

Travaux pratiques : initiation au langage VHDL par l'étude d'un émetteur et d'un récepteur Manchester.

L'objectif de cette séance est de se familiariser avec le langage VHDL par l'implantation de quelques fonctions simples. Nous prendrons pour support d'étude un émetteur récepteur de transmission d'un signal numérique, codé Manchester. L'ensemble sera implanté au sein d'un CPLD.

Nous utiliserons le logiciel Max+plus II comme outils de synthèse, logiciel avec lequel nous supposerons une certaine familiarisation acquise lors de séances précédentes.

Afin de simplifier les tests de l'ensemble, la sortie de l'émetteur et l'entrée du récepteur implantés au sein du même CPLD seront reliées de manière interne. La partie électronique pure sera occultée au profit de la partie programmation. Il reste cependant possible de synthétiser l'émetteur et le récepteur au sein de deux circuits différents (la carte de développement DLP comprend un Max7000 et un Flex10K, ou encore utiliser deux cartes), de réaliser les circuits d'interface de l'attaque de ligne etc...

Ce sujet comprend quatre parties :

- la première présente le problème et justifie l'utilisation du code Manchester ;
- la seconde étudie les fonctions à implanter ;
- la troisième propose une synthèse de l'émetteur et du récepteur en utilisant des composants prédéfinis au sein du logiciel Max+plus II ;
- enfin la dernière partie propose de reprendre la synthèse, mais cette fois en VHDL.

Le lecteur familiarisé avec ce type de transmissions numériques pourra éventuellement passer directement de la seconde à la dernière partie

Remarque : les solutions proposées lors de cette séance aboutissent parfois à la violation de la règle de base des systèmes synchrones, qui est qu'un tel système ne doit avoir qu'une seule horloge. Notre but n'était pas ici de faire un système performant techniquement, mais un système permettant un passage simple d'une description « classique » vers une description VHDL. Pour la même raison, les sous-ensembles sont parfois simplifiés à l'extrême

1. Transmissions numériques et codage Manchester

Le code utilisé le plus couramment pour représenter une donnée numérique est le code NRZ :

- un NL1 (niveau logique 1) est représenté par 5 V par exemple
- un NL0 (niveau logique 0) par 0 V (codage NRZ unipolaire) ou -5 V (codage NRZ antipolaire).

Ce code simple est cependant inadapté pour des transmissions sur des distances supérieures à quelques mètres.

En effet la boucle créée par les fils aller et retour entre émetteur et récepteur, et surtout celle créée par le câble de transmission (ensemble des fils aller et retour) et la terre, réalisent une antenne sensible aux variations du champ électromagnétique ambiant. Il s'ensuit alors la superposition d'un signal parasite sur le signal utile rendant l'interprétation de ce dernier impossible.

Une solution couramment envisagée pour pallier ce problème consiste à ouvrir ces boucles au moyen de transformateurs électromagnétiques.

Ces derniers ne laissent cependant pas passer les signaux continus, et dans le cas d'un signal NRZ, une suite de niveaux logiques identiques introduira forcément la présence d'une composante continue.

