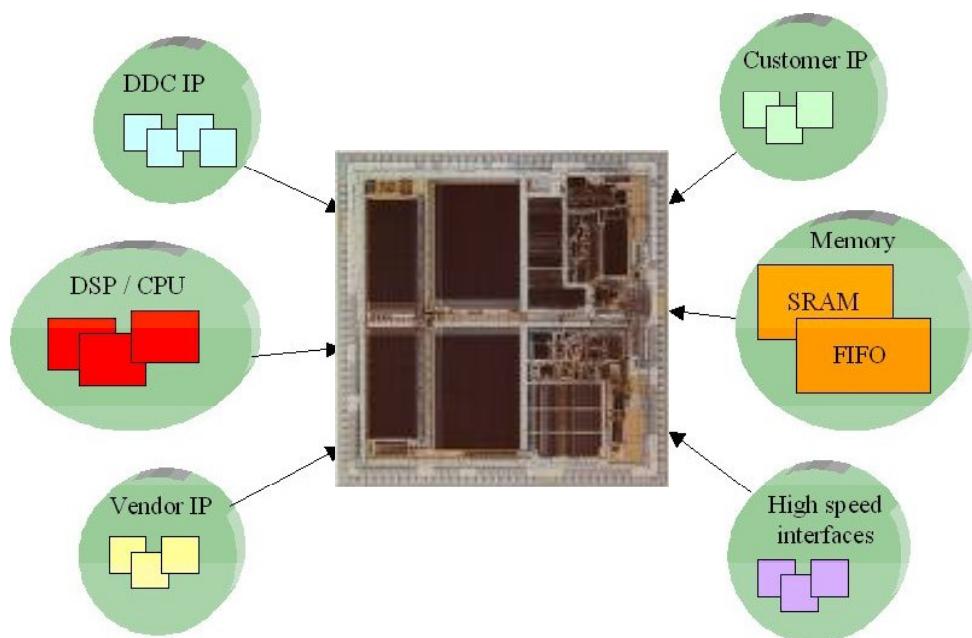


Mini-Projet

Radar 2D



Objectifs

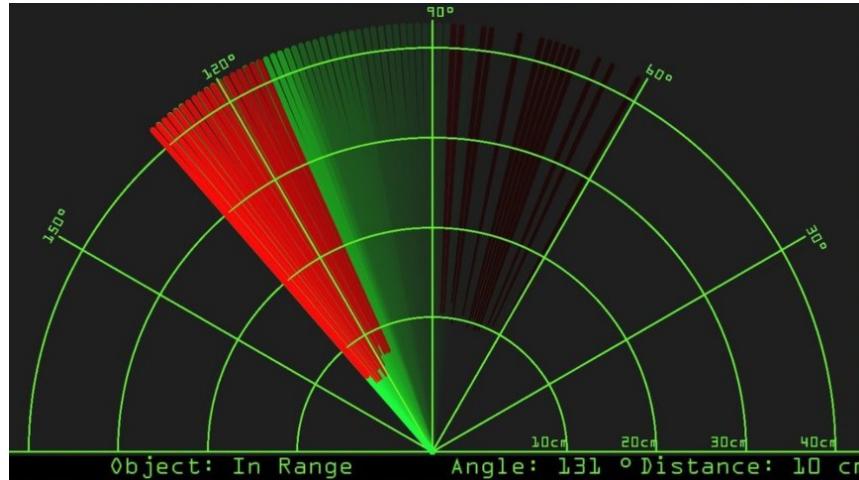
L'objectif de ce mini-projet est de cartographier une scène (Radar 2D) à l'aide d'un télémètre ultrason et un servomoteur.

L'organigramme de ce mini-projet est le suivant :



Dans un premier temps, il sera nécessaire de concevoir une IP télémètre à ultrasons. Cette étape commencera par le développement de la partie opérative (sans intégration avec le bus Avalon) et sa validation à l'aide d'une simulation sur Modelsim, suivie d'un test sur le FPGA pour consolider la vérification de cette partie. Par la suite, une interface avec le bus Avalon sera ajoutée pour intégrer l'IP dans un système basé sur Nios II en tant que périphérique personnalisé (Custom Peripheral).

Ensuite, une IP dédiée au contrôle d'un servomoteur devra être développée. Enfin, il sera nécessaire de concevoir une application permettant de cartographier une scène en faisant tourner le servomoteur tout en effectuant des mesures régulières grâce au télémètre à ultrasons.



