IMPLEMENTATION AND VERIFICATION VHDL MODEL

OF STANDARD PLAYER CONTROL DEVICE

**1 Objective**

Practical application of the hardware description language on the example of the control device of a standard player. Modeling the behavioral model, verification and obtaining the final result, depicted on the waveform.

**2 Working process**

**2.1 Device interface**

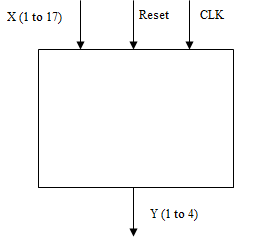
****

Figure 2.1 – Control device interface

**2.2 Working principle**

According to the interface described above, the input signals are:

* X (1 to 17) - informative signals. A vector containing the transition conditions between states.
* Reset is the signal used to transition the device to its initial state.
* CLK - sync signal, (the device operates on the rising edge).

The output signal is the vector Y (1 to 4), which determines in which mode the machine is located.

The device contains five states (modes):

• Stop - initial state;

* Play - start and playback;
* Pause - stop playback at the moment;
* Back - rewind;
* Forward - fast forward.

While device in a certain mode, it changes by pressing the "button" (informative signal).

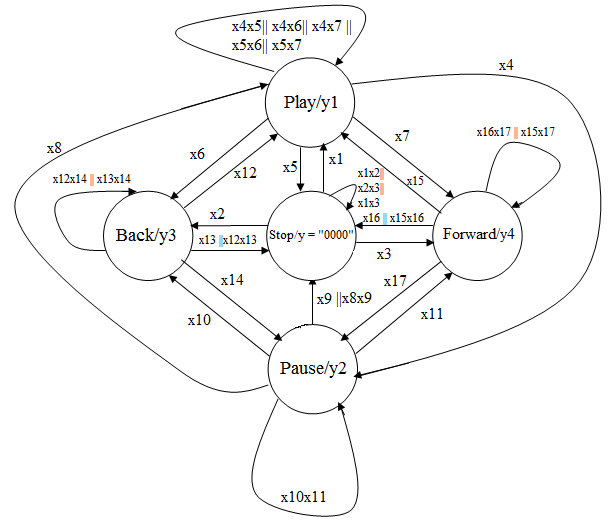
****

Figure 2.2 – Control device transition graph

When two "buttons" are pressed, the machine can change the operating mode, and can leave it, if the "buttons" pressed are Play and Stop. When you press the data of two "buttons" the machine goes into Stop mode, since this mode has priority. Otherwise, the machine is in the current mode.

**2.3 Signal states assigning**

This transition graph reflects the principle of operation of the Moore control automaton.

The states are the modes of operation of the player.

Informative signals are the 17-bit vector x, which each of bit operates with logic conditions when changing from the current state to the following:

X1 - condition of transition from Stop to Play;

X2 is the transition condition from Stop to Back;

X3 - the transition condition from Stop to Forward;

X1X2 OR X2X3 OR X1X3 is a condition for saving the Stop state when two keys are simultaneously pressed, Play-Back or Back-Forward, Play-Forward;

X4 - the transition condition from Play to Pause;

X5 - the condition of transition from Play to Stop;

X6 - the transition condition from Play to Back;

X7 - the condition of transition from Play to Forward;

X4X5 OR X4X6 OR X4X7 OR X5X6 OR X5X7 is a condition for saving the state when two buttons are simultaneously pressed, Pause-Stop or Pause-Back or Pause-Forward, or Stop-Back, or Stop-Forward.

X8 - the transition condition from Pause to Play;

X9 OR X8X9 - the transition condition from Pause to Stop;

X10 - the transition condition from Pause to Back;

X11 - the transition condition from Pause to Forward;

X10X11 - a condition of preservation of a state at simultaneous pressing of two buttons Back-Forward;

X12 - the transition condition from Back to Play;

X13 OR X12X13 - the transition condition from Back to Stop;

X14 - the transition condition from Back to Pause;

X12X14 OR X13X14 - the condition for saving the state by simultaneously pressing two buttons Play-Pause or Stop-Pause;

X15 - transition condition from Forward to Play;

X16 OR X15X16 - transition condition from Forward Stop;

X17 - the transition condition from Forward to Pause;

X15X17 OR X16X17 is a condition for saving the state by simultaneously pressing two Play-Pause or Stop-Pause buttons.

