НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №7

з дисципліни **«**Технології проектування комп’ютерних систем**»**

Виконав:

студент групи ІО-53, ФІОТ

Гріщенко Костянтин Сергійович

Перевірив:

Сергієнко Анатолій Михайлович

Київ - 2018 р.

**Тема:** Проектування ядра мікропроцесору.

**Мета та основні завдання** **роботи**: оволодіти знаннями і практичними навичками з проектування таких складних обчислювальних блоків з програмним управлінням, як ядро мікропроцесору (CPU). Лабораторна робота також служить для оволодіння навичками програмування і налагодження опису CPU на мові VHDL.

**Завдання на лабораторну роботу:** розробити RISC-подібне ядро мікропроцесору, використовуючи модифікації компонентів, які були розроблені під час виконання попередніх лабораторних робіт даного курсу.

**Виконання:**

--^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

-- Title : RISC\_CPU

-- Design : RISC\_ST\_CORE

-- File : RISC\_CPU.vhd

-- Author : Anatoli Sergyienko aser@comsys.ntu-kpi.kiev.ua

-- Version 1.0

-- Description: Ïðèìåð ïðîöåññîðà, ðåàëèçîâàííîãî íà áàçå RISC\_ST\_CORE

-- Warning :This project is intended for teaching and scientifical purposes only,

-- the use for another purposes is prohibited.

-- This file can be copied, edited, but the header has to be unexchanged.

-- (C) ÑÎÐYRIGHT A.Sergyienko

-- Modification history:

-- Changed word size to 24bits

--^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

library IEEE;

use IEEE.std\_logic\_1164.all,work. RISC\_lib.all;

entity RISC\_CPU is port(CLK : in std\_logic; --ñèíõðîâõîä

RST : in std\_logic; --ñáðîñ

INTREQ: in std\_logic\_vector(3 downto 0); --çàïðîñû ïðåðûâàíèÿ

PORT0\_I: in std\_logic\_vector(23 downto 0);--âõîäíîé ïîðò

PORT0\_O: out std\_logic\_vector(23 downto 0));--âûõîäíîé ïîðò

end RISC\_CPU;

architecture RISC\_CPU\_SYNT of RISC\_CPU is

component RISC\_ST\_CORE is port(CLK :in std\_logic;

RST : in std\_logic;

ENA : in std\_logic; --ãîòîâíîñòü äàííûõ â DRAM

INSTR : in WORD; --êîìàíäà èç PROM

INTREQ : in std\_logic\_vector( N\_INTSRC downto 0);--çàïðîñû íà ïðåðûâàíèå

DATAI : in WORD; --äàííoå èç DRAM

ADDRI : out WORD; --àäðåñ êîìàíäû

RE : out std\_logic; --ñèãíàë ÷òåíèÿ èç DRAM

WE : out std\_logic; --ñèãíàë çàïèñè â DRAM

DATAO : out WORD; --äàííoå â DRAM

ADDRD : out WORD ); --àäðåñ äàííîãî

end component;

component DRAM is port(CLK :in std\_logic;

RST : in std\_logic;

RE: in std\_logic;

WE: in std\_logic;

ADDRD : in WORD;

DATAI: in WORD;

RDY: out std\_logic;

DATAO : out WORD );

end component ;

component PRAM is port( CLK :in std\_logic;

RST : in std\_logic;

ENA: in std\_logic;

ADDRI : in WORD;

INSTR : out WORD );

end component;

component PORT0 is port( CLK :in std\_logic;

RST : in std\_logic;