Modalités d'évaluations :

Il faudra faire valider les différentes étapes par l'enseignant lorsque c'est demandé.

Il faut rédiger un compte rendu de projet qui répond aux différentes questions posées dans le sujet et qui illustre les différentes manipulations, codes (C et VHDL) et simulations que vous allez réaliser pour atteindre les objectifs du projet.

Etape 1 : Implémentation de l'IP télémètre ultrason HC SR04 (7 points)

1.1 Introduction : Présentation du capteur HC SR04

Caractéristiques

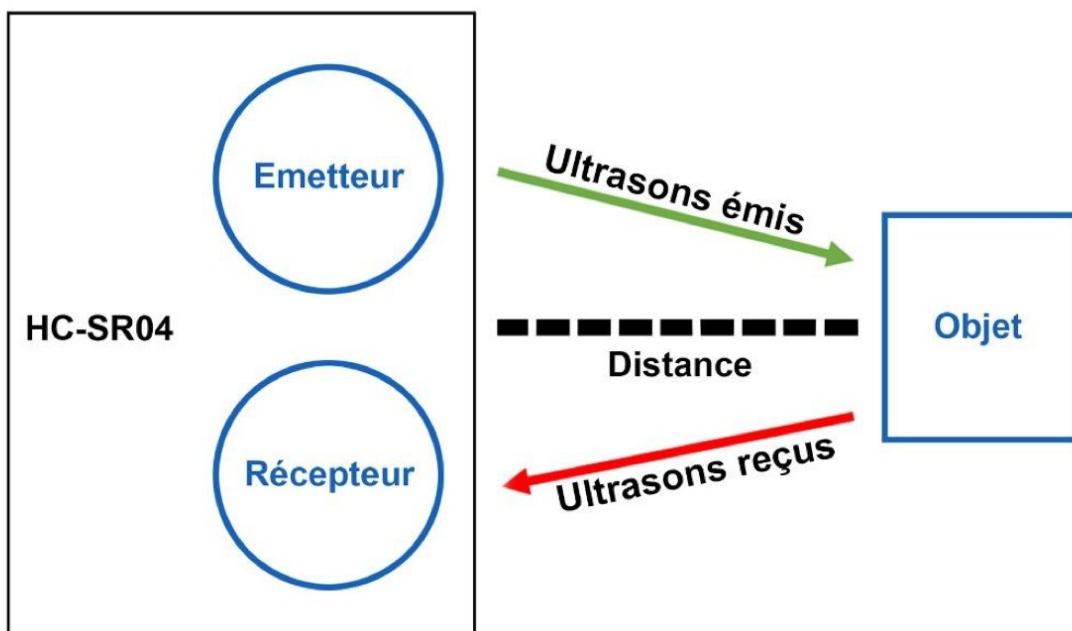
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15 °
- Largeur d'impulsion sur l'entrée de déclenchement : 10 µs (Trigger Input Pulse width)

Broches de connection

- Vcc = Alimentation +5 V DC
- Trig = Entrée de déclenchement de la mesure (Trigger input)
- Echo = Sortie de mesure donnée en écho (Echo output)
- GND = Masse de l'alimentation

Spécifications et limites

Paramètre	Min	Type	Max	Unité
Tension d'alimentation	4.5	5.0	5.5	V
Courant de repos	1.5	2.0	2.5	mA
Courant de fonctionnement	10	15	20	mA
Fréquence des ultrasons	-	40	-	kHz



Fonctionnement

Pour lancer une mesure, il suffit d'envoyer une impulsion **HIGH d'au moins 10 µs** sur l'entrée **Trig**. Le capteur émet alors **8 impulsions ultrasoniques à 40 kHz**, puis attend le retour de l'onde réfléchie.

