Serial Data Transmitter

Presented to: Dr. Mohamed El-Banna

Dr. Nayera Sadek

Eng. Khaled Essam

Eng. Mostafa Ayesh

By: Mostafa Mahmoud Khalil Hassan

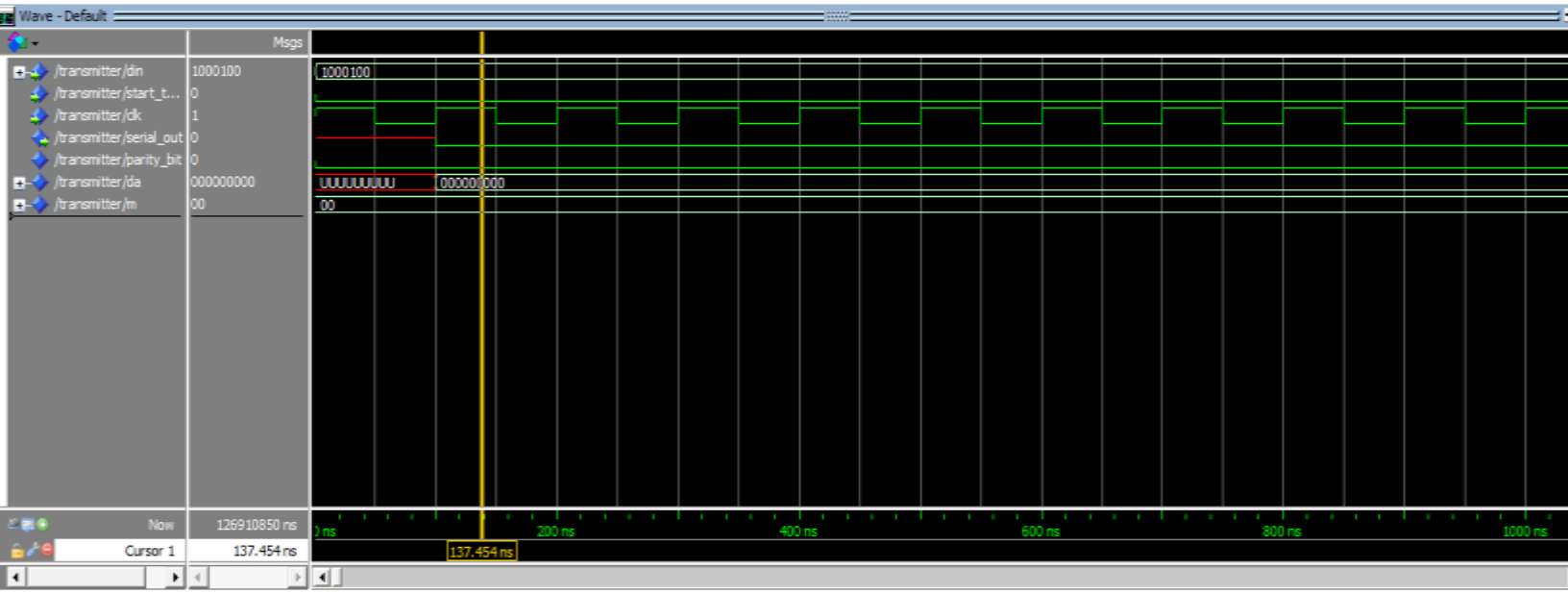
VHDL Code:

```
Library IEEE;
Use IEEE.STD_LOGIC_1164.ALL;
Use IEEE.STD_LOGIC_UNSIGNED.ALL;

Entity transmitter is
        Port ( din              : in  STD_LOGIC_VECTOR (6 downto 0);        -- input data that will be sent
             start_transmit  : in STD_LOGIC;                                -- enable sending
              clk               : in STD_LOGIC;
            serial_out        : BUFFER  STD_LOGIC);                        -- the sent data
End transmitter ;

Architecture Behavior of transmitter is
         Signal parity_bit     : STD_LOGIC;
        Signal da              : STD_LOGIC_VECTOR (8 downto 0);
        Signal m               : STD_LOGIC_vector (1 downto 0) :="00";
                -- That is a counter to not to load the same data again.
Begin
        parity_bit <= din(0) xor din(1) xor din(2) xor din(3) xor din(4) xor din(5) xor din(6) ;
        Process
        Begin
                Wait until  (clk'EVENT AND clk='1') ;
                   if start_transmit='1' then
                        if (m = "00") then
                                da (8 downto 0) <= '1' & parity_bit & din (6 downto 0);
                                serial_out <= '1';
                                m <= "01" ;
                        elsif (m = "01") then
                                Genbits: FOR i IN 0 To 7 LOOP               -- shifting circuit
                                da (i) <=da(i+1);
                                End LOOP;
                                serial_out<= da (0);
                                da (8) <= '0';
                        End if;
                elsif start_transmit = '0' then
                        m <="00";
                        da <= "000000000" ;
                        serial_out <='0';
                End if;
        End process;
End Behavioral;
```

