

TRAVAUX PRATIQUES

SoPC

TP3 : CONCEPTION POUR ZYNQ AVEC LA CARTE PYNQ

1. CRÉER UNE CONCEPTION MATÉRIELLE POUR PYNQ

1.1 Créer un nouveau projet Vivado

Ce didacticiel créera une conception pour la carte PYNQ-Z2 (Zynq). Vous devriez pouvoir suivre les instructions pour créer une conception similaire pour d'autres cartes Zynq ou Zynq Ultrascale+.

- Ouvrez Vivado et créez un nouveau projet.
- Choisissez un nom de projet et sélectionnez votre tableau Zynq comme cible
Dans cet exemple, le PYNQ-Z2 est sélectionné.
- Sélectionnez RTL comme type de projet et cochez la case Ne pas spécifier de sources pour le moment
- Sous IP INTEGRATOR, cliquez sur Créer une conception de bloc
- Ajouter le IP : ZYNQ7 Processing System (PSBlock Automation).

1.2 Ajouter des blocs à votre conception

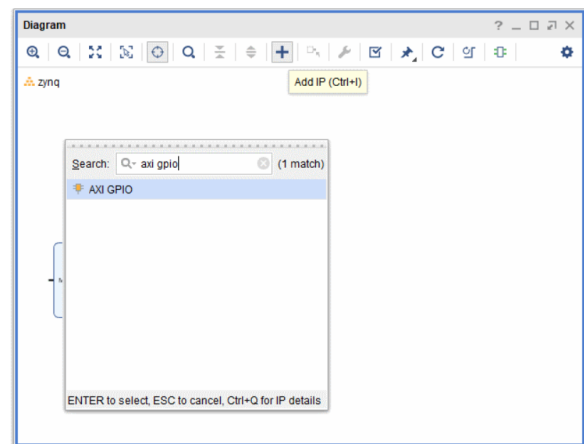
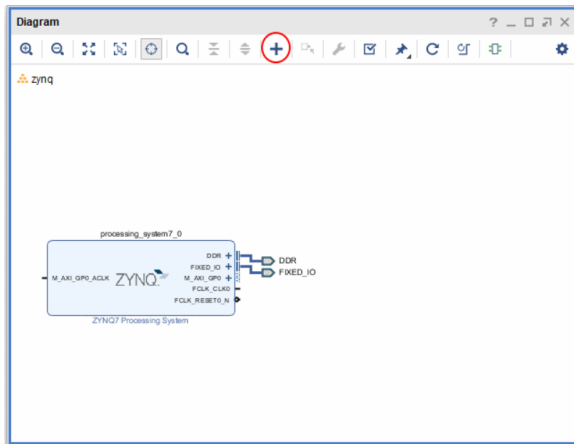
À ce stade, vous pouvez commencer à ajouter des blocs à votre conception. Vous pouvez ajouter une IP à partir du catalogue Vivado ou ajouter votre propre IP personnalisée.