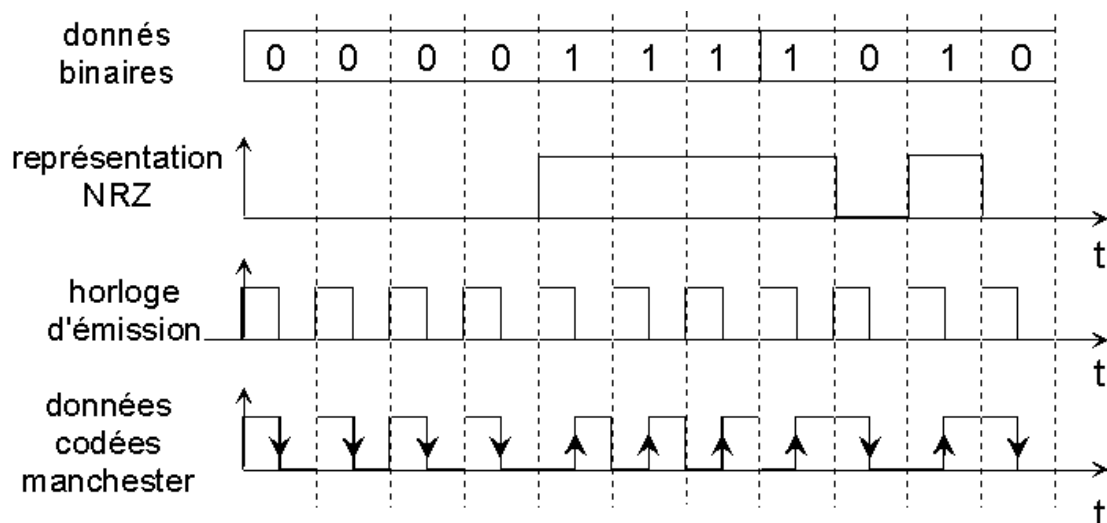
Dans certaines liaisons, on utilise également la même ligne que celle sur laquelle sont transmises les données, pour faire transiter un signal continu destiné à alimenter les étages intermédiaires régénérant le signal. Les données ne doivent donc pas émettre d'énergie basse fréquence afin que l'information ne soit parasitée par le signal d'alimentation.

Enfin, le récepteur a souvent besoin de récupérer le rythme d'émission pour effectuer un décodage correct de l'information lui arrivant ; l'opération est rendue très difficile avec le code NRZ, en particulier dans le cas d'une suite de données identiques.

L'idée du codage Manchester ou biphasé consiste à représenter les informations logiques, non pas par des niveaux (5 V et 0 ou -5 V), mais par des transitions :

- une NL1 par un front descendant au milieu de la période d'horloge ;
- un NL0 par un front montant au milieu de la période d'horloge.

Le choix du type de front est purement conventionnel et nous utilisons en fait dans notre liaison un code Manchester inversé ou un niveau logique 1 est représenté par un front montant en milieu de période (et inversement pour le NL0) comme le montre la figure suivante.



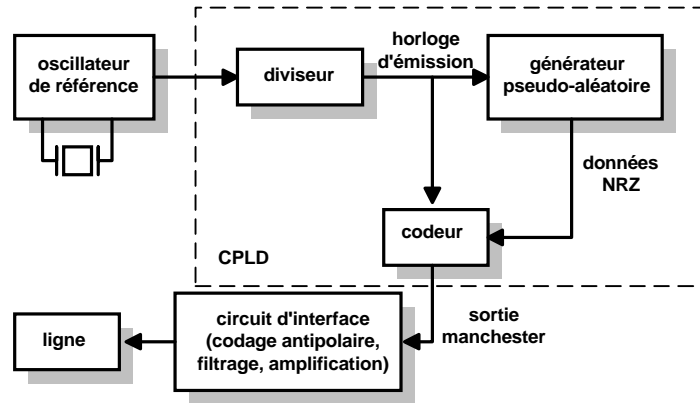
L'utilisation d'un codage Manchester antipolaire (5 V et -5 V par exemple) supprime alors toute composante continue.

Dans la suite l'étude sera faite pour un CPLD alimenté en 0/5 V ; le codage sera donc unipolaire et nous supposons un circuit d'interface (non étudié) permettant la liaison à la ligne. Ce circuit d'interface permettra la transition vers un codage antipolaire (pour l'émetteur) ou depuis un codage antipolaire (pour le récepteur).

2. Préparation

2.1. Etude de l'émetteur

Le schéma synoptique de l'émetteur est donné par la figure suivante :



A partir d'un signal carré issu d'un oscillateur à quartz servant de référence temporelle stable, on obtient le signal d'horloge d'émission par une division par 16 de la fréquence.

