```vhdl
-----------------------------------------
-- Block code:  sync_n_edgeDetector.vhd
-- History:     15.Nov.2017 - 1st version (dqtm)
--              15.Jan.2018 - adapt reset value for usage in mini-project (dqtm)
--              01.Mar.2018 - rename in English (dqtm)
--                 <date> - <changes>  (<author>)
-- Function: edge detector with rise & fall outputs.
--           Declaring FFs as a shift-register.
-----------------------------------------

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY sync_n_edgeDetector IS
  PORT( data_in     : IN    std_logic;
        clock       : IN    std_logic;
        reset_n     : IN    std_logic;
        data_sync   : OUT   std_logic;
        rise        : OUT   std_logic;
        fall        : OUT   std_logic
        );
END sync_n_edgeDetector;


ARCHITECTURE rtl OF sync_n_edgeDetector IS
    -- Signals & Constants Declaration
    SIGNAL shiftreg, next_shiftreg: std_logic_vector(2 downto 0);

BEGIN
    -----------------------------------------
    -- Process for combinatorial logic
    -- OBs.: small logic, could be outside process,
    --       but doing inside for didactical purposes!
    -----------------------------------------
    comb_proc : PROCESS(data_in, shiftreg)
    BEGIN
        next_shiftreg <= data_in & shiftreg(2 downto 1) ;  -- shift direction towards LSB
    END PROCESS comb_proc;

    -----------------------------------------
    -- Process for registers (flip-flops)
    -----------------------------------------
    reg_proc : PROCESS(clock, reset_n)
    BEGIN
        IF reset_n = '0' THEN
            shiftreg <= (OTHERS => '1');
        ELSIF (rising_edge(clock)) THEN
            shiftreg <= next_shiftreg;
        END IF;
    END PROCESS reg_proc;

    -----------------------------------------
    -- Concurrent Assignments
    -- OBs.: no logic after the 1st-FF (shiftreg(2)) because it was added for sync purposes
    -----------------------------------------
    rise        <=     shiftreg(1)  AND NOT(shiftreg(0));
    fall        <= NOT(shiftreg(1)) AND     shiftreg(0);
    data_sync   <=     shiftreg(1) ;    -- take serial_in at same period as fall/rise pulse

END rtl;

```