1.2.1 Aperçu de la conception matérielle

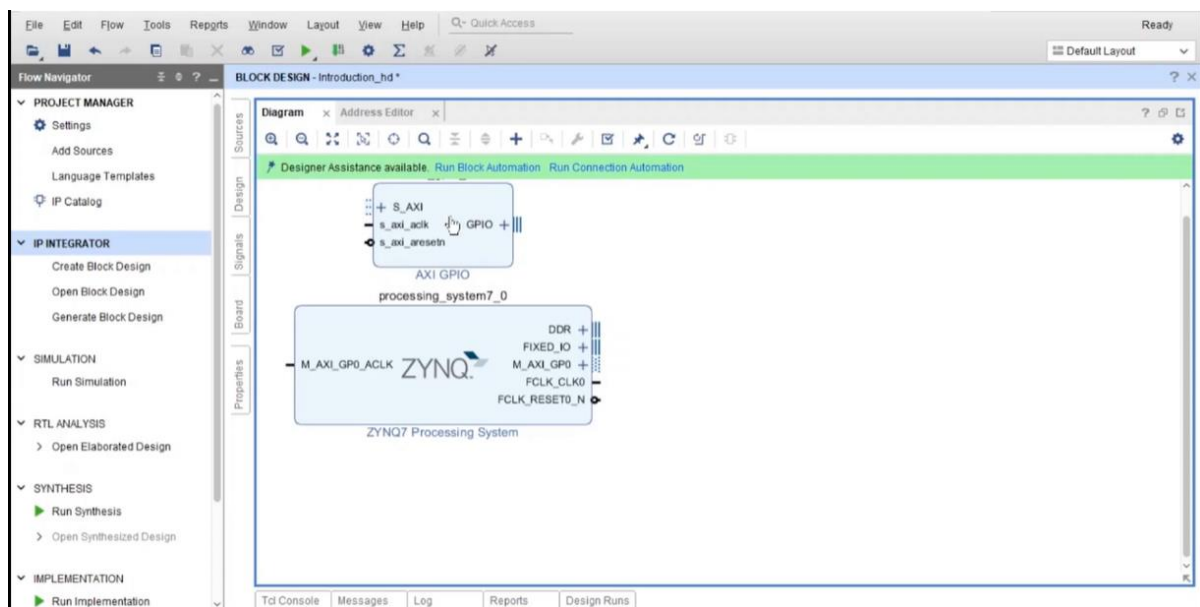
Cette étape consiste à présenter la disposition pour implémenter un contrôleur GPIO dans le Zynq PL.

L'outil Vivado IP Integrator sera utilisé pour créer cette interface à l'aide de l'environnement graphique fourni, et les outils Designer Assistance fournis seront utilisés pour la configuration et la connexion automatiques d'une interconnexion AXI qui connecte le Zynq PS au module IP du PL. Le contrôleur GPIO sera connecté aux LEDs disponibles sur le Pynq Board.

Un aperçu de la conception matérielle qui sera créée dans cet exercice est fourni dans la figure :



