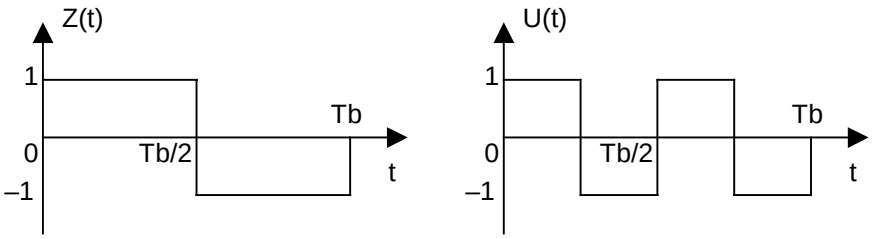
CACHAN
Département d'électronique

Cachan le 12 octobre 2001

Devoir surveillé d'informatique industrielle

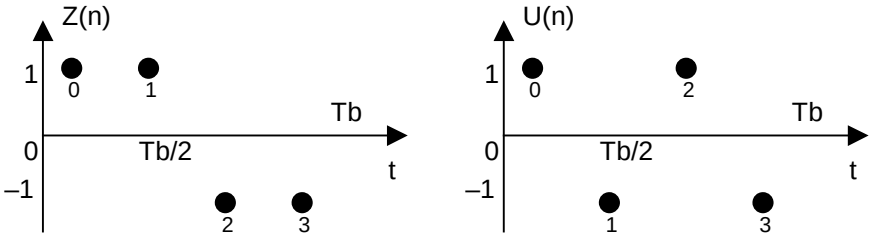
Durée 2h, tous documents autorisés. Lire et comprendre tout le texte avant de répondre aux questions.

Dans une transmission binaire (câble coaxial, par exemple) l'émetteur utilise deux signaux Z(t) et U(t) pour transmettre respectivement un « Zéro » et un « Un » binaires. Chaque signal a une durée Tb, qui correspond à un débit binaire Fb = 1/Tb. Idéalement (si on ne tient pas compte de la forme arrondie des signaux réels), les deux signaux Z(t) et U(t) ont la forme représentée ci-dessous. L'unité de l'axe vertical est arbitraire, on pourra considérer qu'il s'agit de volts.

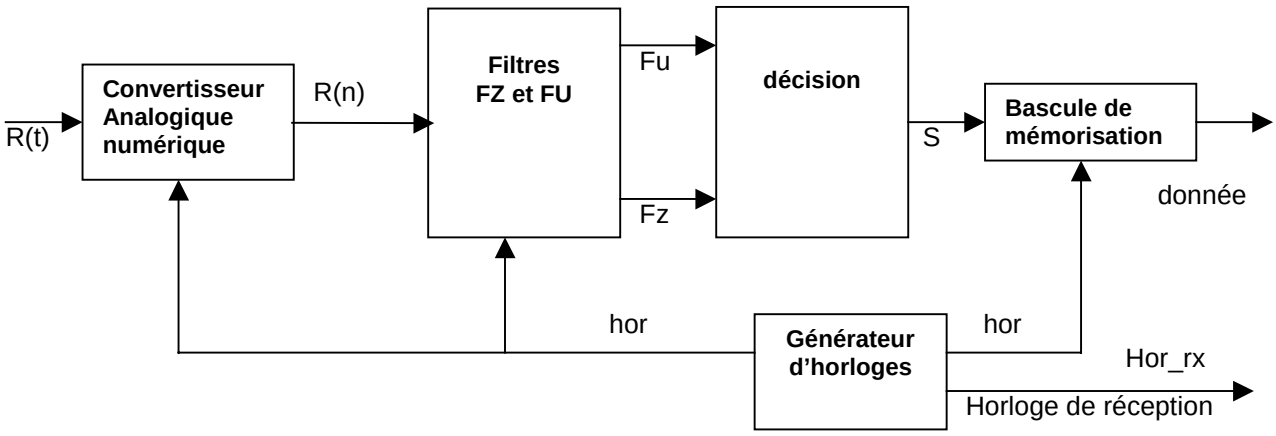


A la réception le signal reçu R(t) est échantillonné et converti en nombres entiers R(n) signés sur 2 bits, pouvant prendre les valeurs +1 ou –1. La **fréquence d'échantillonnage Fh est exactement égale à 4 Fb**, grâce à un circuit de restitution de rythme (PLL) qui n'est pas l'objet de l'étude. Cette fréquence Fh sert de fréquence d'horloge à l'ensemble du récepteur.

