatif.


 Appuyez sur le **bouton navigateur** derrière le champ du **fichier de carte Pin**. Dans la boîte de dialogue localiser le **fichier de broche Altera** accédez au répertoire `exemples/stratix2` dans l'installation **IO Checker** et sélectionnez le fichier `top.pin`. Puis appuyez sur le bouton **Ouvrir**.

(La meilleure façon d'obtenir le répertoire des exemples est de taper : `$INSTALLDATA` et de cliquer sur le bouton de navigation. La variable d'environnement `INSTALLDATA` est fixée par IO Checker pour pointer vers le répertoire des exemples)


Notez que le fichier de contrainte correspondante (`top.qsf`) est automatiquement sélectionné quand il est présent.



*Réglage des données FPGA*

 Appuyez sur le bouton **Suivant**. sur la dernière page de l'assistant, vous devez sélectionner le fichier de netlist PCB et désignation.

 Sélectionnez **Mentor Package Cross Reference** de la liste dans le champ du vendeur de PCB.

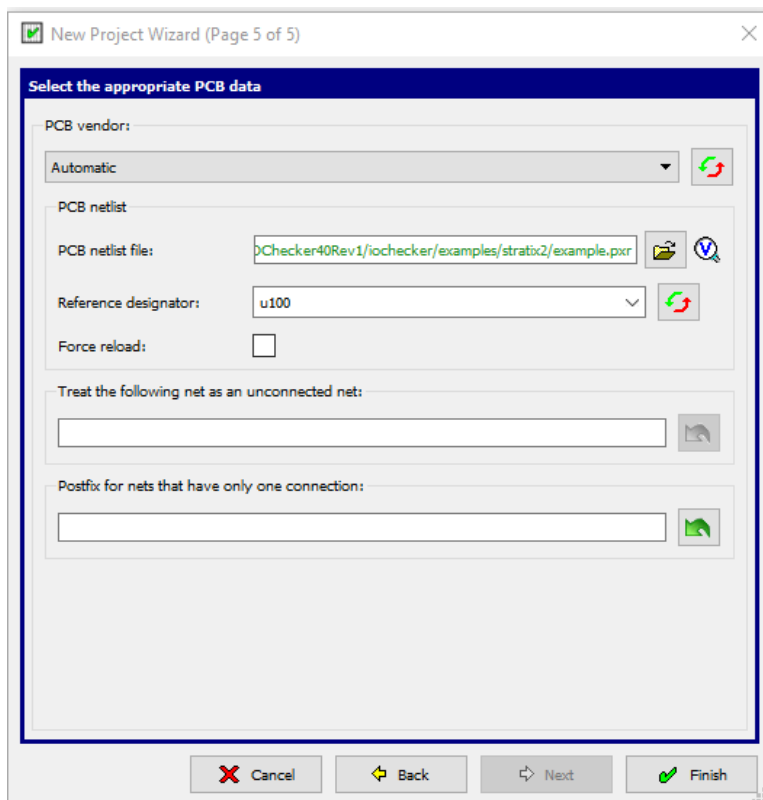
 Cliquez sur le bouton **Navigateur** derrière le champ du **fichier de netlist Mentor**. Dans **Locate Mentor Package Cross Reference** parcourir le répertoire de `exemples/stratix2` dans l'installation IO Checker et sélectionnez le fichier `example.pxr`. Puis appuyez sur le bouton **Ouvrir**.

L'**indicateur de référence** est l'identificateur qui désigne le FPGA dans la netlist et dans cet exemple, c'est **U100**. Il ne peut pas être déterminé automatiquement comme plusieurs FPGA peuvent être présents dans la liste d'interconnexions.

Vous pouvez appuyer sur le bouton Retrieve Designators afin d'obtenir une liste de tous les composants de la liste d'interconnexions avec plus de 32 broches.


(Comme ce tutoriel netlist n'est qu'un exemple, il ne contient que 2 désignateurs appropriés.)

 Remplissez **u100** comme l'**indicateur de référence**.



*Réglage des données PCB*

Maintenant, nous avons fourni toutes les informations requises et nous pouvons terminer l'assistant.

 Appuyez sur le bouton **Terminer**

Lorsque les deux pinlist et PCB netlist sont nouveaux (ou changés) la **boîte de dialogue de correspondance** est invoquée directement après le traitement de deux fichiers. Nous fermons la boîte de dialogue avec le bouton annuler pour que nous puissions inspecter les premiers résultats de **IO Checker**.

 Cliquez sur le bouton **Annuler** dans la **boîte de dialogue**.

**IO Checker** a traité les trois fichiers et rapporte les données suivantes dans la console :

```
Processing file: $INSTALLDATA/iochecker/examples/stratix2/top.pin
Extracted from pin list: primary device is 'ep2s15f672'
```

Processed: 672 pin lines  
Device: ep2s15f672

Loaded family info file 'C:/HDLWorks/IOChecker30ev1/iochecker/devices/  
altera/stratix2/stratix2.enc'

Processing file: \$INSTALLDATA/iochecker/examples/stratix2/top.qsf

Found: 12 global assignments

Found: 357 pin location assignments

Found: 356 IO standard assignments

Processing file: C:/HDLWorks/IOChecker30Rev1/iochecker/examples/stratix2/example.pxr

Parsed 1146 lines.