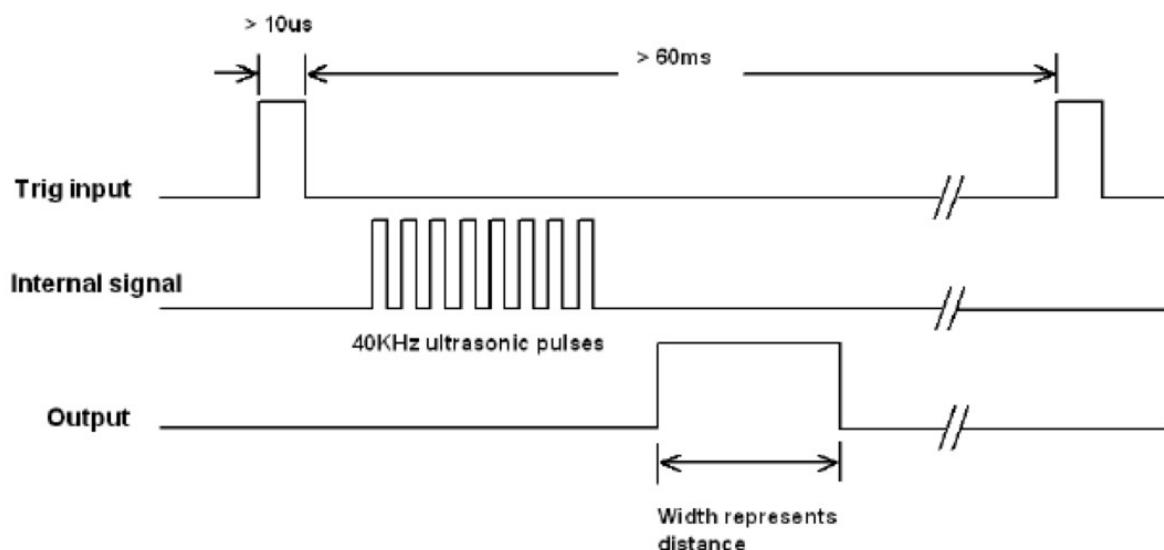
Environ **2 ms** après l'impulsion **Trig**, la sortie **Echo** passe à **HIGH** et reste à **1** tant que l'onde ultrasonore n'a pas été reçue par le récepteur.

Lorsque l'écho est finalement détecté, la sortie **Echo** repasse à **LOW** :

→ la durée pendant laquelle **Echo** reste **HIGH** est proportionnelle à la distance de l'obstacle.

Remarques :

- À grande distance, l'objet à déterminer doit présenter une surface minimale d'environ **0,5 m²**.
- En l'absence d'obstacle, la sortie **Echo** reste **HIGH** environ **40 ms**.
- Il faut attendre au moins **60 ms** entre deux mesures successives.



La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 340 m/s (ou par le temps de propagation, soit : $d = v \cdot t$ (distance = vitesse · temps)).

Il suffit de mesurer la durée de l'écho pour retrouver l'information sur la distance.

1.2 Travail à réaliser

1.2.1 Simulation de l'IP

Dans un premier temps, vous allez décrire la partie opérative de l'IP et la simuler dans Modelsim en écrivant un banc de test et un script de simulation.



Reporter dans le compte-rendu de projet les captures d'écrans de la simulation de l'IP avec les explications sur son fonctionnement et les tests réalisés.

Remarque : On choisit un reset (rst_n) actif à l'état bas pour faciliter la suite.

1.2.2 Test de l'IP sur carte

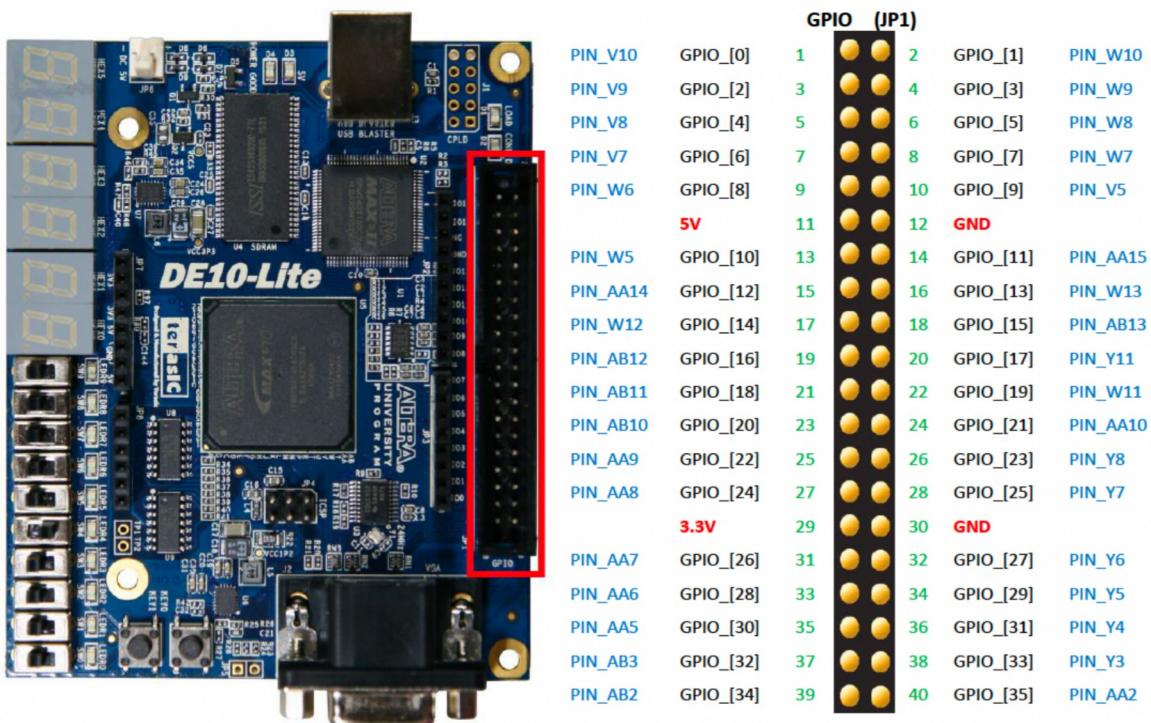
Avant de rajouter l'IP dans le SoPC architecturé autour du Nios2 à l'aide de Platform Designer, vous allez tester l'IP uniquement sur la partie FPGA (sans Nios2) pour vérifier qu'elle fonctionne bien en « standalone ».

