РЕАЛИЗАЦИЯ И ВЕРИФИКАЦИЯ VHDL-МОДЕЛИ УСТРОЙСТВА УПРАВЛЕНИЯ СТАНДАРТНЫМ ПЛЕЙЕРОМ

**1 Цель работы**

Практическое применение языка описания аппаратуры на примере устройства управления стандартным плейером. Моделирование поведенческой модели, верификации, а также получение итогового результата, изображенного на waveform.

**2 Ход работы**

**2.1 Интерфейс устройства**

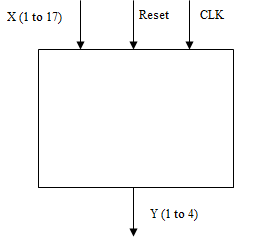
****

Рисунок 2.1 – Интерфейс управляющего устройства

**2.2 Принцип работы**

Согласно интерфейсу, описаному выше, сигналами на вход являются:

* Х(1 to 17) – осведомительные сигналы. Вектор, содержащий условия перехода между состояниями.
* Reset – сигнал, служащий для перехода устройства в начальное состояние.
* СLK – синхросигнал, (устройство работает по переднему фронту).

Сигналом на выходе является вектор Y(1 to 4), который определяет в каком режиме находится автомат.

Устройство содержит пять состояний (режимов):

* Stop – начальное состояние;
* Play – старт и проигрыш;
* Pause – остановка воспроизведения в текущий момент;
* Back – перемотка назад;
* Forward – перемотка вперед;

Устройство, находясь в определенном режиме изменяет его с помощью нажатия «кнопки» (осведомительного сигнала).

При нажатии двух «кнопок» автомат может изменять режим работы, а может и оставить его, если нажатыми «кнопками» являются Play и Stop. При нажатии данных двух «кнопок» автомат переходит в режим Stop, так как этот режим имеет приоритет. Иначе – автомат находится в текущем режиме.

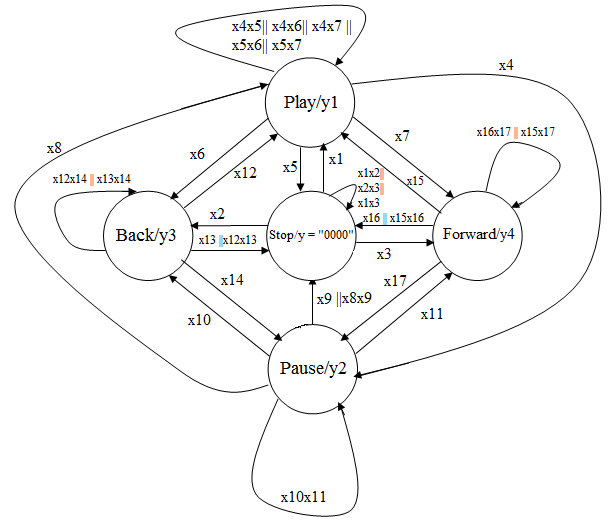
****

Рисунок 2.2 – Граф переходов управляющего устройства

**2.3 Назначение состояний сигналов**

Данный граф переходов отображает принцип работы управляющего автомата Мура.

Состояниями являются режимы работы плейера.  
 Осведомительными сигналами является 17-разрядный вектор х, каждый из бит которого оперирует логическими условиями при переходе из текущего состояния в последующее:

Х1 – условие перехода из Stop в Play;

Х2 – условие перехода из Stop в Back;

Х3 – условие перехода из Stop в Forward;

X1X2 ИЛИ X2X3 ИЛИ X1X3 – условие сохранения состояния Stop при одновременном нажатии двух клавиш, Play-Back или Back\_Forward, Play-Forward;

X4 – условие перехода из Play в Pause;

X5 – условие перехода из Play в Stop;

X6 – условие перехода из Play в Back;

X7 – условие перехода из Play в Forward;

X4X5 ИЛИ X4X6 ИЛИ X4X7 ИЛИ X5X6 ИЛИ X5X7 – условие сохранения состояния при одновременном нажатии двух кнопок, Pause-Stop или Pause-Back или Pause-Forward, или Stop-Back, или Stop-Forward.

X8 – условие перехода из Pause в Play;

X9 ИЛИ X8X9 – условие перехода из Pause в Stop;

X10 – условие перехода из Pause в Back;

X11 – условие перехода из Pause в Forward;

X10X11 – условие сохранения состояния при одновременном нажатии двух кнопок Back-Forward;

X12 – условие перехода из Back в Play;

X13 ИЛИ X12X13 – условие перехода из Back в Stop;

X14 – условие перехода из Back в Pause;

X12X14 ИЛИ X13X14 – условие сохранения состояния при одновременном нажатии двух кнопок Play-Pause или Stop-Pause;

X15 – условие перехода из Forward в Play;

X16 ИЛИ X15X16 – условие перехода из Forward Stop;

X17 – условие перехода из Forward в Pause;

X15X17 ИЛИ X16X17 – условие сохранения состояния при одновременном нажатии двух кнопок Play-Pause или Stop-Pause.