Collected 16 units and 384 nets

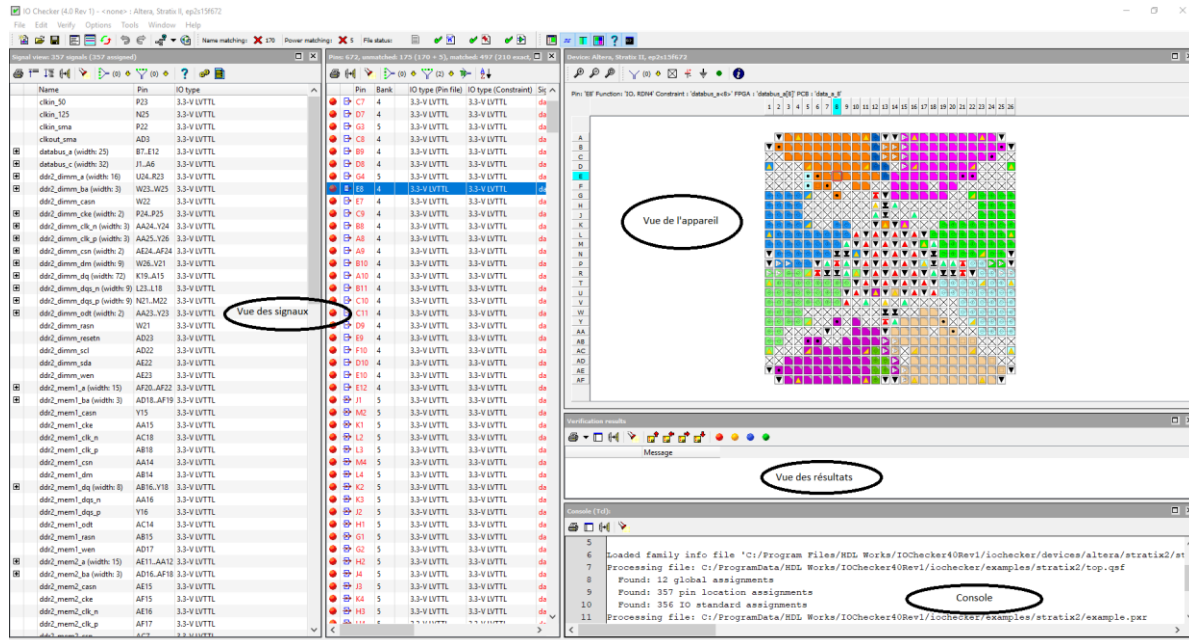
Extracted 535 ports for unit 'U100'

Retrieving device info for: Altera, Stratix II, ep2s15f672

Avant d'inspecter les premiers résultats d'IO **Checker**, nous allons d'abord expliquer les GUI d'IO **Checker**.

## Fenêtre principale de IO Checker

Après la fermeture de la **boîte de dialogue**, nous pouvons maintenant jeter un œil à la fenêtre principale de **IO Checker**.



Fenêtre principale

Les éléments suivants sont affichés dans la fenêtre principale :

- la **barre de titre** affiche le nom du programme et le projet en cours.
- la **barre de menu** répertorie les menus principaux disponibles.
- la **barre d'outils** contient des boutons faciles à utiliser. Ils sont décrits en détail dans le manuel d'utilisateur de IO Checker
- la **vue des signaux** affiche des informations de signal extraites à partir du fichier de broches et données sur les PCB
- la **vue de l'appareil** montre le package de l'appareil utilisé dans le projet.
- la **vue des résultats** montre des messages de vérification. Double-cliquer un message indique la ligne dans la vue de signal.
- la **console** affiche les messages généraux (il contient également un interprète TCL).

Quand un fichier de HDL fait partie du projet, la fenêtre principale affiche également une vue d'HDL. Comme nous n'avons pas ajouté un fichier de HDL à notre projet, ce point de vue est désactivé.

Lorsque l'on pointe la souris au-dessus du nombre de **non correspondance** dans la barre de boutons, cela montre les résultats correspondants comme indiqué ci-dessous.

Name matching: <b>X</b> 170	Power matching: <b>X</b> 5	F
97 (210 exact, 6	Device contains 672 pins, of which	
	170 names did not match	
	210 names matched exactly	
	7 device rule matches	
ype (Constraint)	134 fuzzy matches	
	0 rule matches	
	0 user matches	
	65 ground matches	
	81 power matches	
	5 power errors	
	0 power warnings	

*Résultats des correspondances*



Pointer au-dessus du nombre de **non correspondance**, comme indiqué ci-dessus.

Veuillez vérifier qu'**IO Checker** montre les mêmes résultats correspondants avant de continuer ce tutoriel. Dans le cas contraire, veuillez commencer un nouveau projet et répétez les étapes décrites dans la section précédente.

## 5 Inspection des premiers Résultats

La vue des signaux **IO Checker** affiche 12 colonnes. Les colonnes qui n'affichent qu'une icône ne montrent pas une étiquette de colonne, comme l'étiquette serait plus large que l'icône elle-même. Les colonnes indiquent (de gauche à droite) :

- Résultat de vérification : icône représentant une erreur ou avertissement (le cas échéant).
- Type de signal: icône indiquant entrée / sortie / InOut, sol / puissance, jtag, signal de commande, ou n'est pas disponible
- Pin: la broche nom / numéro de l'appareil
- Banque: le numéro de la broche
- Contrôle de tension: icône indiquant le résultat de la vérification de la tension le cas échéant
- Type IO: la norme E / S spécifié dans le fichier de liste de fixation
- Type IO (contrainte): la norme d'E / S spécifié dans le fichier de contrainte, ou définie par l'utilisateur
- Signal (fichier Pin): nom de signal FPGA dans le fichier de broche
- Signal (contrainte): nom du signal FPGA spécifié dans le fichier de contrainte, ou définie par l'utilisateur
- Type PCB : icône indiquant le type net comptabilité (pull up / down / découplage, balançant, non connecté)
- Signal (PCB): nom de signal PCB dans le fichier netlist
- Règle: le nom de la règle le cas échéant