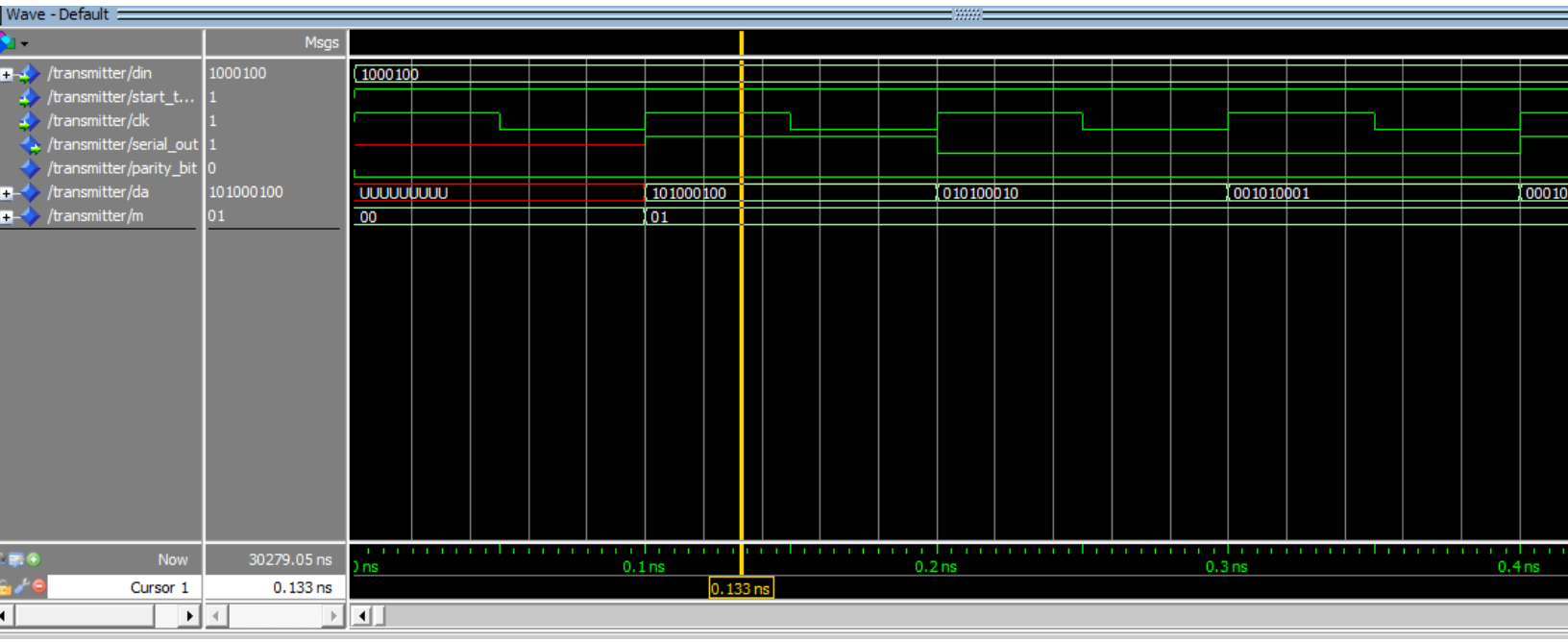
The steps of the design:

The simulation

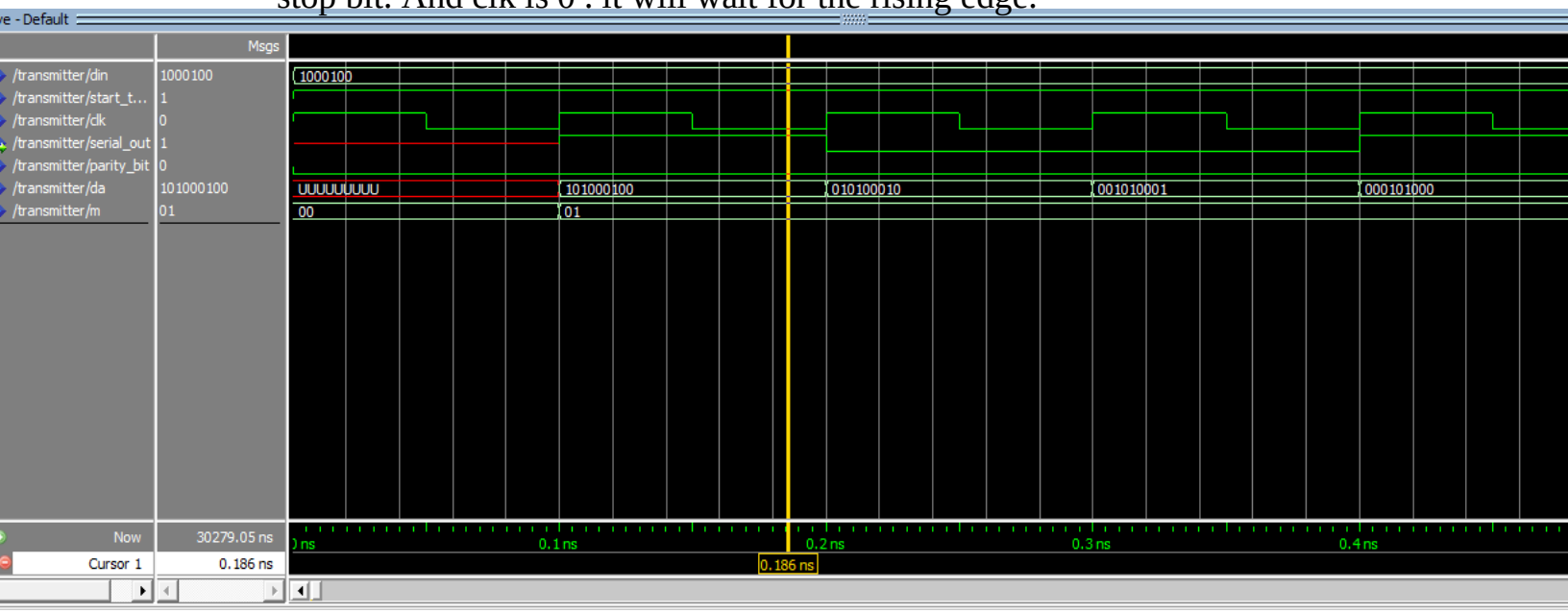
1- When Start_transmit = 0 .