Dans notre cas de figure, nous ne disposons pas d'un système numérique ayant réellement des données à transmettre, aussi nous le simulerons par un générateur pseudo-aléatoire.

Ce type de générateur permet de générer à partir de l'horloge d'émission une séquence codée NRZ. Cette dernière permettra de tester à la fois l'émetteur et le récepteur avec un signal plus proche de la réalité qu'une suite de niveaux identiques ou qu'une alternance de 1 et de 0.

A partir de ce signal et de l'horloge d'émission, le codeur réalise la translation du code NRZ vers le code Manchester.

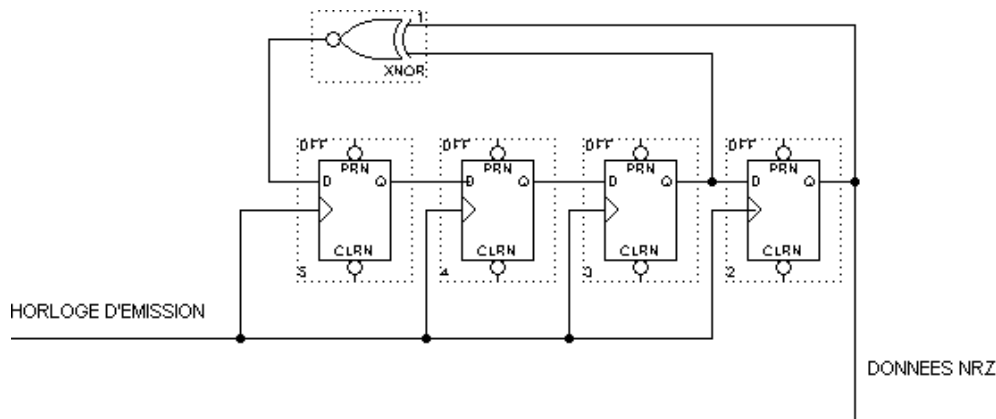
Le circuit d'interface (non étudié) permet ensuite d'attaquer la ligne.

2.1.1. Le codeur

*Le codeur est-il une fonction combinatoire ou séquentielle ?
Donner cette fonction.*

2.1.2. Le générateur pseudo-aléatoire.

Ce bloc permet de générer à partir de N bascules D montées en registre à décalage une suite périodique de $2^N - 1$ bits codés NRZ. Plus le nombre de bascules est important et plus la suite numérique paraît aléatoire ; étant cependant périodique, elle est appelée pseudo-aléatoire. L'entrée du registre à décalage est une combinaison par un ou exclusif (complémenté ou non) de la sortie du registre et d'une ou plusieurs bascules intermédiaires dont le rang dépend de N.



Représenter les chronogrammes des différents signaux de sortie des bascules en fonction de l'horloge d'entrée (on supposera que toutes les bascules ont leur sortie à 0 à l'origine des temps).

Vérifier que nous avons bien une suite pseudo-aléatoire de 2^N-1 bits quelle que soit la sortie choisie.

Dans ce type de générateur il existe toujours au moins un état de sortie qui bloque le système. Que se passe-t-il dans notre exemple si toutes les sorties de bascules sont au NL1 ?

2.1.3. Le diviseur

Nous utiliserons une fonction prédéfinie par le logiciel pour réaliser ce diviseur.

Dans le cas où nous aurions à le réaliser à l'aide de bascules D, proposer un schéma structurel.