R22		✓	<GND>	<GND>	<GND>	<GND>	GROUND	<Ground>
R23	1		3.3-V LVTTTL	3.3-V LVTTTL	ddr2_dimm_a[15]	ddr2_dimm_a<15>	ddr2_dimm_a_15	
R24	1		3.3-V LVTTTL	3.3-V LVTTTL	ddr2_dimm_a[14]	ddr2_dimm_a<14>	ddr2_dimm_a_14	correspondance floue
R25	1				ddr2_dimm_a[11]	ddr2_dimm_a<11>	ddr2_dimm_a_11	
R26	1				ddr2_dimm_a[12]	ddr2_dimm_a<12>	ddr2_dimm_a_12	<Fuzzy>
T1	6	✗	<3.3V>				P1V2 (=1.2V)	
T2	6		3.3-V LVTTTL	3.3-V LVTTTL	seg7_06	seg7_06	seg7_5	
T3	6		3.3-V LVTTTL	3.3-V LVTTTL	seg7_05	seg7_05	seg7_4	
T4	6		3.3-V LVTTTL	3.3-V LVTTTL	seg7_02		seg7_1	
T5	6		3.3-V LVTTTL	3.3-V LVTTTL	rdn8a		rdn8a	<Direct>
T6	6		3.3-V LVTTTL	3.3-V LVTTTL	rdn6a	rdn6a	rdn6a	
T7	6		3.3-V LVTTTL	3.3-V LVTTTL	rdn4a	rdn4a	rdn4a	
T8	6		3.3-V LVTTTL	3.3-V LVTTTL	rdn2a	rdn2a	GROUND	discordance de nom
T9	6		3.3-V LVTTTL	3.3-V LVTTTL	rdn1a	rdn1a	rdn1a	<Direct>
T10		✓	<GND>	<GND>	<GND>	<GND>	GROUND	<Ground>
T11		✓	<1.2V>				P1V2 (=1.2V)	<Power>
T12		✓	<GND>	<GND>	<GND>	<GND>	GROUND	<Ground>

Premiers résultats

La vérification initiale des noms de signaux sur FPGA vs PCB sans appliquer les règles entraîne 170 incohérences de noms et 5 erreurs de puissance sur 672 broches de l'appareil. Les résultats initiaux - comme indiqué ci-dessus - sont classés par nombre de broches afin de montrer une variété de problèmes potentiels.

Pour obtenir le même point de vue, comme ci-dessus, vous devez modifier l'ordre de tri de la vue du signal.



Cliquez sur le bouton **Signalview sort order**. Sélectionnez le **PIN** dans la colonne de tri primaire. Fermez la boîte de dialogue en cliquant sur le bouton **OK** et faites défiler jusqu'à PAD R22

Le code de couleur pour les noms de signaux :

Rouge:	Nom erreurs d'encodage / d'alimentation
Vert foncé :	Match de l'utilisateur accepté
Vert :	Correspondance de la règle de périphérique
Vert :	Correspondance de la règle de l'utilisateur
Feu vert:	Correspondance de la règle floue/partielle
Noir:	Correspondance directe

Il y a 210 correspondances (nom) exactes, 65 correspondances de masse, 81 correspondances de puissance, 134 correspondances floues et 7 correspondances de règles de l'appareil. Les correspondances floues sont des signaux qui correspondent lorsque :

1. Retrait du caractère '\_' (de soulignement) dans le nom
2. En supposant que les caractères d'index de bus '<' et '>' peuvent être ignorés. (Donc données <1> correspond à data\_1).

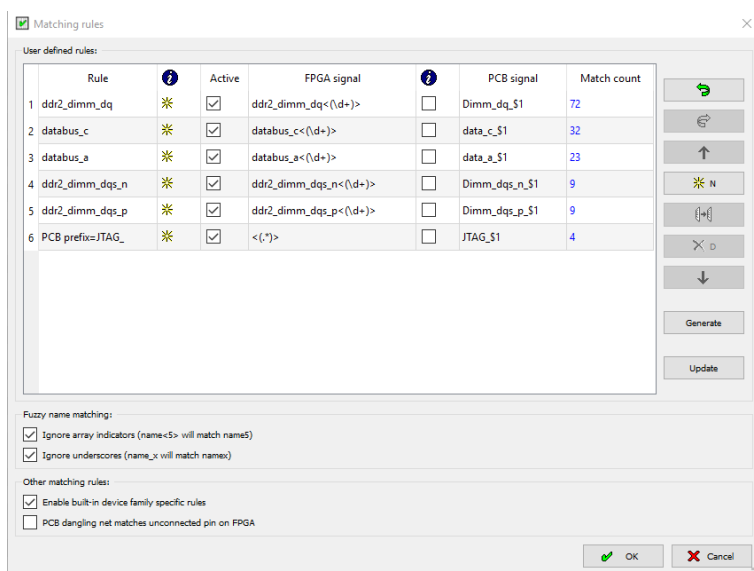
Les correspondances de règles de l'appareil sont les règles d'alimentation / de masse.

Les premières incohérences ne sont pas des erreurs mais proviennent seulement du fait que des noms différents ont été utilisés dans l'environnement FPGA et PCB. L'erreur de tension est un vrai problème, mais nous allons d'abord nous concentrer sur la réduction des incohérences de signal en appliquant le générateur de règles.



Sélectionnez l'option de menu **Tools >> Rule generator**

Le générateur de règle essaie de déterminer les règles (en utilisant des expressions régulières) qui, lorsqu'elles sont appliquées, correspondent au moins 3 fois. Nous supposons qu'il y a une relation entre le nom du signal FPGA et PCB et qu'ils sont connectés comme prévu.



Boite de dialogue de génération de règle



Cliquez sur le bouton **OK**.

Après avoir cliqué sur le bouton **OK**, **IO Checker** montre un rapport de règle de correspondance et l'état des incohérences dans la vue console. En appliquant les règles proposées, nous avons réduit le nombre d'erreurs potentielles à 21. (Les règles ont généré 149 correspondances)

Les règles de correspondance indiquent :

- Règle ddr2\_dimm\_dq correspond 72 fois
- Règle databus\_c correspond 32 fois
- Règle databus\_a correspond 23 fois
- Règle ddr2\_dimm\_dqs\_n correspond 9 fois
- Règle ddr2\_dimm\_dqs\_p correspond 9 fois
- Préfixe règle de PCB = JTAG\_ correspond 4 fois

