# ESN11 : Co-design labs

TP1

Objectif : Intégrer une Nios II sur la carte DE10 pour jouer avec les fonctions d’un SoC. Le but est d’utiliser l’architecture logicielle pour exécuter un script qui va allumer les leds à tour de rôle de sorte à créer un chenillard.

Voici l’architecture à mettre en œuvre :

DE10-LITE

PIO

RAM + Cache

JTAG UART

NIOS II

FPGA

Bus AVALON

# CHENILLARD STANDARD :

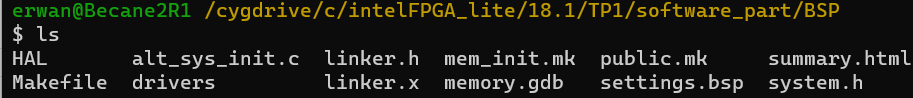
Voici l’architecture qsys réalisée sur platform designer

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

Une fois qu’on a fait l’analyse et le pin assignment, on va générer les fichiers .bsp dans le dossier /projetquartus/soft/BSP

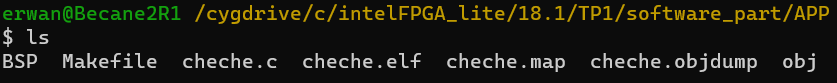




Voici le code source « cheche.c » qui va me permettre de réaliser le chenillard

Une image contenant texte, capture d’écran, logiciel

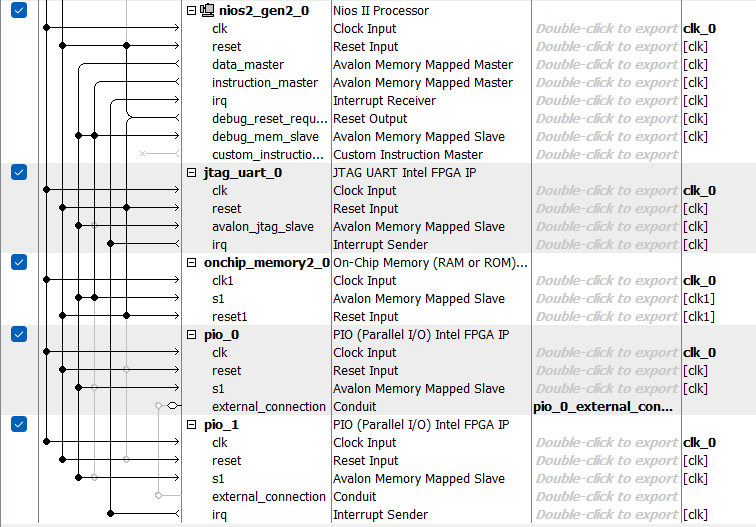
Description générée automatiquement

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

# CHENILLARD COMMANDE PAR INTERRUPTION LORS D’UN APPUI SUR BP :

On commence par modifier le qsys pour y ajouter un autre module PIO, incluant un Interrupt Sender.



Il faut ensuite recompiler le projet pour regénérer un fichier « chenillard.sopcinfo », et regénérer un dossier BSP.

Ensuite, on crée un fichier source « cheche\_IRQ.c » et on l’édite de la manière suivante :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquementVoici la routine d’interruption dans laquelle on exécute le chenillard en nous arrêtant à la dernière LED.

Enfin, on RAZ le flag d’interruption et on sort de la routine et on retourne dans le main().

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquementLa fonction main() consiste à définir un masque d’interruption pour se concentrer uniquement sur le bp key1 et on valide l’interruption en mettant le flag à 0.

Il faut également enregistrer le ISR à l’interruption correspondante.