The control signals are a 4-bit vector y, each bit of which indicates the resulting transition (the current state after the transition).

Y = 0000 - state of Stop (initial state);

Y = 0001 - Play status;

Y = 0010 is the Pause state;

Y = 0100 - state of Back;

Y = 1000 - the Forward state;

**2.4 Description of the behavioral model**

This model corresponds to the principle of describing the Moore control automaton. Each state has several transitions and, depending on the informative signal, the corresponding transition time arises.

Listing 2.1 – Example of a transition from a Stop state to a Play condition

when stop=> if x(1) = '1' then state<= play;

end if;

Also, it should be noted that the description of the model provides for pressing two buttons, the result of which is to save the state of the control device.

Listing 2.2 – Example of simultaneous pressing Back and Forward in the Pause state

if (x(10) = '1' and x(11) = '1') then state <= pause;

end if;

Also need to note that there is a slight priority of pressing Play and Stop. In any state, when you press these two buttons, the result is the transition of the machine to the initial Stop state.

Listing 2.3 – Example of simultaneous pressing Play and Stop in the Pause state

if x(9) = '1' or (x(8) and x(9)) = '1' then state <= stop;

end if;

A complete description of the control device model is given below.

Listing 2.4 – Description of the player control device behavioral model.

library IEEE; use IEEE.STD\_LOGIC\_1164.all;

entity player is port( CLK : in STD\_LOGIC;

RESET : in STD\_LOGIC;

x : in STD\_LOGIC\_VECTOR(1 to 17);

y : out STD\_LOGIC\_VECTOR(1 to 4)

);

end player;

architecture workmode of player is

type TState is(play, pause, stop, forward, back);

signal State: TState;

begin process(CLK,RESET)is

begin if RESET='0' then state<=stop;

elsif rising\_edge(CLK) then

case state is

when stop=> if x(1) = '1' then state<= play;

end if;

if x(2) = '1' then state <=back;

end if;

if x(3) = '1' then state <=forward;

end if;

if (x(1) and x(2)) = '1' or (x(2) and x(3)) = '1' or (x(1) and x(3)) = '1' then state <= stop;

end if;

when play=> if x(4) = '1' then state <=pause;

end if;

if x(5) = '1' then state <=stop;

end if;

if x(6) = '1' then state <=back;

end if;

if x(7) = '1' then state <=forward;

end if;

if ((x(4) and x(5)) or (x(4)and x(6))

or (x(4) and x(7)) or (x(5) and x(6))

or (x(5) and x(7))) = '1' then

state <= play;

end if;

when pause=> if x(8)='1' then state <= play;

end if;

if x(9) = '1' or (x(8) and x(9)) = '1' then

state <= stop;

end if;

if x(10) = '1' then state <= back;

end if;

if x(11) = '1' then state <= forward;

end if;

if (x(10) = '1' and x(11) = '1') then

state <= pause;

end if;

when back=> if x(12)='1' then state <=play;

end if;

if x(13) = '1' or (x(12) and x(13)) = '1' then state <=stop;

end if;

if x(14) = '1' then state <=pause;

end if;

if((x(13) and x(14)) or (x(12) and x(14))) = '1' then

state <= back;

end if;

when forward=> if x(15)='1' then state <=play;

end if;

if x(16) = '1' or (x(15) and x(16)) = '1' then state <=stop;

end if;

if x(17) = '1' then state <=pause;

end if;

if((x(16) and x(17)) or (x(15) and x(17))) = '1' then

state <= forward;

end if;

end case;

end if;

end process;

process(state)is

begin

y <="0000";

case state is

when stop=> Y <="0000";

when play=> Y <="0001";

when pause=> Y <="0010";

when back=> Y <="0100";

when forward=> Y <="1000";

end case;

end process;

end ;

Modeling and operability of the device is presented below in Figure 2.3.

This device is tested for the efficiency of the transitions and when you simultaneously press Play and Stop in the rewind mode (Back).