3. Etude du récepteur

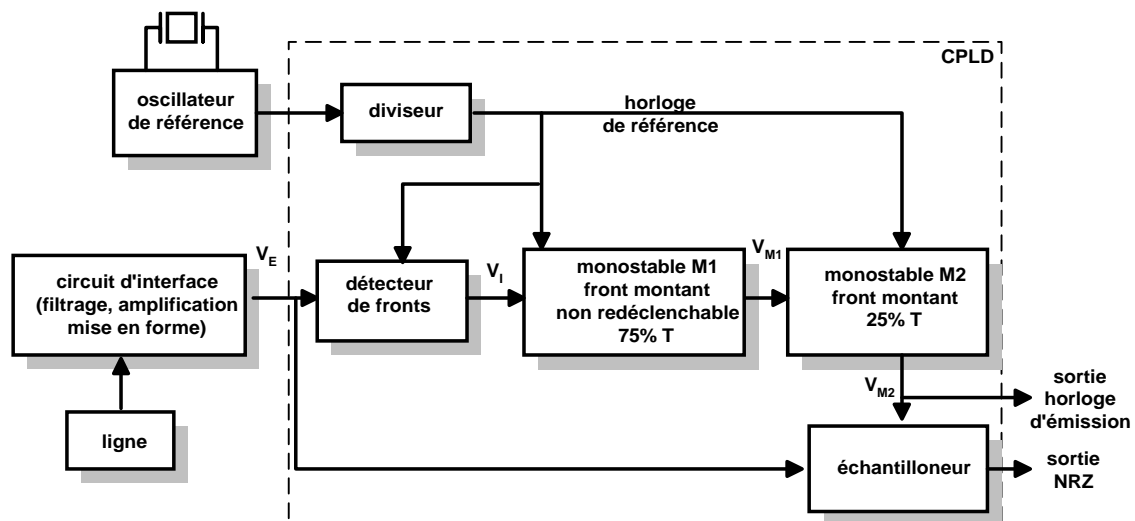
Pour décoder le signal reçu, nous allons utiliser le fait qu'au milieu de chaque période de l'horloge d'émission, un front montant (pour un NL1 de la donnée NRZ) ou descendant (pour un NL0 de la donnée NRZ) représente la valeur cherchée.

Il suffira donc d'aller lire le signal reçu après ce front pour avoir la valeur NRZ, aux trois quarts de la période de l'horloge d'émission par exemple.

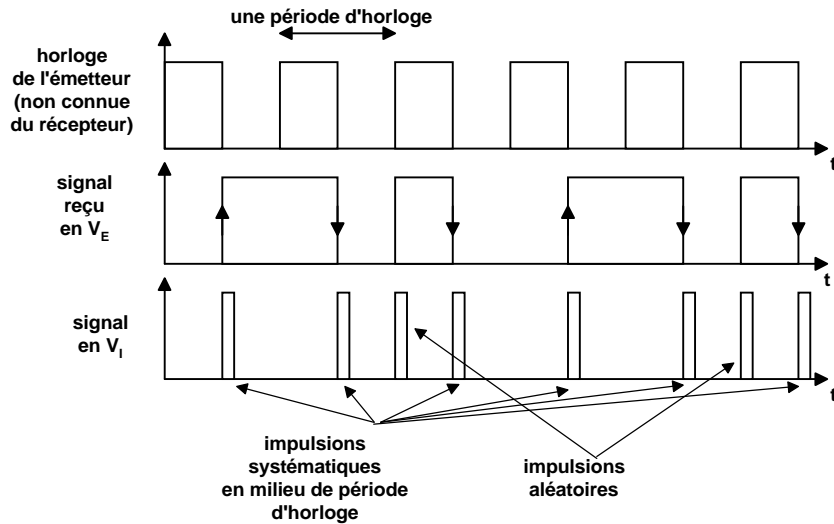
Le problème est que le récepteur connaît la fréquence de l'horloge d'émission, mais pas sa phase (il ne peut prédire à quel endroit se trouve le milieu et les trois quarts de la période).

Il va donc falloir retrouver cette information à partir du signal reçu. Pour cela on peut remarquer que celui-ci comporte systématiquement un front en milieu de période, et aléatoirement un front en début (ou fin) de période (lorsque le signal NRZ ne change pas).

Un détecteur de front va dans un premier temps, à partir du signal incident filtré, amplifié et remis en forme, V_E , fournir une brève impulsion en V_I à chaque front montant ou descendant.