Pour cela, vous allez partir d'un nouveau projet Quartus pour la carte DE10_Lite. Vous allez y rajouter le fichier source de votre télémètre qui sera considéré comme le Top Level.

Connecter les entrées-sorties de cette IP de la manière suivante :

Signaux de l'entité telemetre_us_HC_SR04	Signaux de la carte DE10_Lite (voir User Manual)	Pinout
Rst_n	KEY0	PIN_B8
CLK	MAX10_CLK1_50	PIN_P11
Trig	GPIO_[1]	PIN_W10
Echo	GPIO_[3]	PIN_W9
Dist_cm (9 downto 0)	LEDR[9..0]	Voir page 27 du DE10-Lite_User_Manual.pdf

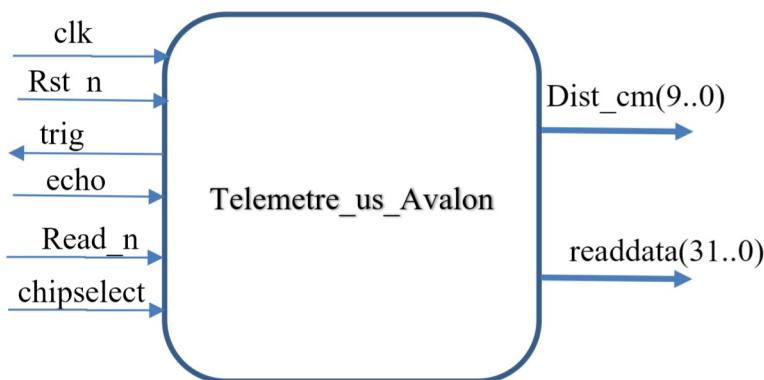
Pour alimenter le télémètre HC_SR04, il faut utiliser les pins 5V et GND du connecteur GPIO (JP1) (voir figure ci-dessous).



Testez votre IP sur la carte DE10_Lite. Vérifiez que les valeurs des LEDs évoluent en fonction de la distance d'un obstacle tenu devant le télémètre. Prenez des photos à rajouter dans votre compte rendu pour montrer que cette partie fonctionne bien.

1.2.3 Intégration de l'IP Télémètre dans Platform Designer

Rajouter l'interface Avalon (voir les signaux de l'entité ci-dessous) à votre IP Télémètre Ultrason afin de pouvoir le connecter au processeur NIOS II.



Dans un premier temps, tester votre IP par simulation pour vous assurer que l'interface Avalon fonctionne bien.

Reporter dans le compte-rendu de projet les captures d'écrans de la simulation de l'IP avec les explications sur les tests réalisés.

Ensuite, il faut intégrer votre IP dans Platform Designer (Custom Peripheral, cf TP2) en prenant le projet que vous avez réalisé pendant le TP1 qui intègre le processeur et l'ensemble des périphériques nécessaires.