➤ Double cliquer sur le composant AXI GPIO



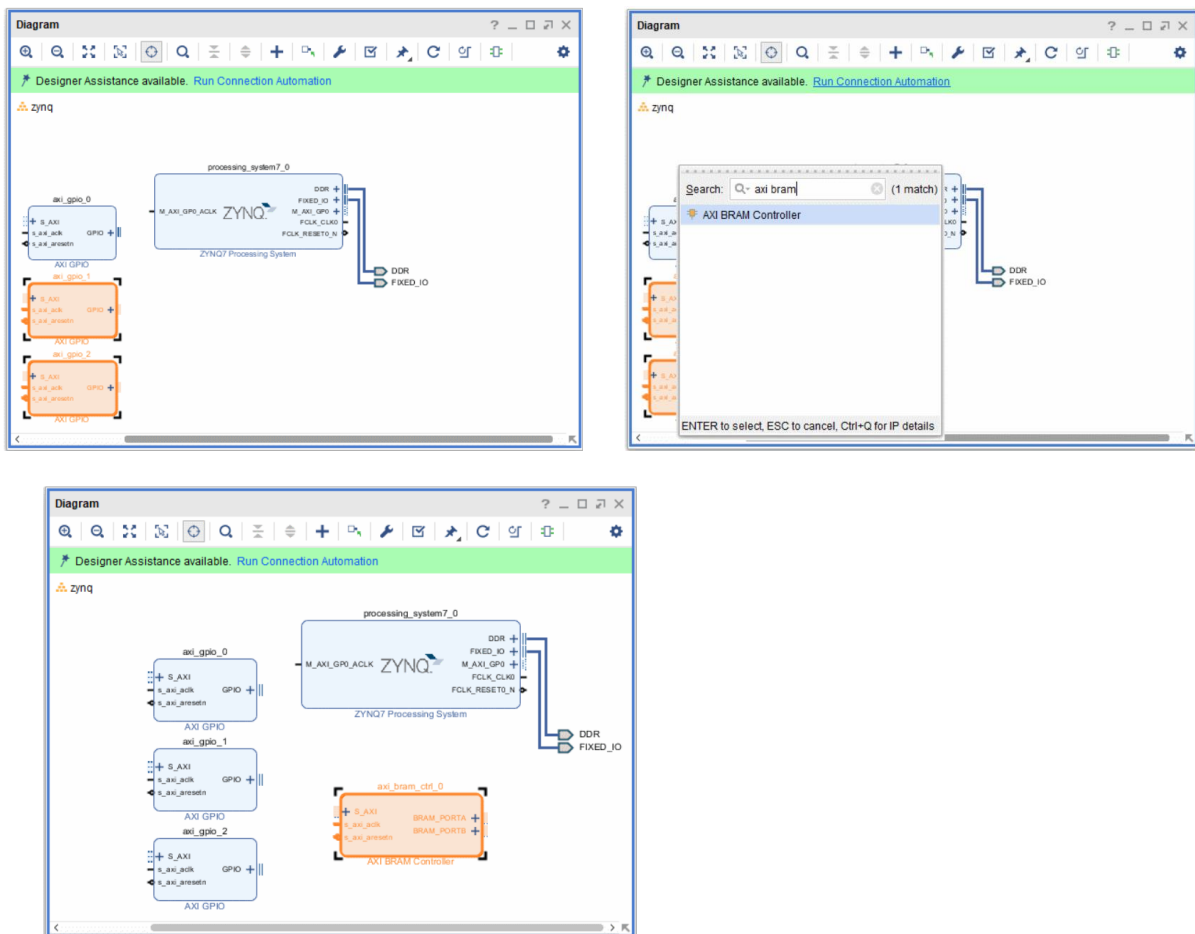
Deux autres IP supplémentaire GPIO, pour générer des interruptions matérielles **buttons**, et **switches**.

Dans cet exemple, nous ajouterons une IP du catalogue Vivado : une BRAM (une mémoire dans la structure FPGA) et 3x GPIO (contrôleurs d'entrée-sortie à usage général. Les blocs GPIO seront utilisés pour contrôler les LED, les boutons-poussoirs et les DIP. interrupteurs sur la carte PYNQ-Z2.

Dans le diagramme, suivez le même processus que précédemment pour rechercher les blocs suivants dans le catalogue IP et les ajouter à votre conception :

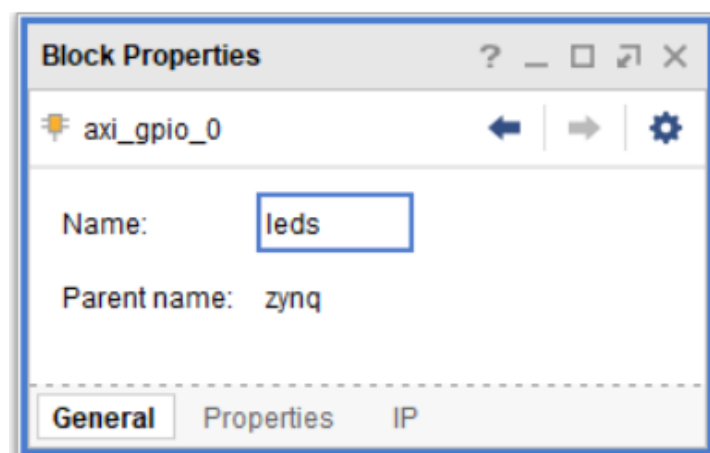
- 3x contrôleur AXI GPIO (vous pouvez ajouter une instance de ce bloc, puis le copier et le coller pour ajouter plus d'instances)

- 1x AXI BRAM (Assurez-vous de sélectionner l' **AXI BRAM**)