En sortie de celui-ci nous aurons donc des « impulsions systématiques » et des « impulsions aléatoires ».



Nous allons éliminer ces dernières au moyen d'un monostable M1 à front montant, non redéclenchable qui fournira un NL1 pendant une durée correspondant à 75% de la période de l'horloge d'émission.

Après un état transitoire, les fronts montants de la sortie de ce monostable seront calés sur la moitié de la période de l'horloge d'émission. Nous avons donc retrouvé l'information phase de cette dernière.

Pour nous placer aux trois quarts de la période de l'horloge d'émission, nous utilisons un monostable M2 à front montant placé en sortie du premier (de manière à ne plus voir les impulsions aléatoires) qui fournit à sa propre sortie un NL0 pendant une durée correspondant à 25% de la période d'horloge. Cette sortie nous donne donc un front montant aux trois quarts de la période d'horloge d'émission.

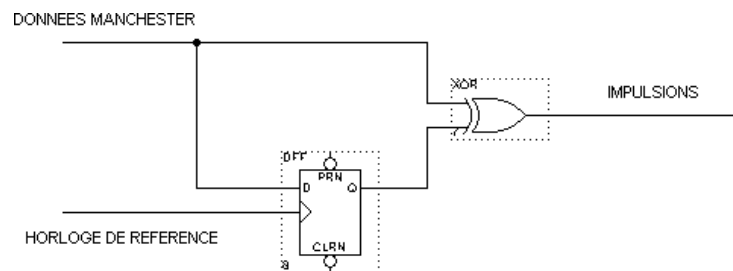
L'échantillonneur numérique recopie alors sur sa sortie la valeur du signal reçu par le récepteur au moment de ce front montant.

Donner les chronogrammes aux différents endroits du schéma structurel lorsque la suite NRZ codée par l'émetteur est « 0111010 ».

Vérifier qu'on retrouve bien cette suite en sortie du décodeur après un certain retard. Justifier ce retard.

3.1.1. Détecteur de front

Le schéma structurel de celui-ci est donné ci-après :



Le signal d'horloge de la bascule D est supposé de fréquence très supérieure à tous les autres signaux et sans relation de phase avec le signal reçu par le récepteur.

Dans la pratique nous utiliserons le signal issu HQ de l'oscillateur à quartz dont la fréquence est 16 fois supérieure à celle de l'horloge d'émission.

On rappelle que sur une bascule D, pour être prise en compte, la donnée à l'entrée D doit être présente un temps, dit temps de prépositionnement (set up time), de quelques ns avant le front d'horloge.

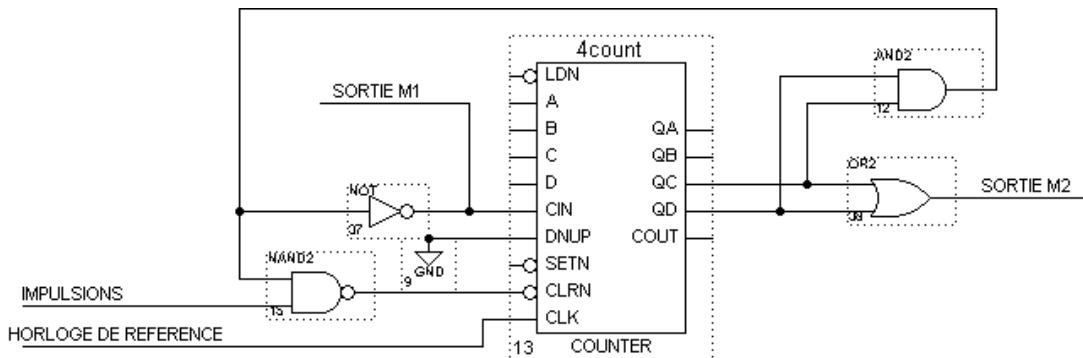
Donner les chronogrammes des différents signaux du détecteur et vérifier que l'on a bien la fonction souhaitée.

