РЕАЛІЗАЦІЯ І ВЕРИФІКАЦІЯ VHDL-МОДЕЛІ ПРИСТРОЇ УПРАВЛІННЯ СТАНДАРТНИМ ПЛЕЙЄРОМ

**1 Мета роботи**

Практичне застосування мови опису апаратури на прикладі пристрою керування стандартним плеєром. Моделювання поведінкової моделі, верифікації, а також отримання підсумкового результату, зображеного на waveform.

**2 Хід роботи**

**2.1 Інтерфейс пристрою**

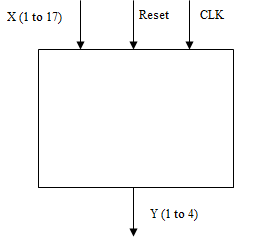
****

Рисунок 2.1 – Інтерфейс керуючого пристрою

**2.2 Принцип роботи**

Згідно з інтерфейсом, описаним вище, сигналами на вхід є:

• Х(1 to 17) – інформаційні сигнали. Вектор містить умови переходу між станами.

• Reset – сигнал, який використовується для переходу пристрою до початкового стану.

• СLK – синхросигнал, (пристрій працює на передньому фронті).

Сигналом на виході є вектор Y(1 to 4), який визначає, в якому режимі знаходиться автомат.

Пристрій містить п'ять станів (режимів):

• Stop – початковий стан;

• Play – старт та програш;

• Pause – зупинка відтворення зараз;

• Back – перемотування назад;

• Forward – перемотування вперед;

Пристрій, перебуваючи у певному режимі, змінює його за допомогою натискання «кнопки» (повідомчого сигналу).

При натисканні двох "кнопок" автомат може змінювати режим роботи, а може і залишити його, якщо натиснутими "кнопками" є Play і Stop. При натисканні двох «кнопок» автомат переходить у режим Stop, оскільки цей режим має пріоритет. Інакше – автомат перебуває у поточному режимі.

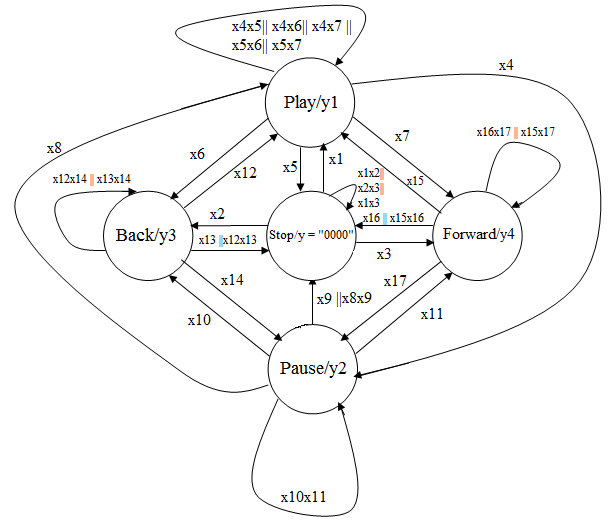
****

Рисунок 2.2 – Граф переходів керуючого пристрою

**2.3 Призначення станів сигналів**

Станом є режими роботи плеєра.

Інформаційними сигналами є 17-розрядний вектор х, кожен з біт якого оперує логічними умовами при переході з поточного стану в наступне:

Х1 – умова переходу зі Stop у Play;

Х2 - умова переходу зі Stop в Back;

Х3 - умова переходу зі Stop в Forward;

X1X2 АБО X2X3 АБО X1X3 – умова збереження стану Stop при одночасному натисканні двох клавіш, Play-Back або Back\_Forward, Play-Forward;

X4 - умова переходу з Play в Pause;

X5 - умова переходу з Play в Stop;

X6 – умова переходу з Play до Back;

X7 – умова переходу з Play до Forward;

X4X5 АБО X4X6 АБО X4X7 АБО X5X6 АБО X5X7 – умова збереження стану при одночасному натисканні двох кнопок, Pause-Stop або Pause-Back або Pause-Forward, або Stop-Back, або Stop-Forward.

X8 – умова переходу з Pause у Play;

X9 АБО X8X9 – умова переходу з Pause у Stop;

X10 - умова переходу з Pause в Back;

X11 – умова переходу з Pause до Forward;

X10X11 – умова збереження стану за одночасного натискання двох кнопок Back-Forward;

X12 – умова переходу з Back у Play;

X13 АБО X12X13 - умова переходу з Back в Stop;

X14 – умова переходу з Back у Pause;

X12X14 АБО X13X14 – умова збереження стану при одночасному натисканні двох кнопок Play-Pause або Stop-Pause;

X15 – умова переходу з Forward у Play;

X16 АБО X15X16 – умова переходу з Forward Stop;

X17 – умова переходу з Forward у Pause;

X15X17 АБО X16X17 – умова збереження стану при одночасному натисканні двох кнопок Play-Pause або Stop-Pause.