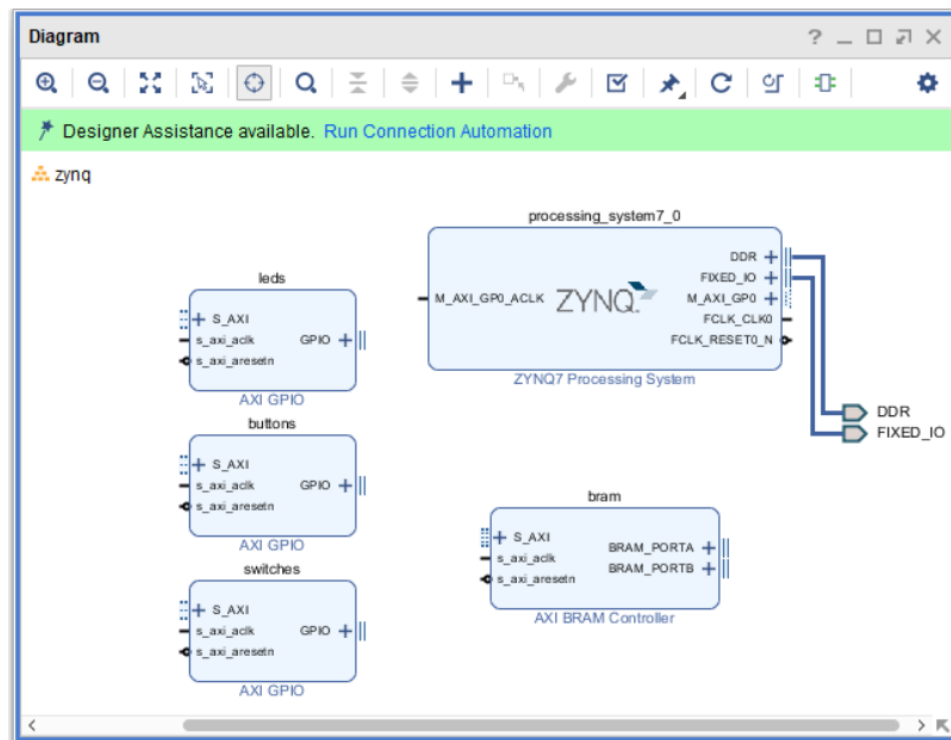


Les blocs reçoivent tous des noms par défaut. Par exemple, axi_gpio_0, axi_gpio_1 etc. Ces noms seront visibles depuis Python plus tard, il est donc utile de les renommer en quelque chose de significatif.

- Sélectionnez chaque bloc AXI GPIO, recherchez la fenêtre Propriétés du bloc et renommez chaque IP respectivement en **leds**, **buttons**, et **switches**.



