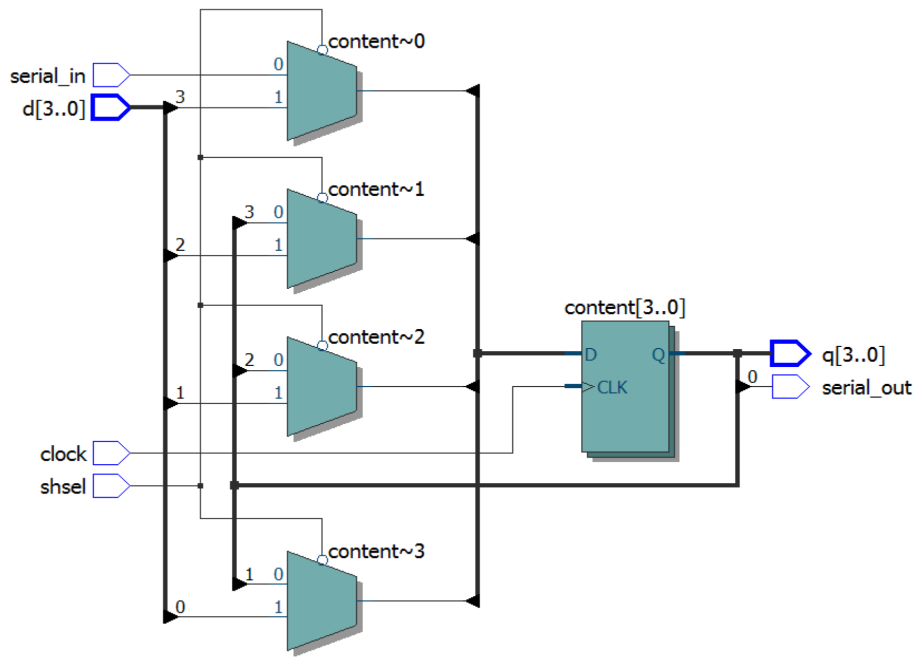
```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity aio_shift_reg is port(
    clock         : in std_logic;
    shsel         : in std_logic;
    serial_in     : in std_logic;
    d               : in std_logic_vector(3 downto 0);
    serial_out    : out std_logic;
    q             : out std_logic_vector(3 downto 0));
end aio_shift_reg;

architecture rtl of aio_shift_reg is
    signal content: std_logic_vector(3 downto 0);
begin
    process(clock)
    begin
        if(rising_edge(clock))then
            case shsel is
                when '0' => content <= d; --load
                when '1' => content <= serial_in & content(3 downto 1); --shift right, pad with bit from
serial_in
                when others => null;
            end case;
        end if;
    end process;

    q <= content;
    serial_out <= content(0);
end rtl;
```



```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity tbaio_shift_reg is
end entity;

architecture beh of tbaio_shift_reg is
    component aio_shift_reg is port(
        clock         : in std_logic;
        shsel         : in std_logic;
        serial_in     : in std_logic;
        d               : in std_logic_vector(3 downto 0);
        serial_out    : out std_logic;
        q             : out std_logic_vector(3 downto 0));
end component;

signal clock         : std_logic;
signal shsel         : std_logic;
signal serial_in     : std_logic;
signal serial_out    : std_logic;
signal q             : std_logic_vector(3 downto 0);

constant period : time := 50 ns;
constant strobe : time := 45 ns;

signal lfsr_temp : std_logic_vector(0 to 3) :=(0 => '1', others => '0');
begin
    clk0 : process
    begin
        clock <= '0'; wait for period;
        clock <= '1'; wait for period;
    end process;

    lfsr : process
    begin
        wait until rising_edge(clock);
        lfsr_temp <= (lfsr_temp(2) xor lfsr_temp(3)) & lfsr_temp(0 to 2);
    end process;

    p0: process
    begin
        loop
            wait for strobe;
            shsel <= '0';
            wait for strobe *2;
            shsel <= '1';

            for i in 0 to q'high loop
                wait until rising_edge(clock);
            end loop;
        end loop;
    end process;

    reg0 : aio_shift_reg
        port map(clock => clock, shsel => shsel, serial_in => serial_in, serial_out => serial_out, d =>
lfsr_temp, q => q);
end beh;
```