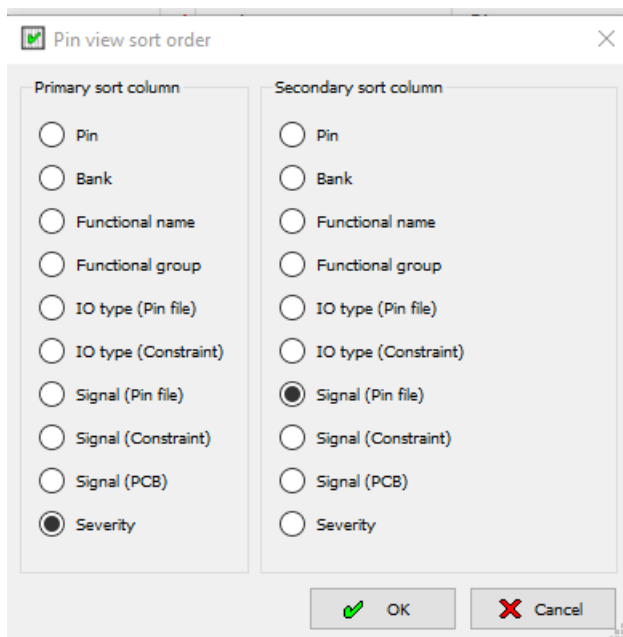
Le dispositif contient 672 broches, dont :

- 26 ne correspondent pas
- 210 correspondances exactes
- 7 règles de correspondance à l'appareil
- 134 correspondances partielles
- 149 règles de correspondance
- 0 matchs de l'utilisateur
- 65 matchs au sol
- 81 matchs de puissance
- 5 erreurs de puissance
- 0 avertissements de puissance


## Se concentrer sur les problèmes potentiels



Les 29 incohérences de signaux sont éparpillées sur les 672 broches de l'appareil. En triant la vue du signal par gravité et par nom de signal FPGA, nous pouvons nous concentrer sur les problèmes potentiels.



Boîte de dialogue vue de signal

 Cliquez sur le bouton **Pin view sort order** dans la vue de signal. Sélectionnez **Severity** (gravité) dans la colonne de tri primaire et sélectionnez **Signal (fichier Pin)** dans la colonne de tri secondaire. Fermez la boîte de dialogue en cliquant sur le bouton **OK**.

En ce moment, nous n'avons aucun intérêt dans les données de contrainte ainsi nous filtrons ces colonnes.

 Cliquez sur le bouton **Select column filters** . Sélectionnez **Hide IO type (fichier de contrainte de FPGA)** et cliquez quelque part en dehors de la boîte pour fermer la boîte.

Nous avons maintenant toutes les incohérences ensemble en vue et nous pouvons voir que le nombre d'incohérences signalées ne sont pas vraiment des incohérences, mais des noms différents dans le FPGA et le PCB.

Pins: 672, unmatched: 26 (21 + 5), matched: 646 (210 exact, 65 ground, 81 power, 134 fuzzy, 7 device, 0 dangling, 149 rule and 0 user)

	Pin	Bank	IO type (Pin file)	Signal (Pin file)	Signal (Constraint)	Signal (PCB)	Rule
⊗	R10	6	✗ <3.3V>			P1V2 (=1.2V)	
⊗	H12		✗ <3.3V>			VCC_P1LLOUT (←P1V2 = 1.2V)	
⊗	T1	6	✗ <3.3V>			P1V2 (=1.2V)	
⊗	AC1	6	✗ <3.3V>			P1V2 (=1.2V)	
⊗	V13		✗ <3.3V>			VCC_P1LLOUT (←P1V2 = 1.2V)	
⊗	A8	4	3.3-V LVTTTL	databus_a[12]	databus_a<12>	data_a_21	
⊗	F10	4	3.3-V LVTTTL	databus_a[21]	databus_a<21>	data_a_12	
⊗	T8	6	3.3-V LVTTTL	rdn2a	rdn2a	GROUND	
⊗	U5	6	3.3-V LVTTTL	seg7_01	seg7_01	seg7_0	
⊗	T4	6	3.3-V LVTTTL	seg7_02	seg7_02	seg7_1	
⊗	U3	6	3.3-V LVTTTL	seg7_03	seg7_03	seg7_2	
⊗	U4	6	3.3-V LVTTTL	seg7_04	seg7_04	seg7_3	
⊗	T3	6	3.3-V LVTTTL	seg7_05	seg7_05	seg7_4	
⊗	T2	6	3.3-V LVTTTL	seg7_06	seg7_06	seg7_5	
⊗	U1	6	3.3-V LVTTTL	seg7_07	seg7_07	seg7_6	
⊗	N20	2		<GND*>		rdn2a	
⊗	G8	4		<MSEL0>		GROUND	
⊗	B2	4		<MSEL1>		P1V2 (=1.2V)	
⊗	E6	4		<MSEL2>		P1V2 (=1.2V)	
⊗	F7	4		<MSEL3>		GROUND	
⊗	E22	3		<nCE>		GROUND	
⊗	AB5	7		<nCEO>			
⊗	F5	1		<TEMPDIODEn>			

Présentation de la vue de signal après application des règles

## Ajouter une règle d'utilisateur

En regardant les « mismatches » des signaux du seg7 (pads U5, T4,...) il est facile de voir que le fichier des signaux des pins utilise des indices de 1 à 7 et le PCB utilise des indices de 0 à 6. Nous pouvons ajouter une règle pour vérifier les tableaux qui ne sont pas alignés de la même manière.

Cliquez sur le bouton **Project matching rules** dans la barre de bouton principale.

Créons une nouvelle règle pour faire correspondre les signaux de seg7 ensemble.

Cliquez sur le bouton **Create new row** et remplissez les données :

Rule:	Seg7
Active:	<input checked="" type="checkbox"/>
FPGA pin	seg7_0(\d)
PCB pin:	seg7_\$_1-1

Cliquez sur le bouton de **mise à jour**.