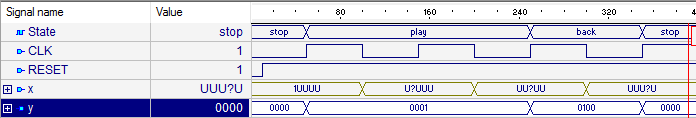


Figure 2.3 – Waveform of the control device behavioral model

**Note**. The signal signals should be looked at in Figure 2.2. Simultaneous operation of all bits of the vector x is not used.

**2.5 Verification of the device**

Verification of the device is the result of the performance of certain actions provided for in the behavioral model. For example, the simultaneous pressing of two buttons, shown in Listing 2.5.

Listing 2.5 – Example of the description of pressing two buttons in verification.

--no back-forward on stop

x(2) <= '1'; x(3) <= '1'; wait for 50 ns;

Complete description of the verification is shown in Listing 2.6

Listing 2.6 – Control device verification

library ieee;

use ieee.std\_logic\_1164.all;

-- Add your library and packages declaration here ...

entity player\_tb is

end player\_tb;

architecture TB\_ARCHITECTURE of player\_tb is

-- Component declaration of the tested unit

component player

port(

CLK : in STD\_LOGIC ;

RESET : in STD\_LOGIC;

x : in STD\_LOGIC\_VECTOR(1 to 17);

y : out STD\_LOGIC\_VECTOR(1 to 4) );

end component;

-- Stimulus signals - signals mapped to the input and inout ports of tested entity

signal CLK : STD\_LOGIC;

signal RESET : STD\_LOGIC:= '0' ;

signal x : STD\_LOGIC\_VECTOR(1 to 17);

-- Observed signals - signals mapped to the output ports of tested entity

signal y : STD\_LOGIC\_VECTOR(1 to 4);

-- Add your code here ...

begin

-- Unit Under Test port map

UUT : player

port map (

CLK => CLK,

RESET => RESET,

x => x,

y => y

);

clk\_process: process

begin

CLK <= '0';

wait for 50 ns;

CLK<='1';

wait for 50 ns;

wait for 0 ns;

end process;

reset\_process: process

begin

wait for 10 ns;

RESET <= '1';

wait for 10 ns;

end process;

two\_buttons:process

begin

--no back-forward on stop

x(2) <= '1'; x(3) <= '1'; wait for 50 ns;

--from stop to forward

x(2) <= '0'; wait for 10 ns;

-- no play pause

x(15) <= '1'; x(17) <= '1'; wait for 10 ns;

--from forward to stop

x(17) <= '0'; x(16) <= '1'; wait for 10 ns;

end process;

end TB\_ARCHITECTURE;

configuration TESTBENCH\_FOR\_player of player\_tb is

for TB\_ARCHITECTURE

for UUT : player

use entity work.player(workmode);

end for;

end for;

end TESTBENCH\_FOR\_player;

Modeling and operability of the device is presented below in Figure 2.4.This verification displays the behavior of the model by simultaneously pressing two buttons in different states and transitioning from the Forward state to the initial Stop state.

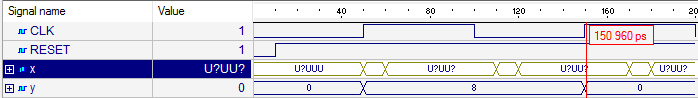


Figure. 2.4 – Waveform of verification

Detailed description of time intervals on the waveform:

0 ns - 50 ns: the machine is in the initial state (Stop);

50 ns - 100 ns: when a leading edge arrives at the sync signal and pressing X3 = 1 (transition from the Stop state to the Forward state), the transition to the Forward state occurs;

100 ns - 150 ns: change the front on the sync signal (trailing edge) - the machine is in the same state even when you press different keys;

150 ns - 200 ns: when a leading edge arrives at the sync signal and pressing X15 = 1 and X16 = 1, the machine returns to the initial state (when the two Play-Stop keys are pressed, the player goes to the initial mode from any state).

**Conclusions**: during the execution of the calculated graphic task, theoretical knowledge based on the principle of operation of the Moore control automata, as well as its language description, were used. In practical application of knowledge, a model of the control device of a standard player is obtained, which includes a description of the behavioral model of the device and the verification of its actions under certain conditions. When simulating the description data, the result is the behavior of the models (waveform) with definite given informative signals, both at the moment of transition from state to state, return to the initial state, and with simultaneous pressing of two keys.