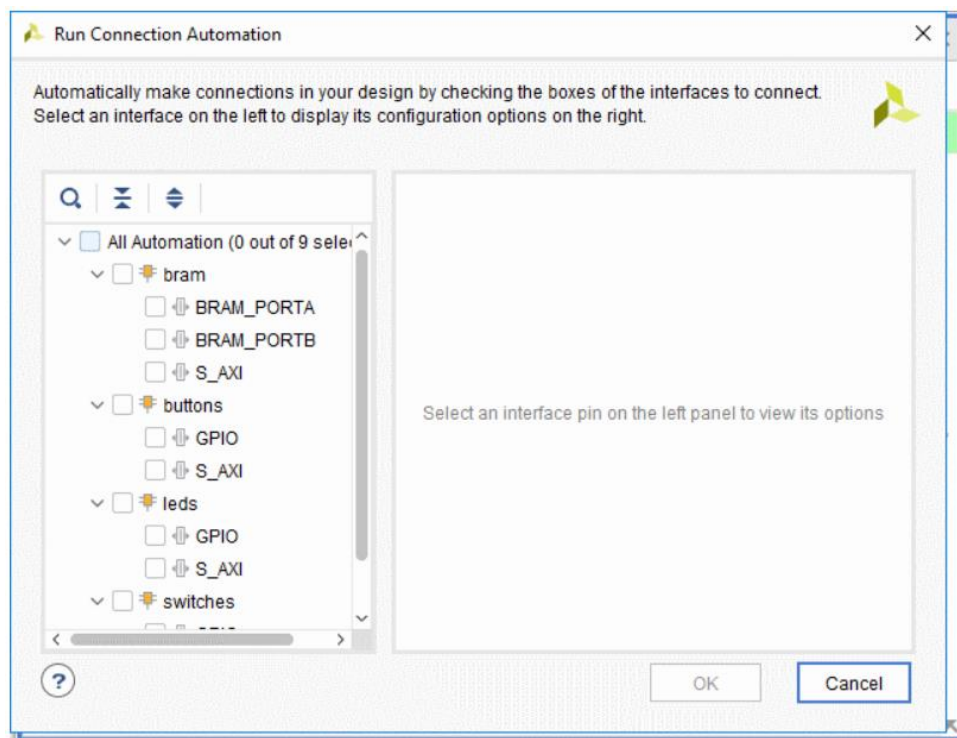
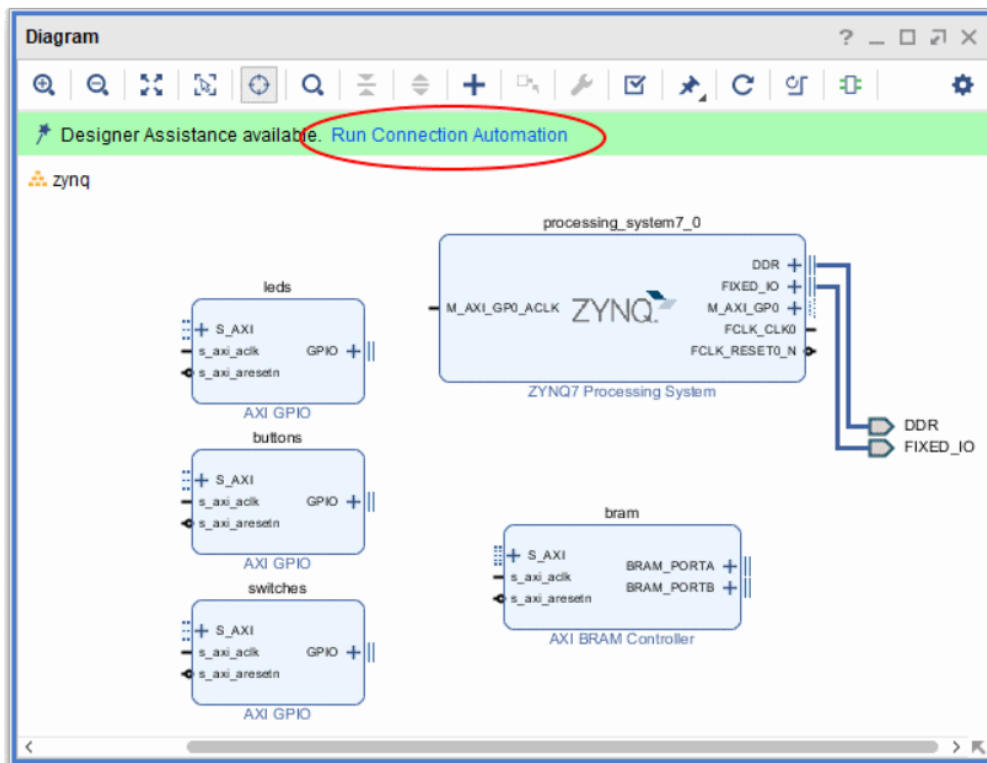
- Renommer *axi_bram_ctrl_0* en **bram**