La boîte de dialogue Règles de correspondance devrait ressembler à :

Rule	Active	FPGA signal	PCB signal	Match count
1 ddr2_dimm_dq	<input checked="" type="checkbox"/>	ddr2_dimm_dq<(\d+)>	Dimm_dq_\$_1	72
2 <NEW RULE>	<input checked="" type="checkbox"/>	seg7_0(\d)	seg7_\$_1-1	7
3 databus_c	<input checked="" type="checkbox"/>	databus_c<(\d+)>	data_c_\$_1	32
4 databus_a	<input checked="" type="checkbox"/>	databus_a<(\d+)>	data_a_\$_1	23
5 ddr2_dimm_dqs_n	<input checked="" type="checkbox"/>	ddr2_dimm_dqs_n<(\d+)>	Dimm_dqs_n_\$_1	9
6 ddr2_dimm_dqs_p	<input checked="" type="checkbox"/>	ddr2_dimm_dqs_p<(\d+)>	Dimm_dqs_p_\$_1	9
7 PCB prefix=JTAG_	<input checked="" type="checkbox"/>	<(.*)>	JTAG_\$_1	4

Fuzzy name matching:  
 Ignore array indicators (name<5> will match name5)  
 Ignore underscores (name\_x will match namex)

Other matching rules:  
 Enable built-in device family specific rules  
 PCB dangling net matches unconnected pin on FPGA

*Ajout de la règle 'Seg7'*

La règle Seg7 doit correspondre 7 fois après avoir cliqué sur le bouton de mise à jour. (Si ce n'est pas le cas, veuillez examiner modifier la règle Seg7).



Cliquez sur le bouton **OK**.

**IO Checker** répète le rapport de règles de correspondance dans la vue de console avec un résumé de l'ensemble des correspondances.

Les règles de correspondance indiquent :

- Règle ddr2\_dimm\_dq correspond 72 fois
- Règle databus\_c correspond 32 fois
- Règle databus\_a correspond 23 fois
- Règle ddr2\_dimm\_dqs\_n correspond 9 fois
- Règle ddr2\_dimm\_dqs\_p correspond 9 fois
- Préfixe règle de PCB = JTAG\_ correspond 4 fois
- Règle Seg7 correspond 7 fois

Le dispositif contient 672 broches, dont :

- 14 ne correspondent pas
- 210 correspondances exactes
- 7 règles de correspondance à l'appareil
- 134 correspondances partielles
- 156 règles de correspondance
- 0 matchs de l'utilisateur
- 65 matchs au sol
- 81 matchs de puissance
- 5 erreurs de puissance
- 0 avertissements de puissance

## Incompatibilités utilisateur acceptées



Après l'application de la règle Seg7 à la **boîte de dialogue des règles de correspondance** nous n'avons que 14 incohérences restantes. Comme la vue du signal est triée par «gravité» nous voyons toutes les incohérences et erreurs de puissance au sommet de la fenêtre.

Certaines des incohérences des signaux sont en effet destinées. Par exemple, les signaux de contrôle FPGA MSEL0 - MSEL3 qui devraient être reliés soit à la masse soit à VCC. Sélectionnons les signaux de commande et flag d'utilisateur sur ces derniers.

Pins: 672, unmatched: 19 (14 + 5), matched: 653 (210 exact, 65 ground, 81 power, 134 fuzzy, 7 device, 0 dangling, 156 rule and 0 user)

Pin	Bank	IO type (Pin file)	Signal (Pin file)	Signal (Constraint)	Signal (PCB)	Rule
R10	6	<3.3V>			P1V2 (=1.2V)	
V13		<3.3V>			VCC_P1V2 (=1.2V)	
H12		<3.3V>			VCC_P1V2 (=1.2V)	
T1	6	<3.3V>			P1V2 (=1.2V)	
AC1	6	<3.3V>			P1V2 (=1.2V)	
A8	4	3.3-V LVTTTL	databus_a[12]	databus_a<12>	data_a_21	
F10	4	3.3-V LVTTTL	databus_a[21]	databus_a<21>	data_a_12	
T8	6	3.3-V LVTTTL	rdn2a	rdn2a	GROUND	
N20	2		<GND*>		rdn2a	
G8	4		<MSEL0>		GROUND	
B2	4		<MSEL1>		P1V2 (=1.2V)	
E6	4		<MSEL2>		P1V2 (=1.2V)	
F7	4		<MSEL3>		GROUND	
E22	3		<nCE>		GROUND	
AB5	7		<nCEO>			
F5	1		<TEMPDIODEn>			
E5	1		<TEMPDIODEp>			
AB21	8		<TRST>		\$2134	
E16	3	3.3-V LVTTTL	<~DATA0~ / RESERVED_INPUT>		RESERVED_INPUT	
AE3	7	3.3-V LVTTTL	speaker_out	speaker_out	speaker_out	<Direct>

*Incompatibilités utilisateur acceptées*



Sélectionner les signaux FPGA **MSEL0 - MSEL3**, **RCE**, **CCEC**, **TEMPDIODEn**, **TEMPDIODEp** et **Data0** et clic droit. Sélectionnez **Accept mismatch** dans le menu contextuel comme indiqué dans l'image ci-dessus.

Pins: 672, unmatched: 10 (5 + 5), matched: 662 (210 exact, 65 ground, 81 power, 134 fuzzy, 7 device, 0 dangling, 156 rule and 9 user)

	Pin	Bank	IO type (Pin file)	Signal (Pin file)	Signal (Constraint)	Signal (PCB)	Rule
●	R10	6	✗ <3.3V>			P1V2 (=1.2V)	
●	H12		✗ <3.3V>			VCC_PLLOUT (←P1V2 = 1.2V)	
●	V13		✗ <3.3V>			VCC_PLLOUT (←P1V2 = 1.2V)	
●	T1	6	✗ <3.3V>			P1V2 (=1.2V)	
●	AC1	6	✗ <3.3V>			P1V2 (=1.2V)	
●	A8	4	3.3-V LVTTTL	databus_a[12]	databus_a<12>	data_a_21	
●	F10	4	3.3-V LVTTTL	databus_a[21]	databus_a<21>	data_a_12	
●	T8	6	3.3-V LVTTTL	rdn2a	rdn2a	GROUND	
●	N20	2		<GND*>		rdn2a	
●	AB21	8		<TRST>		\$2134	
●	AE3	7	3.3-V LVTTTL	speaker_out	speaker_out	speaker_out	<Direct>
●	G8	4		<MSEL0>		GROUND	<User>
●	B2	4		<MSEL1>		P1V2 (=1.2V)	<User>
●	E6	4		<MSEL2>		P1V2 (=1.2V)	<User>
●	F7	4		<MSEL3>		GROUND	<User>
●	E22	3		<nCE>		GROUND	<User>
●	AB5	7		<nCEO>		☒	<User>
●	F5	1		<TEMPDIODEn>		☒	<User>
●	E5	1		<TEMPDIODEp>		☒	<User>
●	E16	3	3.3-V LVTTTL	<~DATA0~ / RESERVED_INPUT>		RESERVED_INPUT	<User>
●	B25	3		<CONF_DONE>		CONF_DONE	<Fuzzy>