Remarque : la bascule D pourrait être remplacée par deux portes inverseuses par exemple, dont le temps de propagation constituerait un retard suffisant pour obtenir une impulsion en sortie du ou exclusif à chaque front. Ce type de solution pose parfois problème lors de la synthèse, le logiciel simplifiant les équations booléennes, les deux portes inverseuses sont alors supprimées. Une solution possible consiste à placer une sortie fictive entre les deux.

3.1.2. Monostables

Sur un circuit purement numérique (ne faisant pas intervenir de circuits RC) une telle fonction peut être réalisée par un compteur.

Dans notre cas le schéma est donné ci-après :



Les impulsions comptées sont celle de l'oscillateur à quartz de fréquence 16 fois supérieure à celle de l'horloge d'émission.

La documentation du compteur est donnée par le tableau suivant :

4count (Counter)															
Macrofunctions															
4-Bit Binary Up/Down Counter with Synchronous Load (LDN), Asynchronous Clear, and Asynchronous Load (SETN)															
Default Signal Levels: <u>GND</u> -A, B, C, D, CLK <u>VCC</u> -LDN, CIN, DNUP, SETN, CLRN															
AHDL Function Prototype (port name and order also apply to Verilog HDL):															
FUNCTION 4count (clk, clrn, setn, ldn, cin, dnup, d, c, b, a)															
RETURNS (qd, qc, qb, qa, cout);															
Inputs										Outputs					
CLK	CLR	SETN	LDN	CIN	DNUP	D	C	B	A	QD	QC	QB	QA	COUT	Note
X	L	X	X	X	X					L	L	L	L	X	
X	H	L	X	X	X	d	c	b	a	d	c	b	a	X	
J	H	H	L	X	X	d	c	b	a	d	c	b	a	X	
J	H	H	H	L	X						Hold			X	
J	H	H	H	H	H					Count Down			L		
J	H	H	H	H	L					Count Up			L		

Vérifier en étudiant les chronogrammes aux différents endroits que l'on a bien la fonction souhaitée.

3.1.3. Echantillonneur numérique

Proposer un schéma structurel de cet échantillonneur en utilisant une bascule D.

4. Synthèse par macro-fonctions et simulation

Nous allons maintenant implanter ces différentes fonctions dans un CPLD à l'aide du logiciel Max+plus II. Les différents éléments de base dont nous allons avoir besoin, bascule D (DFF), inverseurs (NOT), ou exclusif (XOR), entrées (INPUT), sorties (OUTPUT), signaux au niveau logique 1 et 0 (Vcc et GND) se trouvent dans la bibliothèque **c:\maxplus2\max2lib\prim**.

Pour obtenir de la documentation sur un de ces éléments, on pourra cliquer avec le bouton gauche de la souris sur le symbole « ? » de la barre d'outils du haut, puis sur le composant désiré.

On pourra placer une sortie sur chaque signal que l'on souhaite visualiser sur le simulateur afin de pouvoir les repérer facilement dans la liste des signaux proposés par le logiciel.

4.1. Emetteur

Dans le logiciel Max+plus II ouvrir une feuille graphique et la nommer EMET.gdf.

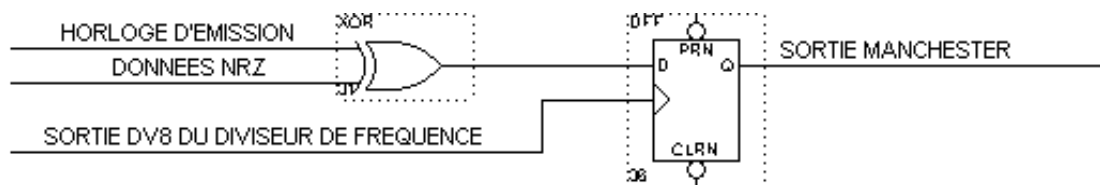
Implanter le diviseur de fréquence par 16 ; son nom est Freqdiv et il se trouve dans la bibliothèque **c:\maxplus2\max2lib\mf**. Imposer une borne de sortie pour le signal d'horloge.

