

Unité de transmission/réception série asynchrone – version sans bit de parité

Programmation en VHDL : Informatique et Mathématiques
Appliquées – Parcours Logiciel

Plan de la présentation

1 Description de l'UART à réaliser (2ème partie)

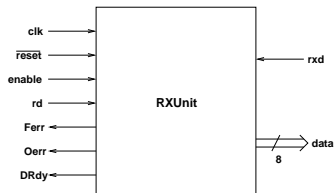
- L'unité de réception
- L'UART : l'assemblage du tout

Plan de la présentation

1 Description de l'UART à réaliser (2ème partie)

- L'unité de réception
- L'UART : l'assemblage du tout

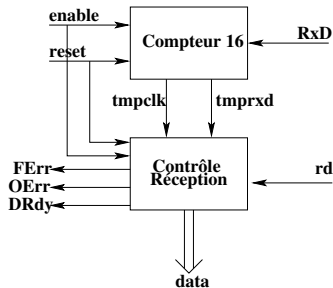
Interface de l'unité de réception



```
entity RxUnit is
  port (
    clk, reset  : in  std_logic;
    enable      : in  std_logic;
    rd          : in  std_logic;
    rxn         : in  std_logic;
    data        : out std_logic_vector(7 downto 0);
    Ferr, Oerr, DRdy : out std_logic);
end RxUnit;
```

- Ferr, OErr : Erreurs de réception
- DRdy : Signal qui informe qu'une donnée a été reçue correctement

Composition de l'unité de réception

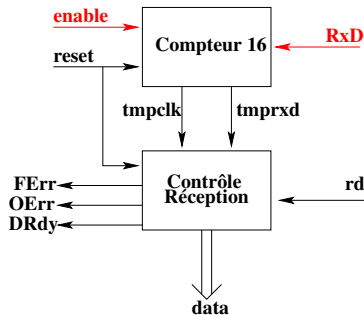


L'unité de réception se compose de :

- un compteur 16 qui fournit, tous les 16 tops de l'horloge de réception enableRX, un signal tmpclk et un signal tmprxd correspondant à la valeur de RxD récupérée à 8 tops de l'horloge de réception
- l'unité de contrôle de la réception récupère un à un les bits reçus via le signal tmprxd et informe s'il y a une erreur ou non

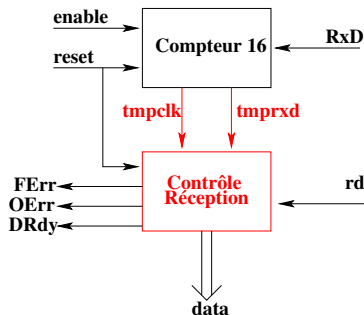
Fonctionnement de l'unité de réception

1 on attend la réception du bit de start sur RxD



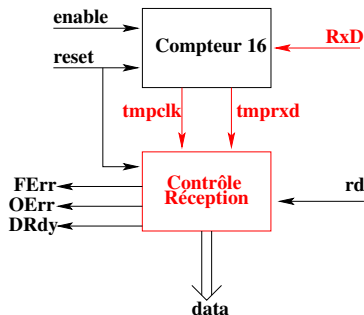
Fonctionnement de l'unité de réception

- 1 on attend la réception du bit de start sur RxD
- 2 la réception du bit de start déclenche l'horloge tmpclk qui simule l'horloge d'émission (soit 16x l'horloge de réception enableRX). Le bit de start est transmis par le signal tmprxd sur un front montant de tmpclk



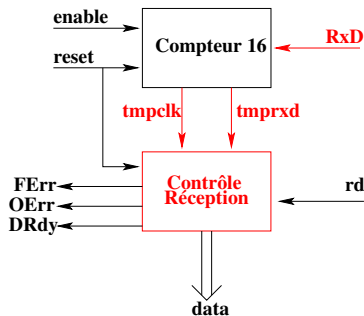
Fonctionnement de l'unité de réception

- 1 on attend la réception du bit de start sur RxD
- 2 la réception du bit de start déclenche l'horloge tmpclk qui simule l'horloge d'émission (soit 16x l'horloge de réception enableRX). Le bit de start est transmis par le signal tmprxd sur un front montant de tmpclk
- 3 on récupère les 8 bits de données sur tmprxd à chaque top de tmpclk



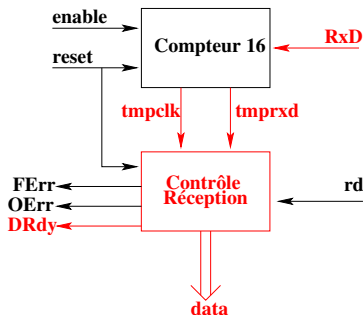
Fonctionnement de l'unité de réception

- 1 on attend la réception du bit de start sur RxD
- 2 la réception du bit de start déclenche l'horloge tmpclk qui simule l'horloge d'émission (soit 16x l'horloge de réception enableRX). Le bit de start est transmis par le signal tmprxd sur un front montant de tmpclk
- 3 on récupère les 8 bits de données sur tmprxd à chaque top de tmpclk
- 4 on récupère le bit de stop