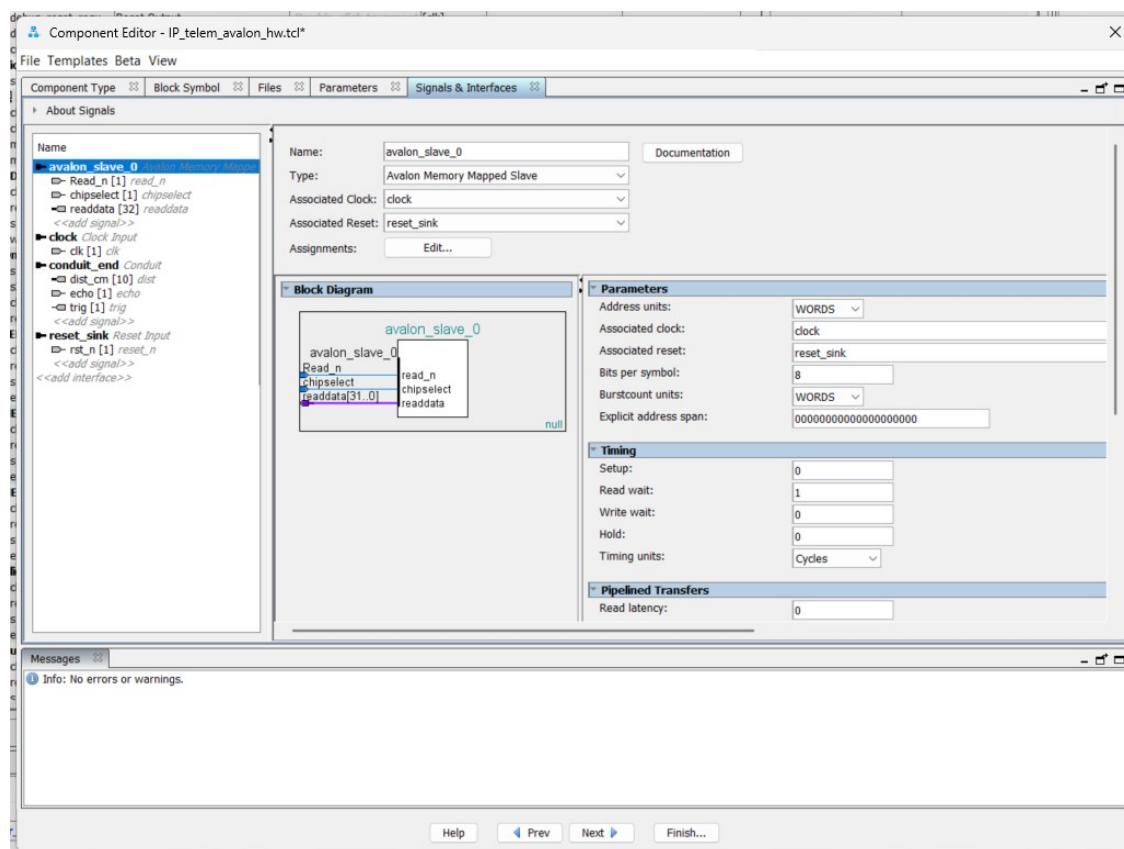
L'intégration d'une nouvelle IP dans un système SoC-FPGA existant se fait par l'intermédiaire d'un Custom Peripheral. Cela consiste à connecter les E/S de votre IP version Avalon avec les signaux disponibles sur le bus Avalon.

Dans Platform Designer, ajouter une IP de type Custom Peripheral (comme vu pendant les TP). Dans le menu **File → New component**, importez fichier VHDL de votre IP (onglet Files).

Connectez les signaux d'entrées / sorties de votre IP avec les signaux du bus Avalon (Onglet Signals ans Interfaces).

Il faut rajouter une interface **conduit_end** en cliquant sur <<add interface>> pour les signaux Trigger, Echo et Dist_Cm.

Votre configuration devrait sensiblement ressembler à la figure ci-dessous :



Générer le nouveau système dans Platform Designer.

Modifier le code VHDL du top-level pour instancier le SoC avec les signaux trig, echo et dist_cm que vous devrez connecter respectivement aux signaux GPIO(1), GPIO(3) et LEDR.

Vous pouvez trouver un modèle de l'instanciation dans le menu **Generate → Show Instantiation Template** de Platform Designer.

Synthétisez et implémentez le système sur le FPGA.

1.2.4 Programmation logicielle et test de l'IP

Maintenant que vous avez intégré la nouvelle IP Télémètre, vous allez programmer le processeur Nios2 pour qu'il acquiert une mesure de distance et qu'il l'affiche dans le terminal de la console du NIOS II.