Tester le bon fonctionnement (on se limitera à une simulation fonctionnelle).

Ajouter le générateur pseudo-aléatoire en imposant une borne de sortie pour le signal NRZ et tester le fonctionnement.

Placer maintenant le codeur et tester de nouveau.

On peut remarquer la présence sur le signal codé manchester d'impulsions parasites (glitch) lorsque le signal NRZ change de niveau. Ces impulsions sont dues à un décalage entre les données NRZ et l'horloge d'émission. Dans un système réel elles seraient supprimées par les filtres de sortie et resteraient sans incidence. Dans notre application, nous allons devoir resynchroniser la sortie du codeur en modifiant le schéma structurel de celui-ci de la manière suivante :



Tester cette solution.

4.2. Récepteur

Ouvrir une nouvelle feuille graphique et la sauvegarder sous RECEPT.gdf.

Implanter et tester le détecteur de front. Pour le test on prendra comme signal d'entrée une tension susceptible de correspondre à un signal manchester en respectant les périodes d'horloge.

Placer ensuite le monostable en utilisant le composant 4count de la bibliothèque **c:\maxplus2\max2lib\mf**. Tester l'ensemble.

Implanter l'échantillonneur numérique et vérifier le fonctionnement.

4.3. Ensemble émetteur récepteur

Ouvrir une nouvelle feuille graphique, la nommer EMET_RECEPT.gdf, y placer l'émetteur et le récepteur puis vérifier le bon fonctionnement de l'ensemble.

5. Synthèse en VHDL et simulation

L'ensemble émetteur récepteur va maintenant être décrit en langage VHDL, description à partir de laquelle le logiciel établira les liaisons entre les différentes portes et bascules à l'intérieur du CPLD.

5.1. Emetteur

5.1.1. Diviseur de fréquence

Le programme suivant décrit un diviseur de fréquence permettant d'obtenir en sortie deux signaux de fréquence 32 fois et 64 fois plus faible que celle du signal d'entrée.

```
-----
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
--les paquetages std_logic_1164 et std_logic_unsigned de la bibliothèque ieee
--permettent respectivement l'utilisation du type std_logic et l'addition avec ce type

entity div_freq is
--H est le signal d'horloge à 25,175 MHz
--S32 et S64 sont des signaux de fréquence respectivement 32 et 64 fois plus faible

port ( H          : in std_logic;
      S32, S64    : out std_logic);
end div_freq;

architecture archdiv of div_freq is
--pour compter jusqu'a (64 -1) il faut 6 bits
    signal X      : std_logic_vector (5 downto 0);

begin

    process (H)
    begin
        if (H'event and H = '1') then X <= X + 1 ;
--la gestion du modulo du compteur est inutile, après « 11111 » X prend la valeur « 00000 »

--à la moitié du comptage on change la valeur de S64 (rapport cyclique 1/2)
                if X >= 32      then S64 <= '0' ;
                                else S64 <= '1' ;
                end if ;

--génération de S32 (rapport cyclique 1/2)
                if (X >= 16 and X<=31) or X >= 48      then S32 <= '0' ;
                                                         ELSE S32 <= '1' ;
                end if ;

            end if;

        end process;
    end archdiv;
-----
```


Modifier le programme pour obtenir les divisions désirées ; tester le fonctionnement.

5.1.2. Générateur pseudo-aléatoire

Le programme suivant décrit un générateur à 6 bascules. Il utilise une instruction de boucle ; il est important de comprendre que ce type d'instruction décrit une redondance de la structure matérielle, il s'agit d'une boucle « spatiale » et non temporelle (comme le « process » par exemple).