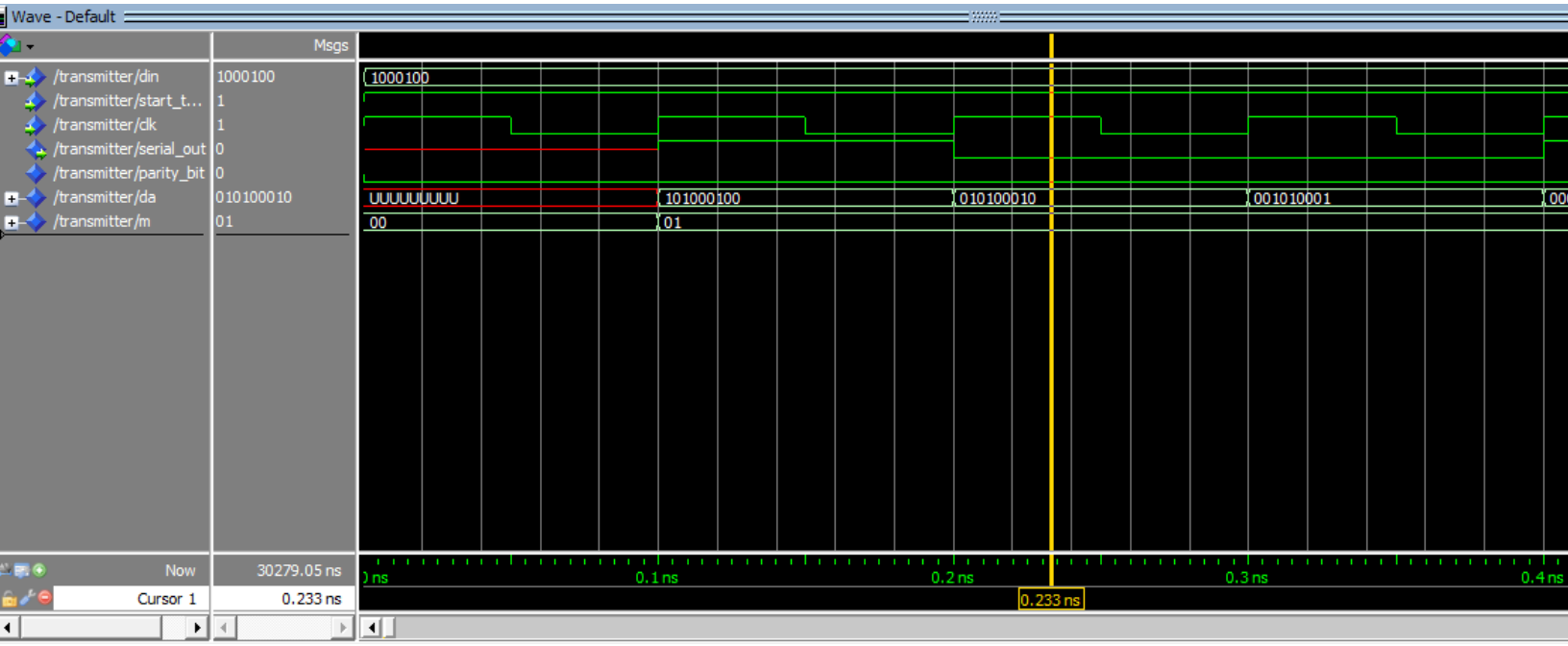
2- When start transmit =1 , it send start bit and clk is 1