Dans le code en langage C, il faut utiliser la fonction IORD(adresse de votre IP) ou plus concrètement la fonction IORD(IP_TELEMETRE_AVALON_BASE,0) ;

Dans un second temps, vous pouvez afficher la distance en cm sur les afficheurs 7 segments.

Prenez une photo à rajouter dans votre compte rendu pour montrer que cette partie fonctionne bien.

Etape 2 : Conception de l'IP Servomoteur (5 points)

Vous devez réaliser une IP qui permet de contrôler un servomoteur. Cette IP permettra de commander la position d'un servomoteur depuis le NIOS II.

2.1 Comment fonctionne un servomoteur

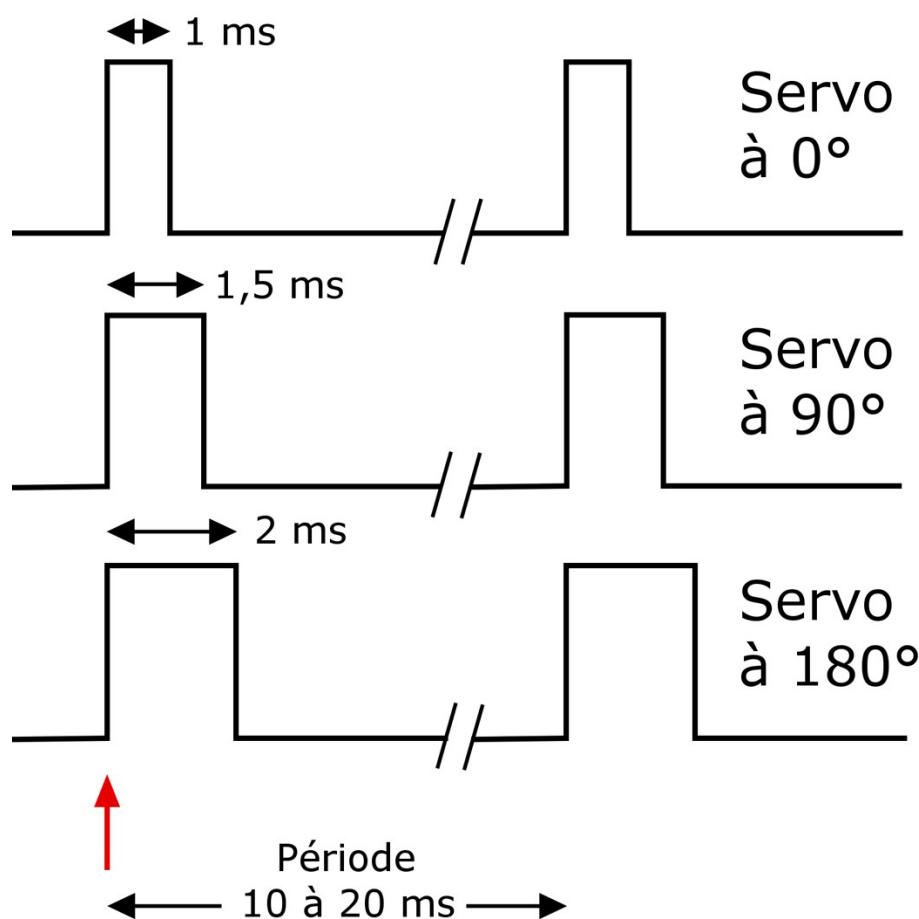
Le principe de base est assez simple. Il suffit d'envoyer une impulsion et c'est le temps que durera cette impulsion qui déterminera l'angle du servo-moteur. Ce temps d'impulsion est de quelques millisecondes et doit être répété à intervalle régulier (toutes les 10 à 20 ms).

Si le temps d'impulsion varie d'un fabricant à l'autre, les valeurs suivantes sont assez standard :

- 1 ms = 0 degré
- 1.50 ms = 90 degrés
- 2 ms = 180 degrés

Mais certains servomoteurs ont les paramètres suivants : 0,5 ms → 0 degré, 2,5 ms → 180 degrés

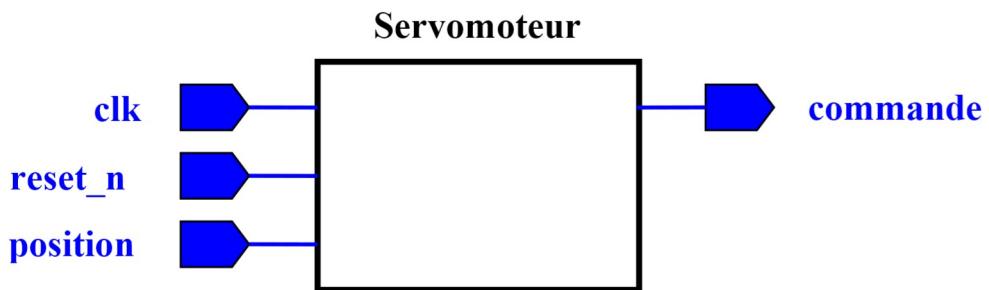
Le graphique ci-dessous montre la correspondance entre la longueur d'impulsion et l'angle du servo-moteur.