Remarque : le programme suivant utilise l'instruction « xnor » (ou exclusif complémenté) qui n'est définie qu'en VHDL 93. Au moment de la compilation, il faudra donc vérifier que le VHDL 93 est bien le langage choisi ; pour cela, dans la fenêtre du compilateur, valider l'option **VHDL 93** dans le menu **Interface / VHDL Netlist Reader Setting**.

```
-----  
library ieee;  
use ieee.std_logic_1164.all;  
--le paquetage std_logic_1164 de la bibliothèque ieee  
--permet l'utilisation du type std_logic  
  
entity gene_alea is  
port ( HOR           : in std_logic;  
       SORTIE_NRZ    : out std_logic);  
end gene_alea;  
  
architecture arch of gene_alea is  
    signal X          : std_logic_vector (5 downto 0);  
  
begin  
    process (HOR)  
    begin  
        if HOR'event and HOR='1' then  
            X(0)<=X(4) xnor X(2);  
            for i in 0 to 3 loop  
                X(i+1)<=X(i);  
            end loop;  
        end if;  
    end process;  
    SORTIE_NRZ<=X(4);  
  
end arch;  
-----
```

Modifier le programme pour obtenir un générateur à 4 bascules ; tester le fonctionnement.

Modifier de nouveau le programme pour remplacer la boucle par l'utilisation de l'opérateur de concaténation « & » ; tester le fonctionnement.

5.1.3. Codeur

Proposer un programme décrivant le codeur, sans tenir compte dans un premier temps des impulsions parasites ; tester le fonctionnement.

Proposer maintenant un programme permettant de supprimer les impulsions parasites, basé sur le même principe que celui utilisé précédemment ; compiler et vérifier qu'il n'y a pas d'erreur (inutile d'effectuer une simulation).

5.1.4. Ensemble émetteur

Ouvrir une feuille graphique et la nommer EMET_VHDL.gdf ; y assembler les symboles des différents éléments décrit précédemment. Tester le fonctionnement de l'ensemble.

5.2. Récepteur

5.2.1. Détecteur de front

Proposer une description VHDL du détecteur de front et la tester.

5.2.2. Monostable

Le programme suivant décrit les deux monostables dans le cas où un rapport 64 existerait entre les fréquences de l'horloge de référence et celle d'émission :

```
-----  
library ieee;  
use ieee.std_logic_1164.all;  
use ieee.std_logic_unsigned.all;  
--les paquetages std_logic_1164 et std_logic_unsigned de la bibliothèque ieee  
--permettent respectivement l'utilisation du type std_logic et l'addition avec ce type  
  
entity MONO is  
  
port ( FRONT, H      : in std_logic;  
      S25            : out std_logic);  
end MONO;  
  
architecture arch of MONO is  
    SIGNAL X      : std_logic_vector (5 downto 0);  
    SIGNAL S75    : std_logic;  
  
begin  
--le front descendant de S75 est placé à 75% de la période  
    S75 <='1' when X<=47 else '0';  
--le front montant de S25 est placé à 25% de la période  
    S25 <='0' when X<=15 else '1';  
  
    process (H, FRONT)  
    begin  
  
        if (H'event and H = '1') then  
--autorisation de comptage pendant 75% du temps après l'impulsion de front  
            if S75='1' then X <= X + 1 ;  
--inhibition de l'entrée de déclenchement pour les impulsions aléatoires  
            elsif FRONT='1' then X <= "000000";  
            end if;  
  
        end if;  
    end process;  
  
end arch;
```

Modifier le programme pour l'adapter à notre cas. Tester.

5.2.3.Echantillonneur numérique

Proposer une description VHDL réalisant la fonction désirée. Tester.

5.2.4.Ensemble récepteur

Ouvrir une feuille graphique et la nommer RECEPT_VHDL.gdf ; y assembler les symboles des différents éléments du récepteur. Tester l'ensemble.

5.3. Ensemble émetteur récepteur

Ouvrir une feuille graphique et la nommer EMET_RECEPT_VHDL.gdf ; y assembler les symboles de l'émetteur et du récepteur et tester le fonctionnement.