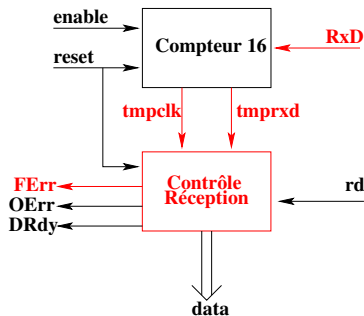
Fonctionnement de l'unité de réception

- 1 on attend la réception du bit de start sur RxD
- 2 la réception du bit de start déclenche l'horloge tmpclk qui simule l'horloge d'émission (soit 16x l'horloge de réception enableRX). Le bit de start est transmis par le signal tmprxd sur un front montant de tmpclk
- 3 on récupère les 8 bits de données sur tmprxd à chaque top de tmpclk
- 4 on récupère le bit de stop
 - si le bit de stop est correct, on positionne le bit DRdy et la donnée en sortie



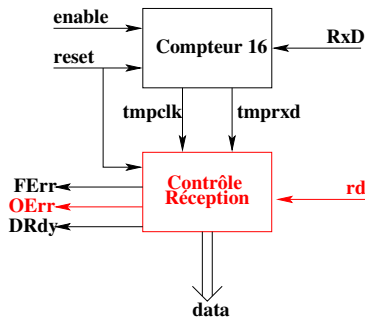
Fonctionnement de l'unité de réception

- 1 on attend la réception du bit de start sur RxD
- 2 la réception du bit de start déclenche l'horloge tmpclk qui simule l'horloge d'émission (soit 16x l'horloge de réception enableRX). Le bit de start est transmis par le signal tmprxd sur un front montant de tmpclk
- 3 on récupère les 8 bits de données sur tmprxd à chaque top de tmpclk
- 4 on récupère le bit de stop
 - si le bit de stop est correct, on positionne le bit DRdy et la donnée en sortie
 - sinon on positionne le bit FErr



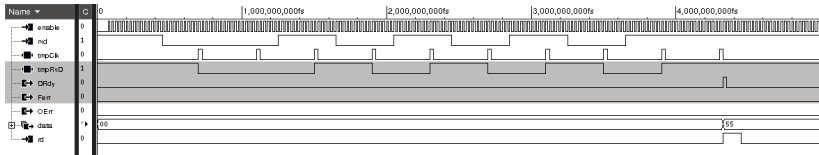
Fonctionnement de l'unité de réception

- 1 on attend la réception du bit de start sur RxD
- 2 la réception du bit de start déclenche l'horloge tmpclk qui simule l'horloge d'émission (soit 16x l'horloge de réception enableRX). Le bit de start est transmis par le signal tmprxd sur un front montant de tmpclk
- 3 on récupère les 8 bits de données sur tmprxd à chaque top de tmpclk
- 4 on récupère le bit de stop
 - si le bit de stop est correct, on positionne le bit DRdy et la donnée en sortie
 - sinon on positionne le bit FErr
- 5 au top de enableRX suivant si la donnée n'est pas récupérée, le bit OErr est positionné



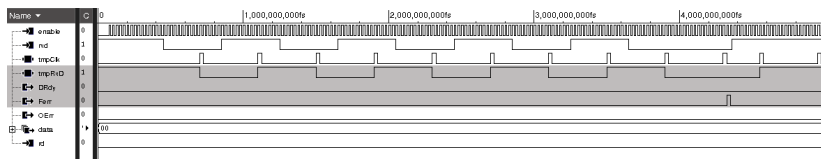
Test de l'unité de réception

■ Réception correcte du caractère 0x55 :



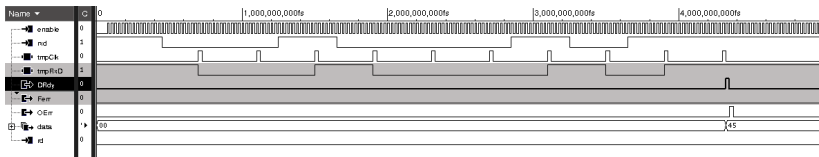
Test de l'unité de réception

■ Réception erronée du caractère 0xAA, erreur bit stop :



Test de l'unité de réception

- Réception erronée du caractère 0xAA, la donnée n'est pas lue :



L'UART : l'assemblage du tout

- L'UART est composée des 4 unités (clkUnit, ctrlUnit, RxUnit, TxUnit) :
 - utilisation de la vue structurelle
 - instantiation des 4 unités (à faire)
- l'unité d'émission TxUnit nécessite un signal ld de chargement, ce signal contrôle l'écriture dans le buffer d'émission :

```
ecriture <='1' when cs='0' and wr='0' else '0';
```

- le processeur lit les données de l'unité de réception ou le registre de contrôle en paramétrant l'adresse de lecture :

```
lecture <='1' when cs='0' and rd='0' else '0';  
data_out <= donnees_recues when lecture='1' and addr="00"  
           else registre_controle when lecture='1' and addr="01"  
           else "00000000";
```