Vue de signal : résultat final





Maintenant, nous n'avons que 10 incohérences restantes (5 incohérences de noms potentiels et 5 erreurs de puissance) et il est facile de voir les erreurs commises lors de la connexion du FPGA sur le PCB :

1. Les signaux **databus\_a <12>** et **databus\_ <21>** ont été échangés
2. Le signal de PCB «rdn2a» a été connecté à la broche **T8** mais devrait être sur **N20**
3. Pour le signal TRST il n'est pas possible de déterminer si il a été correctement connecté, en raison du PCB généré '**\$2134**'

Vous devez maintenant vérifier si ces incohérences sont des erreurs (comme le **databus\_a <21>** et **databus\_a<12>** signaux qui ont été commutés), ou si ce sont des erreurs de design. Quand il y a des erreurs de conception, elles doivent être fixées dans l'environnement FPGA ou PCB.

## Vérification voltages et masses

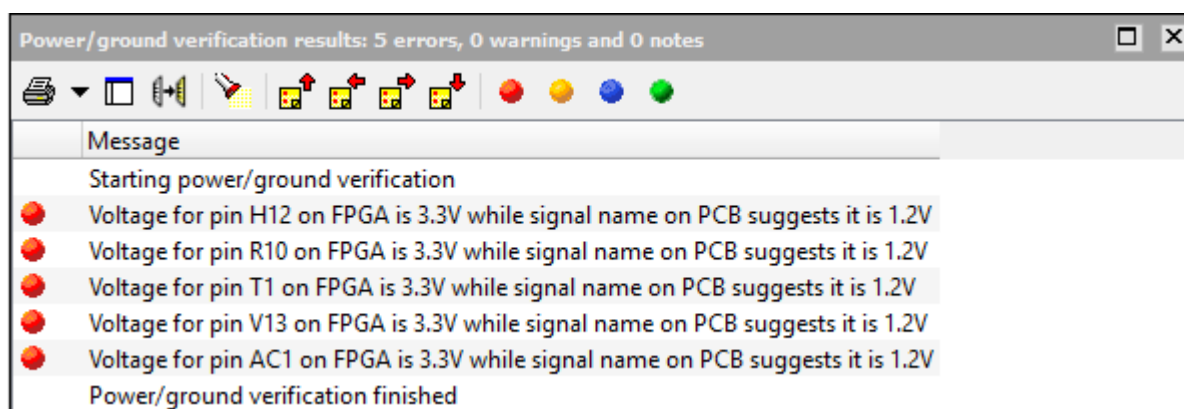
**IO Checker** peut également vérifier certaines règles de tension pour les banques IO. Le fichier pinlist et les règles de la famille de l'appareil définissent quelles broches sont des broches d'alimentation et la tension requise pour ces broches. Les broches d'alimentation sont marquées avec l'icône d'alimentation dans la première colonne de la vue du signal. La colonne 5 indique l'icône d'état pour le contrôle de la tension. Lorsqu'aucune icône n'est présente, le contrôle de la tension n'est pas applicable pour la broche. Lorsque la tension ne peut pas être déduite du nom net PCB ou qu'elle n'a pas été précisée dans la Pin liste FPGA un point d'interrogation orange s'affiche.

Icon	Meaning
	Error detected
	FPGA voltage not specified or PCB voltage could not be determined
	OK
	User accepted power mismatch



Cliquez sur le bouton

La vérification indique les incohérences de tension dans la vue de vérification.



Le rapport ci-dessus montre qu'il y a 5 erreurs de tension et 0 avertissements. La première erreur de tension indique que la broche "A16" a été connectée à la tension de 1,2 alors que la broche de FPGA et / ou les règles spécifiques à chaque dispositif définissent une tension de 3,3.

Maintenez la souris sur l'icône de tension pour afficher le message.

Pins: 672, unmatched: 10 (5 + 5), matched: 662 (210 exact, 65 ground, 81 power, 134 fuzzy, 7 device, 0 dangli

Pin	Bank	IO type (Pin file)	Signal	Signal (PCB)	Rule
R10	6	<3.3V>			
H12		<3.3V>			
V13		<3.3V>			
T1	6	<3.3V>			
AC1	6	<3.3V>			
A8	4	3.3-V LVTTTL	data_a_21	data_a_21	
F10	4	3.3-V LVTTTL	data_a_12	data_a_12	
T8	6	3.3-V LVTTTL	rdn2a	rdn2a	
N20	2		<GND*>		
A...	8		<TRST>	\$2134	
AE3	7	3.3-V LVTTTL	speaker_out	speaker_out	<Dire
G8	4		<MSEL0>	GROUND	<Use
B2	4		<MSEL1>	P1V2 (=1.2V)	<Use

Name : V13  
 Device :  
 Functional name : VCC\_PLL6\_OUT  
 Functional group : VCCIO  
 Pin file :  
 Functional group : VCCIO  
 Signal name : VCC\_PLL6\_OUT  
 Kind : power  
 Voltage : 3.3V  
 PCB netlist :  
 Signal name : VCC\_PLLOUT  
 Voltage : 1.2V

Erreur de tension dans la vue de Signal

La plupart des erreurs de tension doivent être fixées sur le PCB.