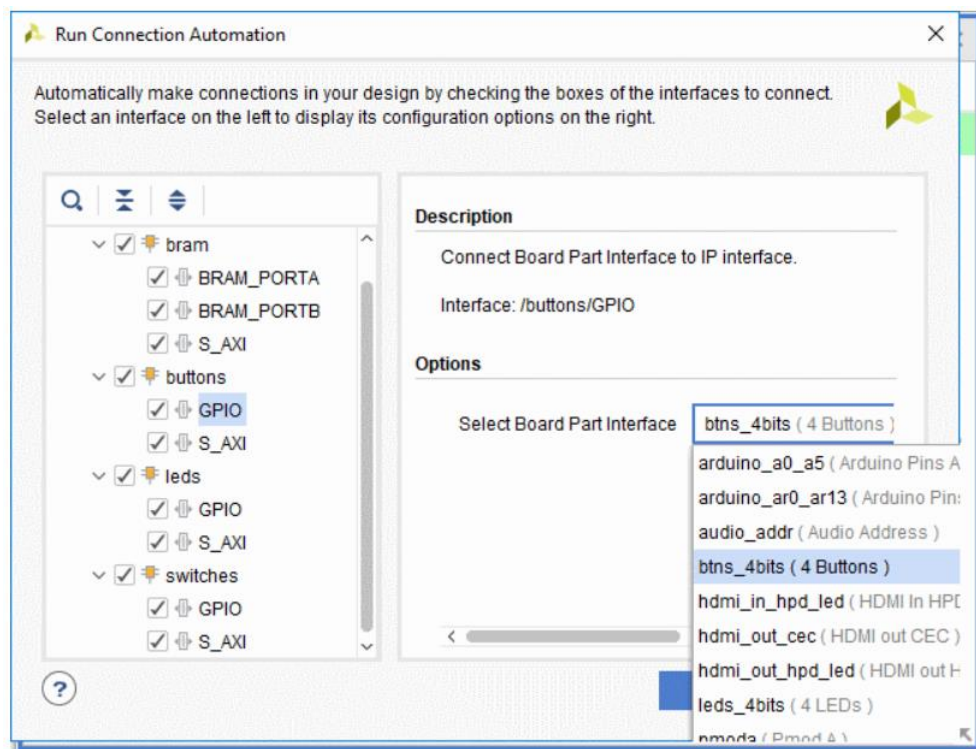
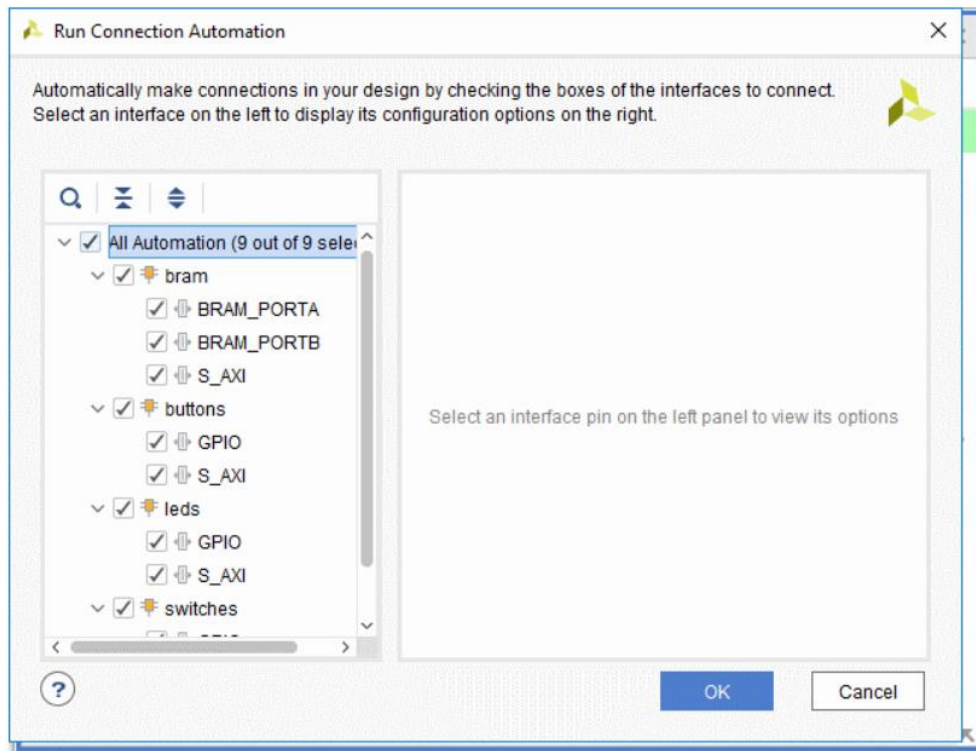


