

INF3500 - Conception et réalisation de systèmes numériques

Labo 5 - Communication sérielle
26 mars 2020



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE

par Olivier Dion

Table des matières

| | |
|-------------------------------------|----------|
| 1 Objectifs | 2 |
| 2 Conseils | 3 |
| 3 Contexte | 4 |
| 4 Module UART | 5 |
| 4.1 Livrable | 5 |
| 5 Banc d'essai | 6 |
| 5.1 Récepteur | 6 |
| 5.2 Transmetteur | 6 |
| 5.3 Livrable | 6 |
| 6 Synthèse et implémentation | 7 |
| 6.1 Terminal | 7 |
| 6.2 Livrable | 7 |
| 7 Rapport | 8 |
| 8 Barème | 9 |

1. Objectifs

Ce laboratoire a pour objectif de mettre en pratique vos connaissances en circuits séquentiels pour implémenter le protocole de communication UART.

2. Conseils

Avant de commencer, voici quelques conseils.

- Utiliser **Git** (*GitBash* sur votre poste de travail) pour sauvegarder l'historique de vos travaux. Indexer seulement les fichiers code sources et autre fichier **non binaire**.
- Travaillez dans le dossier `C:\TEMP` et poussez vos travaux sur un serveur. La génération du *bitstream* sera plus rapide.
- Écrivez toujours le nom des autrices et matricules étudiant aux débuts de vos fichiers en commentaire. Ajoutez optionnellement une licence pour votre code source.
- La synthèse et l'implémentation sont des procédés assez longs. Imaginez un programme `C++` qui compile de 10 à 30 minutes. Relisez-vous pour vous sauver du temps !
- Dans le doute, consultez la **FAQ** du cours (`faq.pdf`).
- Écrivez vos rapports en **LaTeX**. La qualité de ceux-ci sera supérieure.
- Lisez le document en entier avant de commencer à travailler.

3. Contexte

En informatique, de très nombreux protocoles de communication (USB, Sata, PCI Express, Ethernet, UART, etc.) utilisent un lien de transmission série entre un émetteur et un récepteur. Lorsqu'une transmission série est utilisée, une unité d'information (dans ce laboratoire on considère un bit) est envoyée par cycle d'horloge vers le récepteur. Le récepteur réassemble ensuite l'information reçue en un mot, pour l'utiliser sur un bus de donnée ayant une largeur de plusieurs bits. Ainsi, ces protocoles nécessitent un module émetteur faisant la conversion lien parallèle (bus de plusieurs bits) vers un lien sériel, et un récepteur faisant l'opération inverse (conversion lien sériel vers lien parallèle).

L'UART est un protocole de communication bidirectionnelle, asynchrone, permettant d'échanger de l'information entre un récepteur et un émetteur. L'échange d'information est réalisé au travers d'un lien de transmission série. L'information échangée entre l'émetteur et le récepteur est transmise à raison d'un bit par cycle d'horloge. La période d'horloge est l'inverse de la vitesse du lien de connexion UART entre l'émetteur et du récepteur, qui est exprimé en Baud/seconde. Afin que le récepteur puisse détecter qu'une information lui est transmise, le protocole UART spécifie un format sous lequel l'information doit être envoyée.

Consultez le web afin de vous familiariser avec le protocole UART. Sans une bonne compréhension du protocole UART, il n'est pas possible de réaliser ce laboratoire. À la suite de vos lectures, vous devez être capable de répondre aux questions suivantes.

1. Qu'est-ce qu'une trame UART ?
2. Où le mot transmis par l'émetteur est-il inséré au sein de la trame UART ?
3. Comment le récepteur UART peut-il détecter que l'émetteur est en train de lui envoyer un nouveau mot ? Quelle est la valeur du premier bit transmis par l'émetteur lors de l'envoi d'un message ?
4. Comment le récepteur UART peut-il détecter que l'envoi d'un mot par le transmetteur est terminé ? Est-ce que la longueur du mot envoyée est fixe ?

4. Module UART

Dans le cadre de ce laboratoire, le protocole UART à 8 bits de données avec 1 STOP bit et la parité à impair est utilisé.

Vous devez implémenter le récepteur ainsi que le transmetteur pour le protocole UART. Compléter le fichier `uart.vhd`. Commencez par le récepteur, le transmetteur étant beaucoup plus facile à faire.

Le récepteur (RX) et le transmetteur (TX) possèdent deux états machines différents et sont donc dans deux procédures différentes. Notez que le *reset* est synchronisé avec l'horloge.

Vous devez connecter les pins qui sont marqués en commentaire par (connect me) à un signal interne. Vous devez aussi trouver la valeur pour *CLK_DIV*.

4.1 Livrable

- `uart.vhd`
- Vos machines à états dans votre rapport

5. Banc d'essai

Vous devez par la suite faire un banc d'essai pour les deux états machine de votre module.

5.1 Récepteur

Vous devez tester les cas suivants pour le récepteur. Les messages sont tous sur 1 caractère (8 bits), sauf si indiqué.

1. Un message correct (sans erreur)
2. Un message correct de 16 bits
3. Un message avec une mauvaise condition STOP
4. Un message avec une mauvaise parité

5.2 Transmetteur

Vous devez juste transmettre un message et vérifier que celui-ci est bon.

5.3 Livrable

Votre banc d'essai `uart-tb.vhd`. Mettez aussi des figures de votre simulation dans votre rapport.

6. Questions

- (1) Dans quel cas le calcul de la parité d'une trame ne marche pas ? i.e. il n'y a pas d'erreur dans la parité, mais il y a une erreur dans la trame.
- (1) Pour quelle raison le récepteur UART doit faire un délai après la réception de la condition START ?

7. Rapport

Vous devez écrire un rapport selon les directives dans le fichier `rapport.pdf`.

Seuls les fichiers de type *pdf*, *DjVu* et *ps* sont acceptés pour le rapport. Aucun fichier de type Word (doc, docx) n'est accepté.

Assurez-vous d'inclure tous les fichiers listés dans les sections `Livable(s)`.

8. Barème

| Critères | Points |
|--|--------|
| Module UART | 8 |
| Banc d'essai | 8 |
| Questions | 2 |
| Rapport : Présentation et qualité de la langue | 2 |
| Total | 20 |