## Mise à jour du fichier contrainte

Notre conclusion de la comparaison entre le fichier de broches et la netlist PCB était :

1. Les signaux **databus\_a <12>** et **databus\_ <21>** ont été échangés
2. Le signal de PCB **rdn2a** a été connecté à la broche **T8** mais devrait être sur **N20**
3. Pour le signal **TRST** il n'est pas possible de déterminer si il a été correctement connecté, en raison de la PCB générée **\$2134**

Supposons que le numéro 3 est correctement connecté, les deux autres discordances doivent être résolues dans l'outil FPGA ou PCB. Pour les outils FPGA **IO Checker** est capable de créer ou mettre à jour un fichier de contraintes. Ce fichier peut être utilisé pour régénérer le fichier de broche. Nous allons maintenant expliquer comment mettre à jour le fichier de contrainte. Nous allons d'abord réafficher les colonnes de données de contrainte.



Cliquez sur le bouton **Hide column filter** Masquer et effacer la sélection. La colonne signal (contrainte) montre toutes les contraintes de localisation définies dans le fichier de contrainte. Normalement les colonnes du signal (FPGA) et du signal (contrainte) devraient avoir les mêmes noms de signaux.

Nous allons maintenant modifier la contrainte de localisation pour les trois signaux mal connectés. Vous pouvez utiliser trois commandes pour changer une contrainte de localisation :

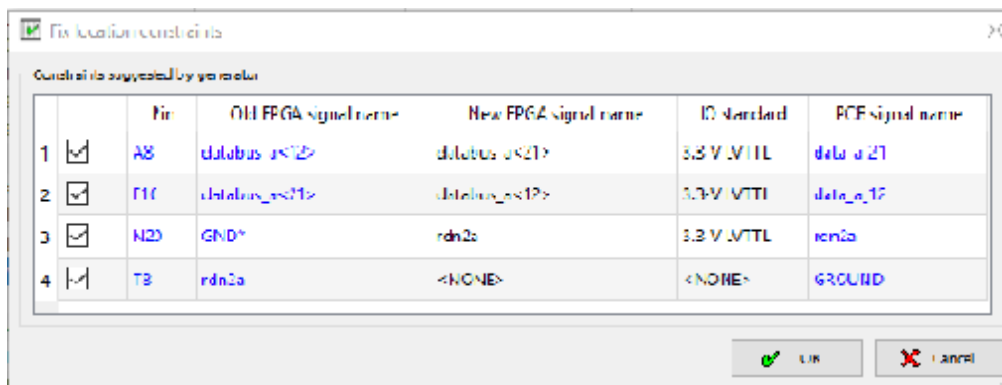
- Utilisez la commande de résolution de contrainte
- Sélectionner deux lignes et sélectionnez **Swap pins** dans le menu contextuel de la vue de signal.
- Sélectionnez **Propriétés** dans le menu contextuel de la vue de signal et entrez le nom du signal dans la boîte de dialogue

Nous allons utiliser la commande de contrainte de solution. C'est la commande la plus simple, dans ce cas, nous laisserons **IO Checker** faire le travail.



Sélectionnez **Tools >> Fix location constraints**.

La **boîte de dialogue de génération de contrainte** apparaît avec des suggestions pour les changements de contraintes.



Dialogue de génération contrainte

Comme nous pouvons le voir dans la boîte de dialogue, **IO Checker** suggère **d'échanger** les deux signaux de databus\_a et placer le rdh2a de signal à la **broche numéro "N20"**.

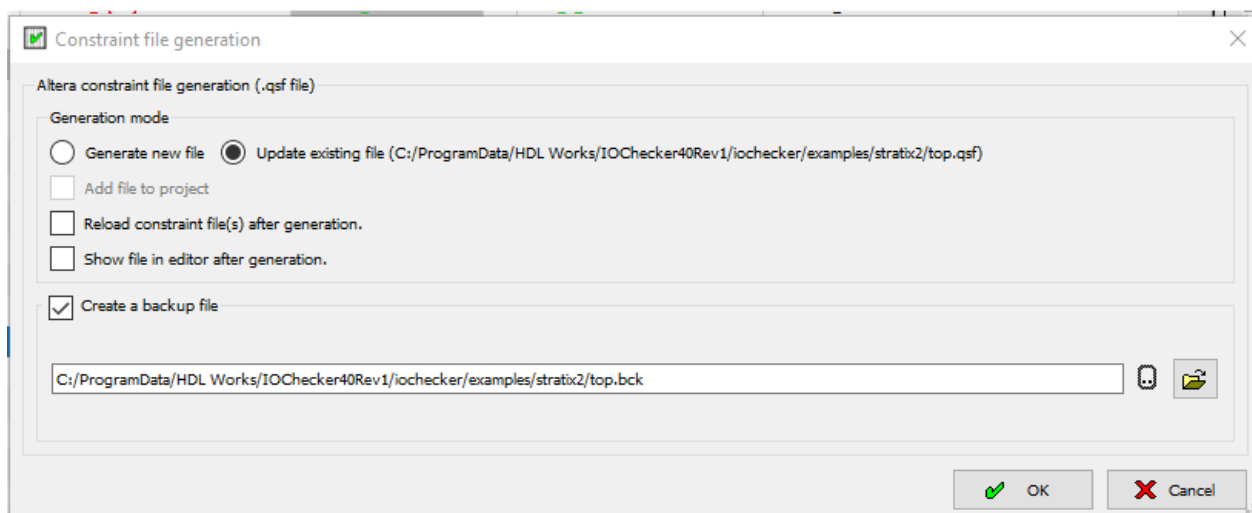


Acceptez les suggestions en cliquant sur le bouton **OK**.

Enfin, nous devons mettre à jour le fichier de contraintes afin que nous puissions relancer l'outil de FPGA pour régénérer le fichier de broche.



Sélectionnez **Tools >>Generate Constraint file**.



Dialogue de génération de fichiers contrainte

La boîte de dialogue de génération du fichier de contrainte de dialogue apparaît. Dans ce dialogue, vous pouvez choisir de générer un nouveau fichier ou mettre à jour le fichier existant.