2.2 Conception de la partie opérationnelle du composant Servomoteur

Dans un premier temps vous allez réaliser un composant VHDL qui doit permettre de commander le servomoteur depuis les interrupteurs (switch) afin de vérifier la partie opérationnelle de votre IP.

L'entité à développer en VHDL devra être conforme aux spécifications données ci-dessous :



L'IP devra comporter en entrée :

- un signal position sur 8 ou 10 bit qui permet de choisir la position du servomoteur en degrés.
- un signal reset actif à l'état bas.
- une horloge système à 50 MHz.

L'IP devra comporter en sortie :

- un signal commande d'un bit connecté sur l'entrée du servo-moteur.

2.2.1 Développement de l'IP Servomoteur

Vous êtes libre de concevoir cette IP comme vous le souhaitez. C'est-à-dire, soit de manière comportementale, soit de manière structurelle, soit avec une machine à état.

Expliquez l'architecture de votre IP sur votre compte rendu.

2.2.2 Simulation et validation

Effectuer la simulation de l'IP avec Modelsim. Il est demandé d'écrire un banc de test et un script de simulation qui doit permettre de vérifier rapidement la validité du fonctionnement de l'IP.

Reportez vos résultats de simulation sur votre compte-rendu de projet.

2.2.3 Vérification sur la carte DE10_Lite

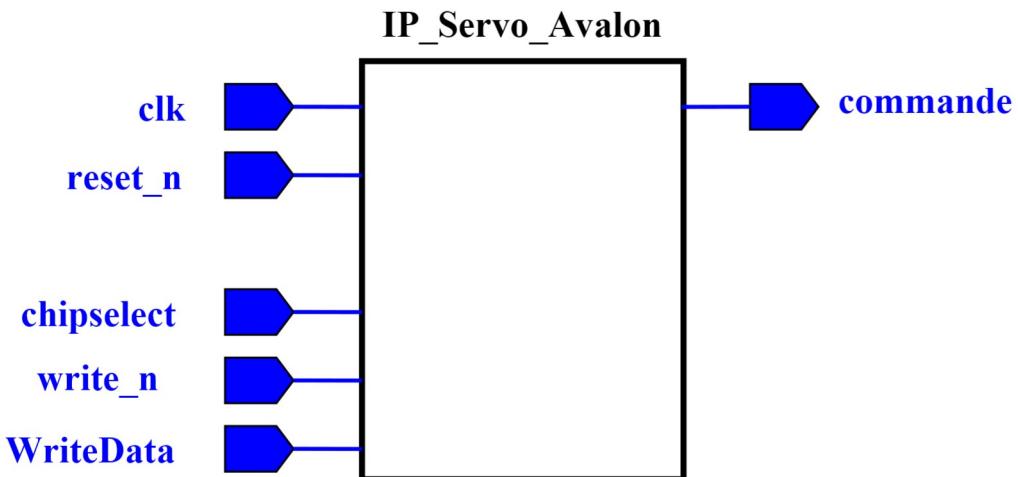
Tester votre IP sur la carte DE10-Lite. Pour cela, vous devez commander la position du servomoteur à partir des interrupteurs (switch). Configurez la sortie commande du servomoteur sur la broche PIN_V10 du FPGA, ce qui correspond à la broche GPIO_[0].

A l'aide d'un oscilloscope, vérifiez que le timing de la sortie commande correspond bien aux valeurs attendues et ensuite testez avec un servomoteur.

Faites vérifier cette étape par l'enseignant.