Управляющими сигналами является 4-разрядный вектор у, каждый из бит которого указывает на результирующий переход (текущее состояние после выполнения перехода).

Y = 0000 – состояние Stop (начальное состояние);

Y = 0001 – состояние Play;

Y = 0010 – состояние Pause;

Y = 0100 – состояние Back;

Y = 1000 – состояние Forward;

**2.4 Описание поведенческой модели**

Данная модель соответствует принципу описания управляющего автомата Мура.

Каждое состояние имеет несколько переходов и в зависимости от осведомительного сигнала возникает соответствующий момент перехода.

Листинг 2.1 – пример перехода из состояния Stop в состоянние Play

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

end if;

Также, следует отметить, что в описании модели предусмотрено нажатие двух кнопок, результатом которого является сохранение состояния управляющего устройства.

Листинг 2.2 – пример одновременного нажатия Back и Forward в состоянии Pause

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

end if;

Также следует заметить, что существует небольшой приоритет нажатия Play и Stop. В любом состоянии при нажатии двух данных кнопок результатом является переход автомата в начальное состояние Stop

Листинг 2.3 – пример одновременного нажатия Play и Stop в состоянии Pause

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

end if;

Полное описание модели управляющего устройства представлено ниже.

Листинг 2.4 – Описание поведенческой модели устройства управления пленером

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 ;

Моделирование и работоспособность устройства представлено ниже на рисунке 2.3

Данное устройство проверено на работоспособность переходов и при одновременном нажатии Play и Stop в режиме перемотки назад (Back).

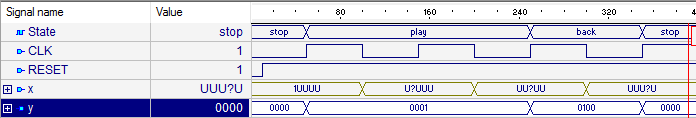


Рисунок 2.3 – Waveform поведенческой модели управляющего устройства

Примечание. Осведомительные сигналы следует посмотреть на рисунке 2.2, т.к. не использована одновременная работа всех бит вектора х.

**2.5 Верификация устройства**

Верификация устройства является результатом выполнения определенных действий, предусмотренных в поведенческой модели. Например, одновременное нажатие двух кнопок, представленное в листинге 2.5

Листинг 2.5 – Пример описания нажатия двух кнопок в верификации

--no back-forward on stop

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

Полное описание верификации приведено в листинге 2.6

Листинг 2.6 – Верификация устройства управления

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;

Моделирование и работоспособность устройства представлено ниже на рисунке 2.4

Данная верификация отображает поведение модели при одновременном нажатии двух кнопок в разных состояниях и переход из состояния Forward в начальное состояние Stop.

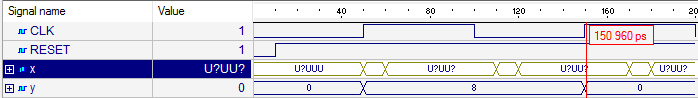


Рисунок. 2.4 – Waveform верификации

Подробное описание интервалов времени на waveform:

0 ns – 50 ns: автомат находится в начальном состоянии (Stop);

50 ns – 100 ns: при поступлении переднего фронта на синхросигнал и нажатии Х3 = 1 (переход из состояния Stop в состояние Forward) возникает переход в состояние Forward;

100 ns – 150 ns: изменение фронта на синхросигнале (задний фронт) – автомат находится в том же состоянии даже при нажатии разных клавиш;

150 ns – 200 ns: при поступлении переднего фронта на синхросигнал и нажатии Х15 = 1 и Х16 = 1 автомат возвращается в начальное состояние (при нажатии двух клавиш Play-Stop плеер переходит в начальный режим из любого состояния).

Выводы: в ходе выполнения расчетного графического задания использованы теоретические знания по принципу работы управляющих автоматов Мура, а также его языковое описание. При практическом применении знаний получена модель устройства управления стандартным плейером, которая заключает в себе описание поведенческой модели устройства и верификации его действий при определенных условиях. При моделировании данных описаний результатом является поведение моделей (waveform) при определенно заданных осведомительных сигналов как в момент перехода из состояния в состояние, возврата в начальное состояние, так и при одновременном нажатии двух клавиш.