```
--
--  File: K:\CSCI25˜1\VENDING\VendABEL.vhd
-- created: 11/07/00 16:16:07
--  from: ’K:\CSCI25˜1\VENDING\VendABEL.asf’
--  by fsm2hdl - version: 2.0.1.53
--
library IEEE;
use IEEE.std_logic_1164.all;

use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

library SYNOPSYS;
use SYNOPSYS.attributes.all;

entity vendabel is
  port (CLK: in STD_LOGIC;
        InThree: in STD_LOGIC;
        InTwo: in STD_LOGIC;
        RESET: in STD_LOGIC;
        Output: out STD_LOGIC);
end;

architecture vendabel_arch of vendabel is


-- SYMBOLIC ENCODED state machine: Sreg0
type Sreg0_type is (ConFiveSix, ConFour, ConThree, ConTwo, ConZero);
signal Sreg0: Sreg0_type;

begin
--concurrent signal assignments


Sreg0_machine: process (CLK)

begin

if CLK’event and CLK = ’1’ then
   if RESET
       then
           Sreg0 <= ConZero;
       else
           case Sreg0 is
               when ConFiveSix =>
                   if InTwo # InThree then
                       Sreg0 <= ConZero;
                   elsif !InTwo & !InThree then
                       Sreg0 <= ConFiveSix;
                   end if;
               when ConFour =>
                   if InTwo & !InThree then
                       Sreg0 <= ConFiveSix;
                   elsif !InTwo & !InThree then
                       Sreg0 <= ConFour;
                   elsif !InTwo & InThree then
                       Sreg0 <= ConZero;
                   end if;
               when ConThree =>
                   if !InTwo & !InThree then
                        Sreg0 <= ConThree;
                   elsif InTwo # InThree then
                        Sreg0 <= ConFiveSix;
                   end if;
               when ConTwo =>
                   if !InTwo & InThree then
                       Sreg0 <= ConFiveSix;
                   elsif InTwo & !InThree then
                       Sreg0 <= ConFour;
                   elsif !InTwo & !InThree then
                       Sreg0 <= ConTwo;
                   end if;
               when ConZero =>
                   if InTwo & !InThree then
                       Sreg0 <= ConTwo;
                   elsif !InTwo & !InThree then
                       Sreg0 <= ConZero;
                   elsif !InTwo & InThree then
                       Sreg0 <= ConThree;
                   end if;
               when others =>
                   null;
           end case;
       end if;
    end if;
end process;

-- signal assignment statements for combinatorial outputs
Output_assignment:
Output <= ^b1 when (Sreg0 = ConFiveSix and InTwo # InThree) else
          ^b0 when (Sreg0 = ConFiveSix and !InTwo & !InThree) else
          ^b0 when (Sreg0 = ConFour and InTwo & !InThree) else
          ^b0 when (Sreg0 = ConFour and !InTwo & !InThree) else
          ^b1 when (Sreg0 = ConFour and !InTwo & InThree) else
          ^b0 when (Sreg0 = ConThree and !InTwo & !InThree) else
          ^b0 when (Sreg0 = ConThree and InTwo # InThree) else
          ^b0 when (Sreg0 = ConTwo and !InTwo & InThree) else
          ^b0 when (Sreg0 = ConTwo and InTwo & !InThree) else
          ^b0 when (Sreg0 = ConTwo and !InTwo & !InThree) else
          ^b0 when (Sreg0 = ConZero and InTwo & !InThree) else
          ^b0 when (Sreg0 = ConZero and !InTwo & !InThree) else
          ^b0 when (Sreg0 = ConZero and !InTwo & InThree) else
          ^b0;

end vendabel_arch;
```