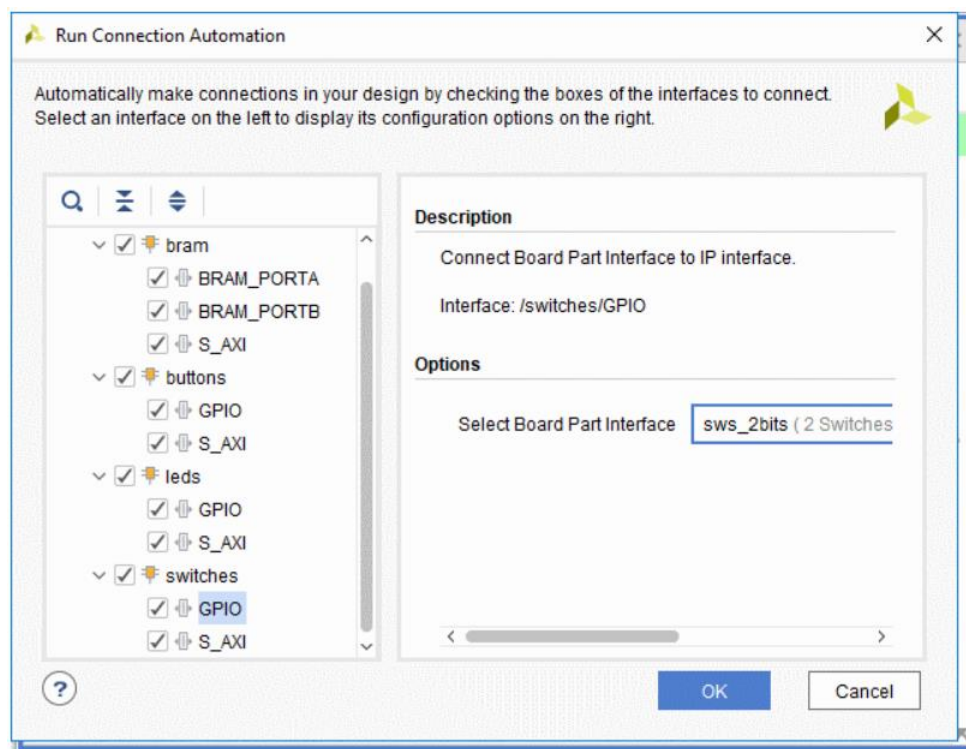
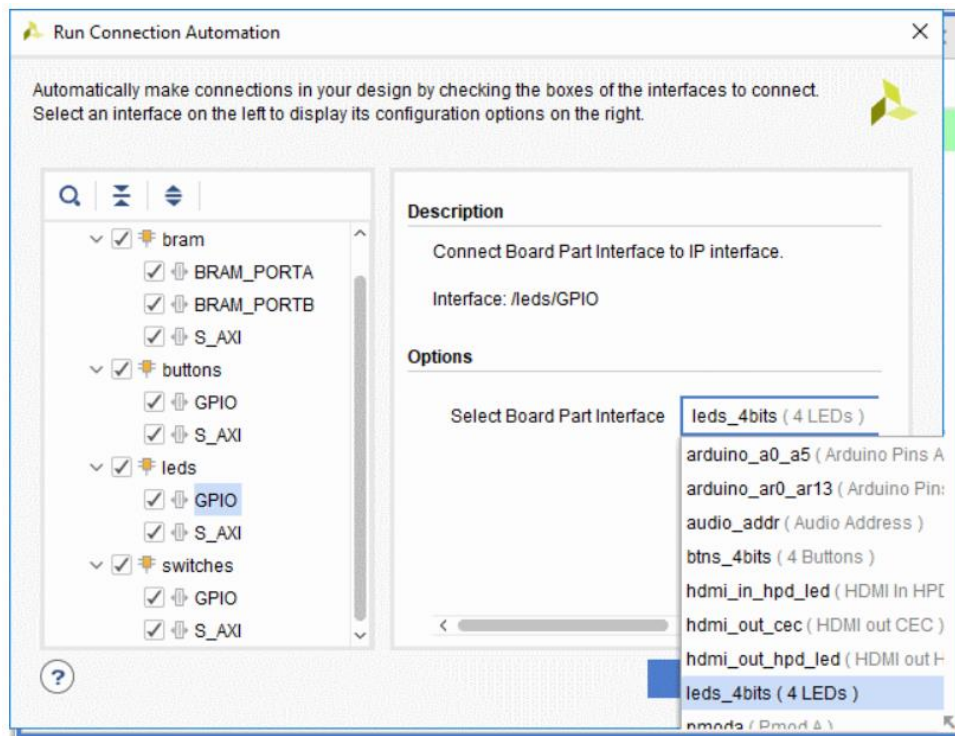
Vous pouvez créer manuellement des fils et connecter des ports en cliquant sur les ports et en faisant glisser le curseur vers un port correspondant, mais IP Integrator peut également connecter les ports automatiquement.

