

# Document d'aide pour l'utilisation de l'IP module\_RS232 pour réaliser une communication UART sur FPGA

## 1- Utilisation d'une IP dans un projet Vivado

- (conseillé) Créez un dossier "IPs" sur votre home qui va contenir toutes les IP avec lesquelles vous allez travailler (au cas ou vous en utiliseriez d'autres).
- Déplacez le dossier (décompressé !) de l'IP module\_RS232 (i.e. le dossier contenant a minima un sous-dossier "src", un sous dossier "xgui" et un fichier "component") dans le dossier "IPs" créé à l'étape précédente.
- Dans Vivado ouvrez le projet dans lequel vous voulez intégrer l'IP
- Allez dans "Tools" puis "Settings..."
- Développez le menu "IP" dans la colonne de gauche puis cliquez sur "Repository"
- cliquez sur + dans la fenêtre "IP repositories" puis sélectionnez le dossier "IPs" (toutes les IP contenus dans ce dossier seront disponibles dans votre projet, inutile de refaire cette manip à chaque fois).
- Validez avec "Select" puis "OK"
- Pour intégrer une IP dans un design schématique, il suffit de cliquer sur + puis taper le nom de l'IP dans la barre de recherche et valider.

## 2- L'IP Module RS232

Elle contient toutes les descriptions nécessaires (basiquement double registre à décalage pilotés par une horloge paramétrable) pour réaliser une interface UART élémentaire.

### a - Paramètres de communication

Le composant est paramétré pour une communication 115200 8N1

- Baud Rate 115200 Bits par seconde environ
- 8 bits de données
- Pas de bit de parité
- Un bit de Stop

### b - Description des entrées

- uart\_rxd : signal d'entrée série RX
- RAZ : signal de reset (à placer au niveau logique haut pour permettre la communication)
- EN : signal de déclenchement de la transmission (transmission des 8 bits contenus dans DataTX en série vers la sortie uart\_txd)
- DataTX[7:0] : signal de données (8bits) qui seront transmises à partir de l'activation de EN
- CLOCK : signal d'horloge cadencant le composant (ce signal doit être à la fréquence 125MHz pour permettre une transmission avec un baud rate d'environ 115200 bit/s)

### c - Description des sorties

- `uart_txd` : signal de sortie série TX
- `DataRX[7:0]` : signal de données (8bit) reconstitué lors de la réception d'un paquet.

## 3 - Le composant transmissionDonnee

Ce composant (fourni sous forme d'une description VHDL modifiable à votre guise) permet :

- Un échantillonnage des données à transmettre (période d'échantillonnage définie par le paramètre `Period`)
- Un découpage des données en paquets de 8bits pour l'envoi via le module RS232 (un temps mort de `periodoctet` est respecté entre l'envoi successif de deux octets)
- Une transmission de données de taille paramétrable (paramètre `DataSize`) pouvant aller jusqu'à 5 octets par données.
- Le pilotage du début de la transmission (via l'entrée `debutTransmission`) et de la fin (via le paramètre `Ndata`) quand un nombre suffisant de données a été transmise (afin de ne pas saturer le terminal par la suite)

### a - Description des paramètres

- `Ndata` : nombre de données qui seront transmises avant l'arrêt de transmission (valeur entière sur 32 bits)
- `DataSize` : taille (en octet) de chaque donnée à transmettre (valeur entière inférieure ou égale à 5)
- `Period` : Période (en us) d'échantillonnage du signal d'entrée pour la transmission (valeur entière sur 32 bits)
- `Periodoctet` : temps mort (en us) entre la transmission successive de deux octets de la même donnée. (valeur entière sur 32 bits)

### b - Description des entrées

- `H` : Horloge système à 125MHz
- `T1us` : signal périodique permettant de maîtriser les temporisations décrites précédemment. Ce signal doit présenter un niveau logique haut durant exactement 8ns toutes les microsecondes.
- `DébutTransmission` : signal qui, par son passage à 1 va initier une transmission de `NData` données échantillonnées.
- `data[39:0]` : signal qui sera échantillonné pour la transmission -> prévu pour 5 octets par donnée (transmission de 5 caractères codés en ASCII)

### c - Description des sorties

- TX[7:0] : signal de donnée 8 bits à envoyer via le module RS232
- validTx : signal de déclenchement de la transmission série de la donnée 8 bits TX.

## 4 - Le composant DecodBin\_ASCII

Ce composant (non fourni, à vous de jouer...) doit permettre de convertir une donnée (valeur entière sur 32 bits ou autre chose à votre guise) en un paquet de 5 codes ASCII représentant la donnée sur 5 digits (ou plus judicieusement un octet pour le signe, 3 pour la valeur et un pour le retour à la ligne) afin de permettre un affichage immédiat des données sur un terminal série coté PC.

## 5- Exemple partiel de design combinant ces éléments

