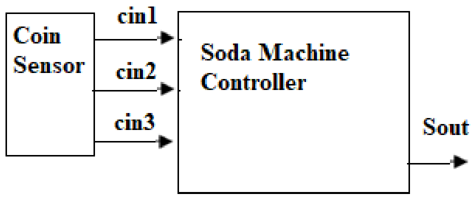
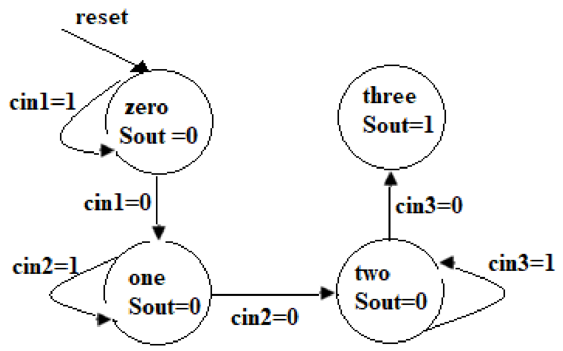
1. Design a soda machine controller, given that a soda costs 75 cents and your machine accepts quarters only. The coin sensor has three outputs to detect first coin, second coin and the third coin. The Once the soda is dispensed, the controller must be reset to be able to detect the coins again. The black-box view, the state diagram and the FSM VHDL code, are given here. The FPGA board push button are used to simulate the reset button and the coin sensor. **Since these push buttons have a pull-up resistor, they read 1 when they are not pressed. When you press them they will read in a zero.** A coin in is simulated by pressing the push button, therefore when the coin is inserted, cin = 0. Download the VHDL code into board and examine its functionality to practice.





As shown in the user constraint file the FPGA board push buttons are used for:

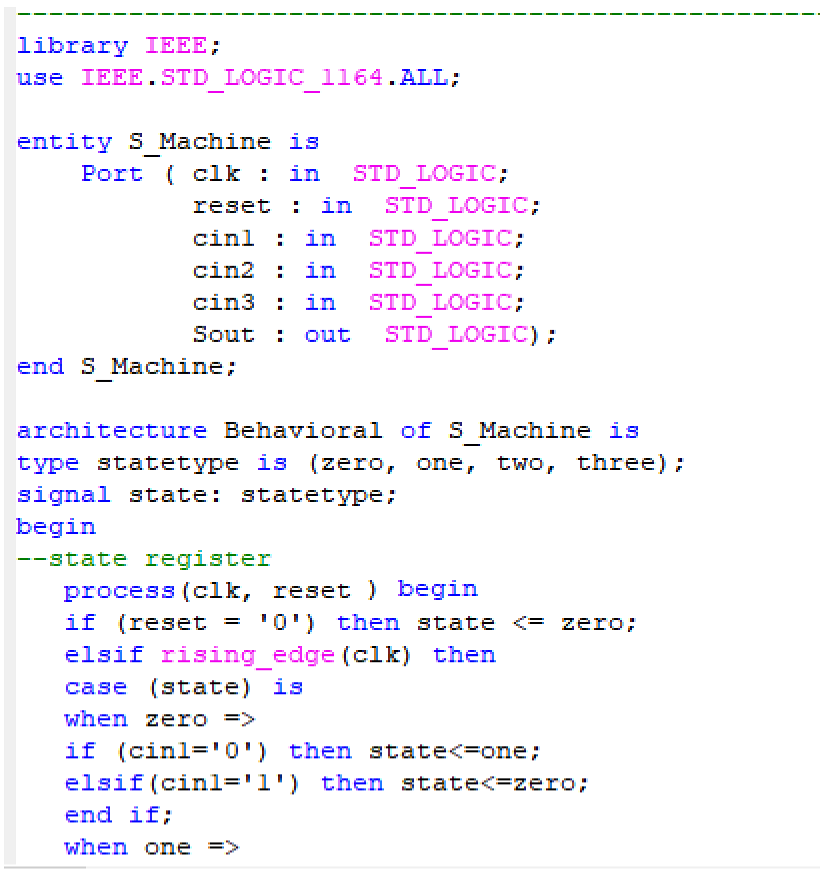
SW1 at pin 80 for reset button

SW2 at pin 79 for cin1

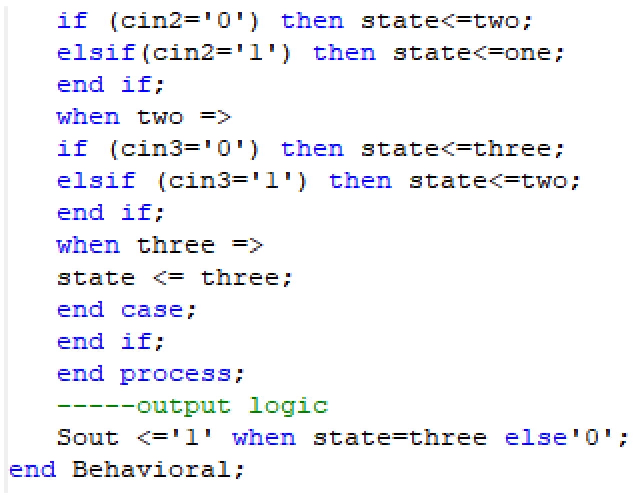
SW3 at pin 78 for cin2

SW4 at pin 77 for cin3

LED at pin 46 is used for Sout

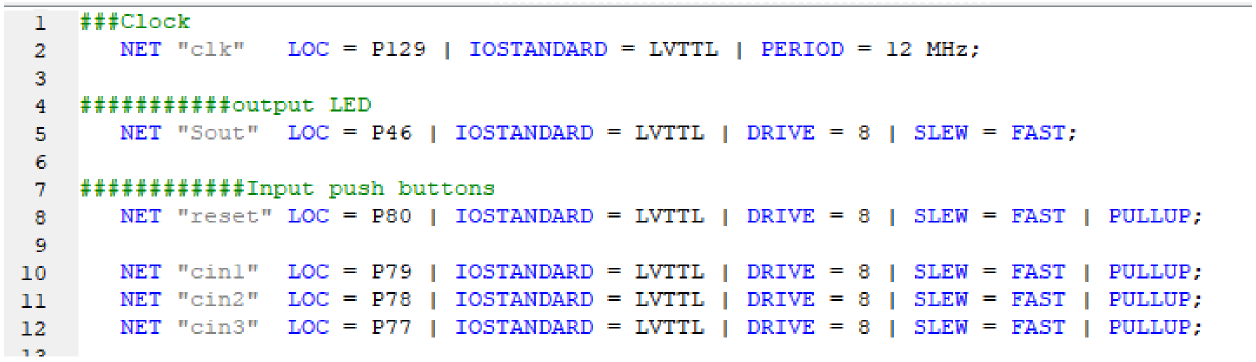


```vhdl
----------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity S_Machine is
    Port ( clk : in  STD_LOGIC;
           reset : in  STD_LOGIC;
           cin1 : in  STD_LOGIC;
           cin2 : in  STD_LOGIC;
           cin3 : in  STD_LOGIC;
           Sout : out  STD_LOGIC);
end S_Machine;

architecture Behavioral of S_Machine is
type statetype is (zero, one, two, three);
signal state: statetype;
begin
--state register
   process(clk, reset ) begin
   if (reset = '0') then state <= zero;
   elsif rising_edge(clk) then
   case (state) is
   when zero =>
   if (cin1='0') then state<=one;
   elsif(cin1='1') then state<=zero;
   end if;
   when one =>
```



```vhdl
   if (cin2='0') then state<=two;
   elsif(cin2='1') then state<=one;
   end if;
   when two =>
   if (cin3='0') then state<=three;
   elsif (cin3='1') then state<=two;
   end if;
   when three =>
   state <= three;
   end case;
   end if;
   end process;
   -----output logic
   Sout <='1' when state=three else'0';
end Behavioral;
```

User Constraint File



```
###Clock
   NET "clk"   LOC = P129 | IOSTANDARD = LVTTL | PERIOD = 12 MHz;

###########output LED
   NET "Sout"  LOC = P46 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST;

############Input push buttons
   NET "reset" LOC = P80 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;

   NET "cin1"  LOC = P79 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
   NET "cin2"  LOC = P78 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
   NET "cin3"  LOC = P77 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;
```

```
###Clock

	NET "clk"	LOC = P129 | IOSTANDARD = LVTTL | PERIOD = 12 MHz;

############Input push buttons

	NET "reset"	LOC = P80 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;

	NET "cin1"	LOC = P79 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;

	NET "cin2"	LOC = P78 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;

	NET "cin3"	LOC = P77 | IOSTANDARD = LVTTL | DRIVE = 8 | SLEW = FAST | PULLUP;

###########output LED

	NET "Sout"	LOC = P46 | IOSTANDARD = LVTTL | DRIVE = 8| SLEW = FAST;
```