Керуючими сигналами є 4-розрядний вектор, кожен з біт якого вказує на результуючий перехід (поточний стан після виконання переходу).

Y = 0000 - стан Stop (початковий стан);

Y = 0001 - стан Play;

Y = 0010 – стан Pause;

Y = 0100 - стан Back;

Y = 1000 – стан Forward;

**2.4 Опис поведінкової моделі**

Ця модель відповідає принципу опису управляючого автомата Мура.

Кожен стан має кілька переходів і, залежно від інформаційного сигналу, виникає відповідний момент переходу.

Лістинг 2.1 – приклад переходу зі стану Stop у стан Play

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

end if;

Також слід зазначити, що в описі моделі передбачено натискання двох кнопок, результатом якого є збереження стану керуючого пристрою.

Лістинг 2.2 – приклад одночасного натискання Back та Forward у стані Pause

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

end if;

Також слід зазначити, що є невеликий пріоритет натискання Play і Stop. У будь-якому стані при натисканні двох даних кнопок результатом є перехід автомата до початкового стану Stop

Лістинг 2.3 – приклад одночасного натискання Play та Stop у стані Pause

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

end if;

Повний опис моделі керуючого пристрою наведено нижче.

Лістинг 2.4 - Опис поведінкової моделі пристрою керування плеєром

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 ;

Моделювання та працездатність пристрою представлено нижче на малюнку 2.3

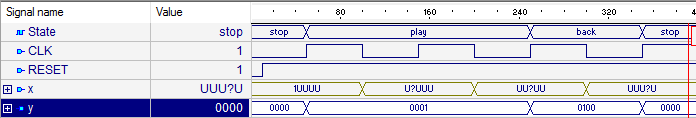
Цей пристрій перевірено на працездатність переходів та при одночасному натисканні Play і Stop у режимі перемотування назад (Back).

Рисунок 2.3 – Waveform поведінкової моделі керуючого пристрою

Примітка. Поінформаційні сигнали слід подивитися на рис. 2.2, так як не використано одночасну роботу всіх біт вектора х.

**2.5 Верифікація пристрою**

Верифікація пристрою є результатом виконання певних дій, передбачених у поведінковій моделі. Наприклад, одночасне натискання двох кнопок, представлене у лістингу 2.5

Лістинг 2.5 – приклад опису натискання двох кнопок у верифікації

--no back-forward on stop

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

Повний опис верифікації наведено у лістингу 2.6

Лістинг 2.6 - Верифікація пристрою керування

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;

Моделювання та працездатність пристрою представлено нижче на рисунку 2.4

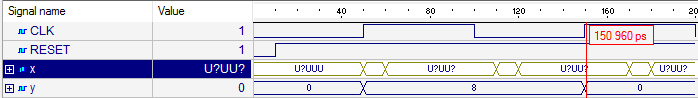
Ця верифікація відображає поведінку моделі при одночасному натисканні двох кнопок у різних станах та перехід зі стану Forward у початковий стан Stop.

Рисунок. 2.4 - Waveform верифікації

Детальний опис інтервалів часу на waveform:

0 ns – 50 ns: автомат перебуває у початковому стані (Stop);

50 ns - 100 ns: при надходженні переднього фронту на синхросигнал і натисканні Х3 = 1 (перехід зі стану Stop у стан Forward) виникає перехід у стан Forward;

100 ns – 150 ns: зміна фронту на синхросигналі (задній фронт) – автомат перебуває у тому стані навіть за натисканні різних кнопок;

150 ns - 200 ns: при надходженні переднього фронту на синхросигнал і натисканні Х15 = 1 і Х16 = 1 автомат повертається в початковий стан (при натисканні двох клавіш Play-Stop плеєр переходить у початковий режим з будь-якого стану).

**Висновки**: в ході виконання розрахункового графічного завдання використано теоретичні знання за принципом роботи автоматів Мура, що управляють, а також його мовний опис. При практичному застосуванні знань отримано модель пристрою керування стандартним плеєром, що містить у собі опис поведінкової моделі пристрою та верифікації його дій за певних умов. При моделюванні даних описів результатом є поведінка моделей (waveform) при визначено заданих інформаційних сигналів як у момент переходу зі стану в стан, повернення в початковий стан, так і при одночасному натисканні двох клавіш.