```vhdl
----------------------------------------------------------------------------
-- horloge fonctionnement clock_fpga=50MHz
----------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity reception_RS232 is
port(
   clock_fpga      : in std_logic;
   RX              : in std_logic;
   data_valide     : out std_logic;
   data_recu       : out std_logic_vector(15 downto 0));

end entity reception_RS232;

architecture Behavioral of reception_RS232 is


------------------------------------------------------------------------
-- SIGNALS
------------------------------------------------------------------------

signal horloge_prelevement : std_logic :='0';
signal mot_complet : std_logic_vector(17 downto 0) :="000000000000000000";
signal compteur: std_logic_vector(4 downto 0):="00000";
signal compteur_baud: integer:=0;
signal prelevement_enable : std_logic :='0';

begin

----------------------------------------------------------------------------
--reorganisation de la data recu; bit poid faible en premier------------

data_presente: process(horloge_prelevement,compteur)
  begin
  if horloge_prelevement'event and horloge_prelevement='0' then
   if (compteur="10001") then
    data_recu(15) <= mot_complet(0);
    data_recu(14) <= mot_complet(1);
    data_recu(13) <= mot_complet(2);
    data_recu(12) <= mot_complet(3);
    data_recu(11) <= mot_complet(4);
    data_recu(10) <= mot_complet(5);
    data_recu(9) <= mot_complet(6);
    data_recu(8) <= mot_complet(7);
    data_recu(7) <= mot_complet(8);
    data_recu(6) <= mot_complet(9);
    data_recu(5) <= mot_complet(10);
    data_recu(4) <= mot_complet(11);
    data_recu(3) <= mot_complet(12);
    data_recu(2) <= mot_complet(13);
    data_recu(1) <= mot_complet(14);
    data_recu(0) <= mot_complet(15);
   end if;
  end if;
  if (compteur="10010") then
   data_valide<='1';
  else
   data_valide<='0';

  end if;
  end process data_presente;



-----------------------------------------------------------------------------
--reception des bits dans un registre à decalage----------------
---------------------------------------------------
```

```
mot_entier_recu: process(horloge_prelevement,compteur)
   begin
      if horloge_prelevement'event and horloge_prelevement='1' then
         compteur <= compteur + 1;
         mot_complet <= mot_complet(16 downto 0) & RX;
      end if;
      if compteur ="10010" then
            compteur<="00000";
           -- data_valide<='1';
           -- else
           -- data_valide<='0';
      end if;
   end process mot_entier_recu;
-----------------------------------------------------------------------------

------------------------------------------------------------------------
--horloge permettant de prelever au bon moment la data présente sur RX--
------------------------------------------------------------------------

generation_horloge_prelevement: process(clock_fpga)
   begin
     if clock_fpga'event and clock_fpga='1' then
        if prelevement_enable = '1' then
           compteur_baud<=compteur_baud+1;
           if (compteur_baud =2604 and compteur ="00000") then --Frequence/(baud/2)
              horloge_prelevement<='1';
              compteur_baud<=0;
           else
               if (compteur_baud =5208 and compteur /="00000") then --Frequence/(baud)
                 horloge_prelevement<='1';
                 compteur_baud<=0;
               else
                 horloge_prelevement<='0';
               end if;
           end if;
        end if;
      end if;
   end process generation_horloge_prelevement;
-----------------------------------------------------------------------------

-----------------------------------------------
--mise à 1 lorsque detection bit start--
--repasse à 0 à la fin de la reception du mot--
-----------------------------------------------

generation_prelevement_enable: process(RX,compteur)
   begin
    if (RX'event and RX='0') then
       if compteur="00000" then
          prelevement_enable<='1';
       end if;
    end if;
    if (compteur="10010") then
         prelevement_enable<='0';

    end if;
   end process generation_prelevement_enable;

 -----------------------------------------------------------------------------


end behavioral;
```