Practica 9. Cartas ASM

**Experimento 1.** Pasar la carta ASM a VHDL e identificar que hace.

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

entity cocheV2 is
port(clk, reset: in std_logic;
s: in std_logic_vector(1 downto 0);
estados, motorx, motorz: out std_logic_vector (1 downto 0));
end entity cocheV2;

architecture acacheV2 of cocheV2 is
signal present_state, next_state: state;
constant alto: state :="00";
constant mf: state :="01";
constant md: state :="10";
constant mi: state :="11";

begin
process(clk)
begin
if rising_edge(clk) then
if (reset='0') then
present_state<=alto;
else
present_state<= next_state;
end if;
end if;
end process;
```
process(present_state, s)
begin
  case present_state is
  when alto => case s is
    when "00" => next_state <= alto;
    when "01" => next_state <= mi;
    when "10" => next_state <= md;
    when others => next_state <= mf;
    end case;
    motorizq<= "00";
    motorder<= "00";
  --avanza
  when mf => case s is
    when "00" => next_state <= Alto;
    when "01" => next_state <= mi;
    when "10" => next_state <= md;
    when others => next_state <= mf;
    end case;
    motorizq<= "01";
    motorder<= "01";
  --giroder
  when md => case s is
    when "00" => next_state <= Alto;
    when "01" => next_state <= mi;
    when "10" => next_state <= md;
    when others => next_state <= mf;
    end case;
    motorizq<= "01";
    motorder<= "00";
  --giroizq
  when mi => case s is
    when "00" => next_state <= Alto;
    when "01" => next_state <= mi;
    when "10" => next_state <= md;
    when others => next_state <= mf;
    end case;
    motorizq<= "00";
    motorder<= "01";
  when others => next_state<= alto;
  end case;
  estados <= present_state;
end process;
end acocheV2;