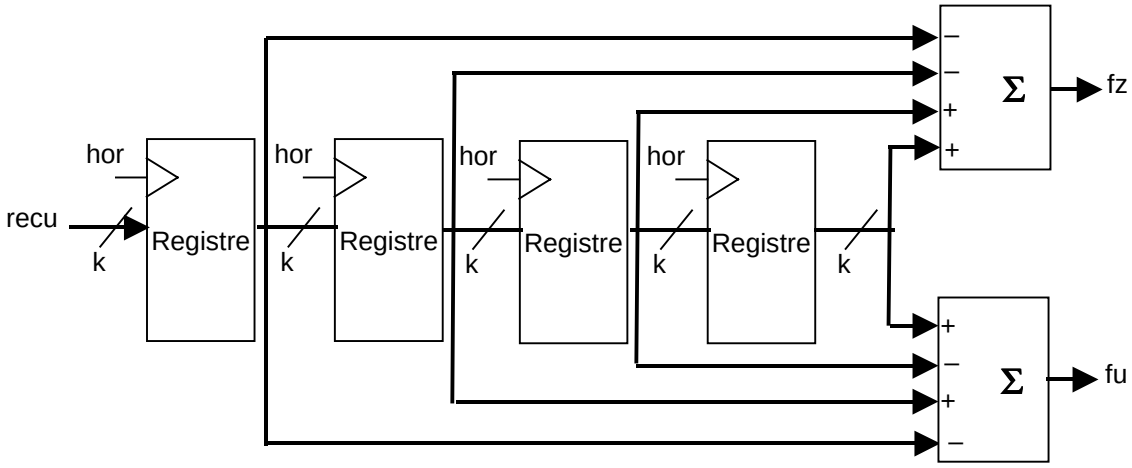
Après mise en forme et échantillonnage les signaux reçus possibles sont modélisés par :



Le synoptique du récepteur répond au schéma de principe ci-dessous :



Les filtres FU et FZ sont réalisés par le montage suivant :



Les blocs marqués '**Σ**' calculent la somme algébrique de leurs entrées, affectées du signe '+' ou '–' indiqué.

Le programme VHDL qui décrit une version possible du récepteur est fourni en annexe.

On rappelle que le produit scalaire de deux signaux sur 4 échantillons est donné par :

$$P = \sum_{i=0}^{3} S1(i) * S2(i)$$

Questions :

1. Montrer que les signaux Z(n) et U(n) sont orthogonaux.
2. Déterminer la réponse impulsionnelle des filtres FZ et FU.
3. Quelle est la nature de ces filtres (IIR, FIR) ?
4. Quelle doit-être la valeur minimum de k, nombre de bits de calcul d'un sommateur '**Σ**' ?
5. En admettant que les registres sont initialisés à zéro, déterminer les séquences de réponse des deux filtres aux séquences d'entrée U(n) et Z(n) (on donnera toutes les valeurs intermédiaires).
6. Quelle comparaison peut-on faire avec une technique de corrélation ?
7. Quelles sont les séquences de sortie des deux filtres si, après une initialisation, on applique la séquence « UZZU » sans remettre à zéro les registres entre deux motifs (sous séquences U ou Z) ?
8. Identifier dans le programme fourni les différents blocs du synoptique et du schéma des filtres (on donnera les numéros de lignes correspondants).
9. Reprendre la réponse à la question 7 en utilisant le programme VHDL. Qu'a-t-on rajouté et pourquoi ?
10. Donner le chronogramme des signaux de sortie (dout, hor_rx) que l'on obtient avec la séquence précédente (au moins 16 périodes d'horloge après une initialisation du circuit par l'entrée raz).
11. A quoi sert la ligne 46 du programme (ne pas se contenter du commentaire fourni !) ?
12. Le programme fourni est-il synthétisable ? Si oui quelle doit être la taille du circuit en nombre minimum de macrocellules ?

```
1   -- fil_opt_2bits.vhd
2
3   library ieee ;
4   use ieee.std_logic_1164.all, ieee.numeric_std.all ;
5
6   entity fil_opt is
7      port (hor, raz : in std_logic ;
8            din : in std_logic_vector(1 downto 0) ;
9            dout, hor_rx : out std_logic ) ;
10  end fil_opt ;
11
12  architecture simple of fil_opt is
13     subtype echantillon is signed(3 downto 0) ; -- vecteur représentant un
14                                                 -- nombre signé
15     type histoire is array(1 to 4) of echantillon ;
16     signal reg : histoire ; -- tableau de 4 nombres
17     signal recu, fz, fu : echantillon ;
18     signal hor_donnees : std_logic ;
19  begin
20  recu(1) <= din(1) ; -- recuperation de l'entree sous
21  recu(0) <= din(0) ; -- forme de nombre signe
22  recu(3 downto 2) <= (others => din(1)) ; -- extension du signe
23  fz <= reg(4) + reg(3) - reg(2) - reg(1) ;-- detection d'un zero en entree
24  fu <= reg(4) - reg(3) + reg(2) - reg(1) ;-- detection d'un un en entree
25  acquisition : process
26  begin
27     wait until rising_edge(hor) ;
28        if raz = '1' then
29           hor_donnees <= '0' ;
30           reg <= (others => (others => '0')) ; --remise à zero generale
31        else
32           reg(1) <= recu ;
33           if fu >= 4 or fz >= 4 then  -- on vient de trouver un motif valable
34              hor_donnees <= '1' ;
35              reg(2 to 4) <= (others => (others => '0')) ; -- raz cellules 2 a 4
36           else
37              hor_donnees <= '0' ;
38              reg(2 to 4) <= reg(1 to 3) ;
39           end if ;
40        end if ;
41  end process ;
42
43  sorties : process
44  begin
45     wait until rising_edge(hor) ;
46     hor_rx <= hor_donnees ; -- on cree un retard d'une periode d'horloge
47     if fu >= 4 then
48        dout <= '1' ;
49     elsif fz >= 4 then
50        dout <= '0' ;
51     end if ;
52  end process ;
53  end simple ;
```