2.3 Extension de l'IP Servomoteur vers une version connectable au bus Avalon

Dans cette partie, nous allons modifier l'IP conçue auparavant afin que son interface prenne en compte les signaux disponibles sur le bus Avalon. L'entité à développer en VHDL devra être conforme aux spécifications données ci-dessous :



L'IP devra comporter en entrée :

- un signal chipselect : sélection du périphérique sur le bus Avalon
- un signal write_n : signal d'autorisation d'écriture dans les registres de l'IP
- un signal reset_n : remise à zéro des registres de l'IP actif à l'état bas.
- une horloge système à 50 MHz
- un signal WriteData sur 8, 16 ou 32 bits, qui permet d'écrire dans l'IP

L'IP devra comporter en sortie :

- un signal commande : entrée du servomoteur

2.3.1 Développement de l'IP

Apporter à la première version de l'IP les modifications nécessaires afin qu'elle prenne en compte les signaux du bus Avalon.

2.3.2 Simulation et validation

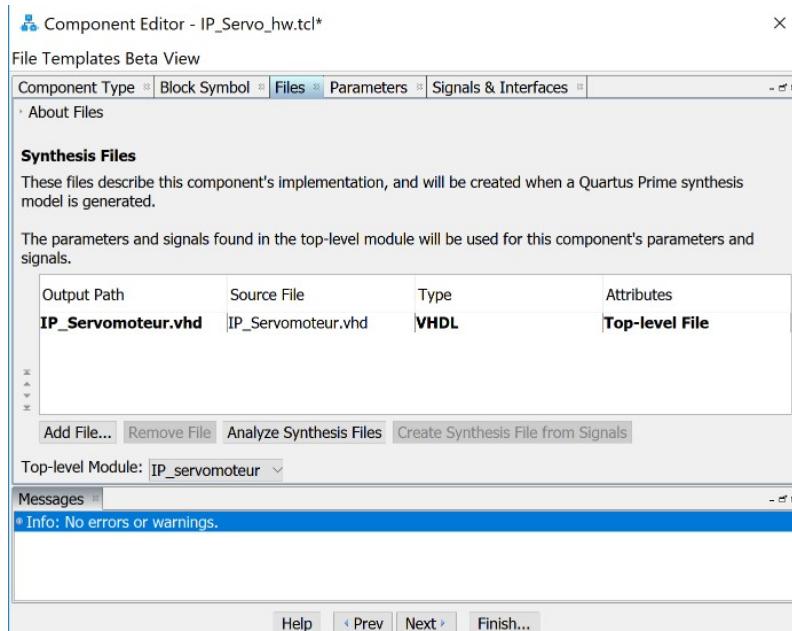
Effectuer les simulations du contrôleur avec Modelsim. Il est demandé d'écrire un banc de test et un script de simulation qui doivent permettre de vérifier rapidement la validité du fonctionnement de l'IP.

Reportez vos résultats de simulation sur votre compte-rendu de projet.

2.4 Intégration matérielle de l'IP Servomoteur

L'intégration d'une nouvelle IP dans un système SoC-FPGA existant se fait par l'intermédiaire d'un Custom Peripheral. Cela consiste à connecter les E/S de votre IP version Avalon avec les signaux disponibles sur le bus Avalon.

Dans Platform Designer, ajouter une IP de type Custom Peripheral (comme vu pendant les TP). Dans le menu **File → New component**, importez fichier VHDL de votre IP (onglet Files).



Connectez les signaux d'entrées / sorties de votre IP avec les signaux du bus Avalon (Onglet Signals ans Interfaces).

Il faut rajouter une interface **conduit_end** en cliquant sur <<add interface>> pour le signal de Commande.

Rajouter l'IP Servomoteur ainsi créé dans votre projet et connecter les signaux en prenant pour exemple l'IP télémètre.

Générer le nouveau système. Modifier le code VHDL du top-level, synthétisez et implémentez le système sur le FPGA.

2.5. Programmation logicielle et test de l'IP Servomoteur

Maintenant que vous avez intégré la nouvelles IP Servomoteur, vous allez programmer le processeur pour qu'il commande le servomoteur en position depuis les interrupteurs (Switch). Pour cela, vous pouvez prendre comme point de départ le programme donné pour l'IP télémètre.

Consignez votre programme dans votre Compte Rendu.

Prenez une photo du résultat à mettre dans votre compte-rendu.

Etape 3 : Affichage des obstacles (3 points)

Dans cette étape il est demandé de détecter les obstacles en faisant tourner le servomoteur sur 180 ° tout en faisant des mesures régulières avec le télémètre ultrason.

Affichez les résultats dans le terminal de NIOS2 SBT.

Par exemple sous la forme suivante :

1° -> 60 cm

2° -> 61 cm

3° -> 60 cm

4° -> 61 cm

...

Consignez votre programme et vos résultats dans votre Compte Rendu.

Faites vérifier cette étape par l'enseignant.

.