- Cliquez sur **Run connection automation** pour ouvrir la boîte de dialogue
- Cochez la case à côté de **All Automation**
- Pour chacun des blocs AXI GPIO (boutons, leds, switches), sélectionnez le label **GPIO** (veillez à ne pas décocher la case lors de la sélection)
- Pour chaque bloc, sous la section **Options**, sélectionnez l'interface correspondante. c'est à dire boutons: **btns_4bits**, leds: **leds_4bits**, interrupteurs: **sws_2bits**

Cela connectera chaque AXI GPIO aux broches correspondantes sur la carte.







- Cliquez sur OK pour exécuter l'automatisation de la connexion

Cela connectera automatiquement chaque bloc IP du système, ajoutera des ports et tous les blocs d'interconnexion et de réinitialisation supplémentaires dont le système a besoin.

Notez qu'un wrapper VHDL ou Verilog peut être créé, en fonction des paramètres du projet. (Si vous avez une préférence, vous pouvez également définir Verilog ou VHDL comme valeur par défaut dans les paramètres du projet.)

- Dans Vivado, cliquez sur **Generate Bitstream** (sous Program and Debug) en bas du *Flow Navigator*.

Vous remarquerez peut-être un certain nombre d'informations, d'avertissements et de messages dans le (même si, espérons-le, il n'y aura pas de messages d'erreur pour cette conception !). Ces messages peuvent être ignorés pour l'instant.

- Une fois le bitstream créé, générez le Tcl correspondant pour la conception en sélectionnant **File > Export > Export Block Design**. (Assurez-vous que la conception du bloc est ouverte lors de cette opération.)

2. CRÉER UNE CONCEPTION LOGICIELLE POUR PYNQ

Après avoir exporter le design, une interface de développement logicielle se lance :

Ecrire un programme qui permet de tester les interfaces conçu dans la partie conception matérielle.

Bon courage