Sélectionnez la mise à jour du fichier existant et autoriser le rechargement du fichier de contraintes après génération.



Mettez à jour le fichier de contrainte en cliquant sur le bouton **OK**.

La vue de console montre les modifications :

```
Processing file: $INSTALL/iochecker/examples/stratix2/top.qsf
Found: 12 global assignments
Found: 357 pin location assignments
Found: 356 IO standard assignments
```

Updating constraint file '\$INSTALL/iochecker/examples/stratix2/top.qsf'  
Modified location for signal 'rdn2a' (old: 'T8', new: 'N20')  
Modified location for signal 'databus\_a[21]' (old: 'F10', new: 'A8')  
Modified location for signal 'databus\_a[12]' (old: 'A8', new: 'F10')

Removed IoStandard for signal 'rdn2a'  
Finished updating constraint file '\$INSTALL/iochecker/examples/stratix2/top.qsf'

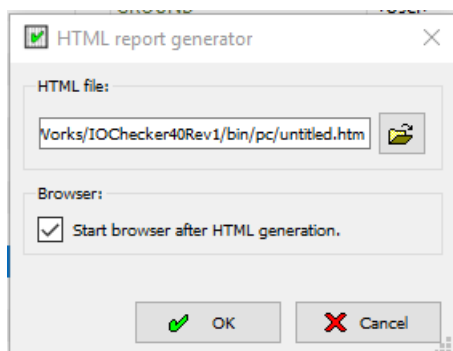
## Générateur HTML



Une des dernières étapes de ce tutoriel vous permet de générer un rapport HTML. Le **HTML Report Generator** crée un rapport de votre projet contenant :

- Un résumé
- Un rapport des règles
- Une table de Pin, triés par gravité
- Une table de Pin, triés par code PIN
- Une vérification des résultats

Le fichier HTML proposé est situé par défaut dans le répertoire du projet et a une extension htm, mais vous pouvez le placer n'importe où. Dans un projet sans titre le nom par défaut est untitled.htm. Le fichier HTML généré est autonome et peut être copié à n'importe quel endroit.



*Rapport HTML Générateur de dialogue*



Cliquez sur le bouton **HTML Report Generator**. Définissez le nom et l'emplacement souhaité. Fermez la boîte de dialogue en cliquant sur le bouton **OK**.

Après la fermeture de la boîte de dialogue, le rapport HTML est généré et le navigateur HTML, associé à des fichiers HTML sur votre système, sera lancé. Vous verrez que les tables de broches ont une grande ressemblance avec la vue du signal.

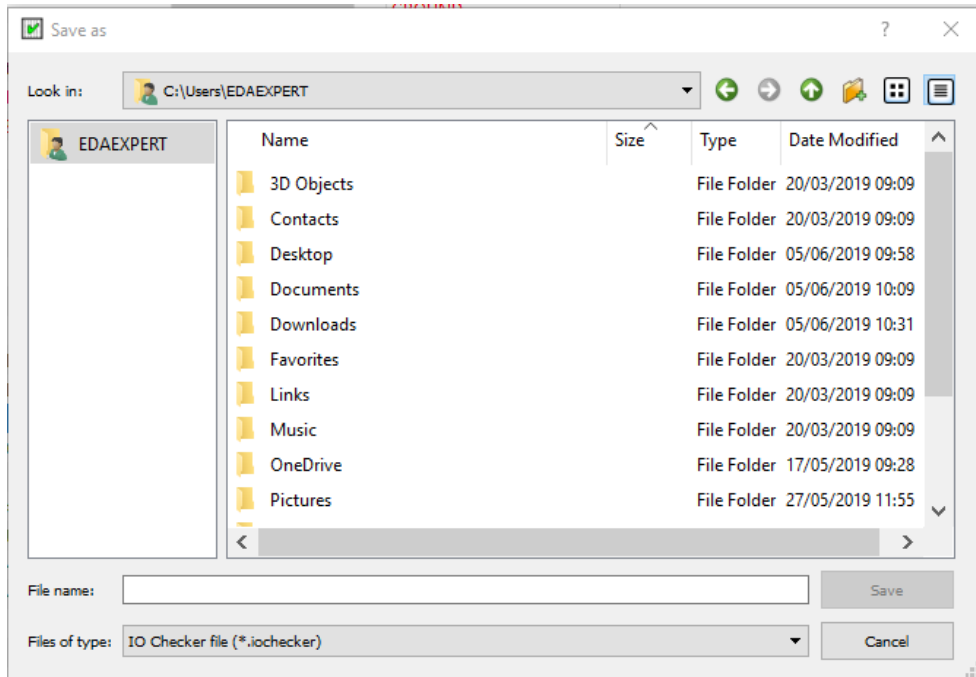
## Sauvegarder le projet

La dernière étape de ce tutoriel est de sauvegarder les résultats dans un fichier de projet.



Cliquez sur le bouton **Save the current project**.

La boîte de dialogue **Enregistrer sous** apparaît dans laquelle vous pouvez parcourir le répertoire dans lequel vous voulez placer le fichier de projet et indiquer un nom pour le fichier de projet.



Boîte de dialogue : enregistrer en tant que



Naviguez jusqu'au répertoire où vous souhaitez placer votre fichier de projet et entrer "tutoriel" dans le champ **Nom de fichier**. Puis cliquez sur le bouton **Enregistrer**.

Vous avez maintenant enregistré votre travail dans le dossier projet de tutoriel iochecker.

Ceci termine notre tutoriel **IO Checker**. Nous espérons que vous avez apprécié la lecture de ce dernier.

Plus d'informations peuvent être trouvées dans le manuel de l'utilisateur **IO Checker** et sur le site web: [www.hdlworks.com](http://www.hdlworks.com).

## Informations contact

---



EDA Expert

11-13 avenue de la division Leclerc  
94230 Cachan  
France

Téléphone: +33 (0) 1 49 84 86 26

**Site web:** [www.eda-expert.com](http://www.eda-expert.com)

**Service clientèle:** [contact@eda-expert.com](mailto:contact@eda-expert.com)