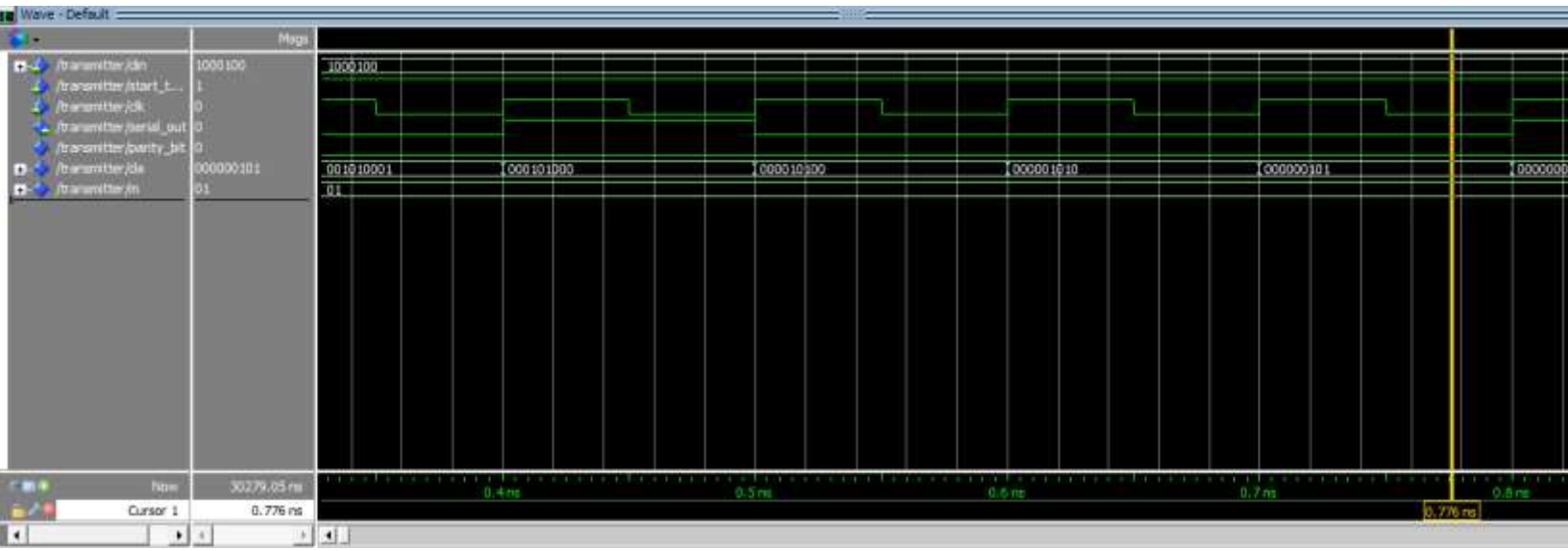


3- When start transmit =1 , it send all data bits including the parity_bit and the stop bit. And clk is 0 . it will wait for the rising edge.

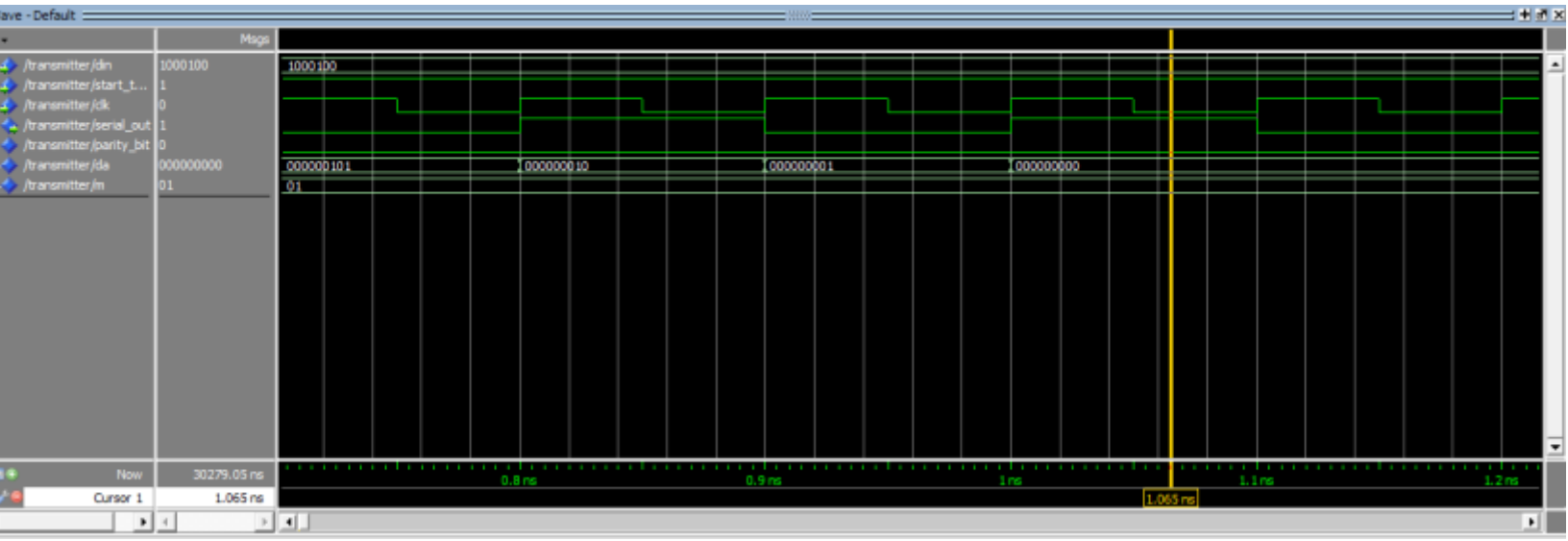


4- When start transmit = 1 , clk = 1 ,
start shifting 1 bit then clk change to 0 (will wait for the next rising edge to shift the other bit .... etc.)





all din are sent so it will back to 0000000000



block diagram :