DATA\_I : in WORD; --øèíà äàííûõ

ADDRD : in WORD; --Øèíà àäðåñà

WE: in std\_logic; --ðàçðåøåíèå çàïèñè

RE: in std\_logic; --ðàçðåøåíèå ÷òåíèÿ

RDY:out std\_logic; -- ãîòîâíîñòü ïîðòà

PORT0\_I : in WORD; -- Âõîä ïîðòà

DATA\_O : out WORD; --øèíà äàííûõ

PORT0\_O : out WORD ); --Âûõîä ïîðòà

end component;

signal rdy,re,we,ena:std\_logic;

signal instr:WORD;

signal datai,datao:WORD;

signal addri:WORD;

signal addrd:WORD;

begin

U\_CORE: RISC\_ST\_CORE port map( CLK =>CLK,

RST=>RST,

ENA =>rdy,

INSTR =>instr,

INTREQ =>INTREQ,

DATAI =>datai,

ADDRI =>addri,

RE =>re,

WE =>we,

DATAO =>datao,

ADDRD =>addrd );

U\_DRAM: DRAM port map(CLK=>CLK,

RST=>RST,

RE=>RE,

WE=>WE,

ADDRD =>addrd,

DATAI=>datao,

RDY=>rdy,

DATAO=>datai );

U\_PRAM: PRAM port map( CLK =>CLK,

RST=>RST,

ENA=>rdy,

ADDRI =>addri,

INSTR =>instr);

U\_PORT0:PORT0 port map( CLK=>CLK,

RST =>RST,

DATA\_I =>datao,

ADDRD =>addrd,

WE=>we,

RE=>re,

RDY=>rdy,

PORT0\_I=>PORT0\_I,

DATA\_O =>datai,

PORT0\_O =>PORT0\_O);

end RISC\_CPU\_SYNT;

**Тестовий стенд:**

--^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

-- Title : RISC\_lib

-- Design : RISC\_ST\_CORE

-- File : RISC\_lib.vhd

-- Author : Anatoli Sergyienko aser@comsys.ntu-kpi.kiev.ua

-- Version 1.0

-- Description: Èñïûòàòåëüíûé ñòåíä äëÿ RISC\_ST\_CPU ïðè òåñòèðîâàíèè ïðåðûâàíèé

-- Warning :This project is intended for teaching and scientifical purposes only,

-- the use for another purposes is prohibited.

-- This file can be copied, edited, but the header has to be unexchanged.

-- (C) ÑÎÐYRIGHT A.Sergyienko

-- Modification history:

-- Changed word size to 24bits

--^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

library ieee;

use ieee.std\_logic\_1164.all, work. RISC\_lib.all;

entity risc\_cpu\_tb is generic(TCLK:time:=20 ns);--ïåðèîä ñèíõðîñåðèè

end risc\_cpu\_tb;

architecture TB\_ARCHITECTURE of risc\_cpu\_tb is

component risc\_cpu port( CLK : in std\_logic;

RST : in std\_logic;

INTREQ : in std\_logic\_vector(3 downto 0);

PORT0\_I : in std\_logic\_vector(23 downto 0);

PORT0\_O : out std\_logic\_vector(23 downto 0) );

end component;

signal CLK : std\_logic:='0';

signal RST : std\_logic;

signal INTREQ : std\_logic\_vector(3 downto 0):="0000";

signal PORT0\_I : std\_logic\_vector(23 downto 0);

signal PORT0\_O : std\_logic\_vector(23 downto 0);

signal end\_s:boolean;

begin

UUT : risc\_cpu port map (CLK => CLK,

RST => RST,

INTREQ => INTREQ,

PORT0\_I => PORT0\_I,

PORT0\_O => PORT0\_O );

--ñòèìóëèðóþùèå ñèãíàëû

INTREQ(2)<=not INTREQ(2) after 0.5 us when not end\_simulation else '0' ;

INTREQ(1)<=not INTREQ(1) after 0.65 us when not end\_simulation else '0' ;

INTREQ(0)<=not INTREQ(0) after 0.8 us when not end\_simulation else '0' ;

CLK<=not CLK after TCLK/2 when not end\_simulation else '0' ;

RST<='1', '0' after 30 ns;

END\_SIM:process begin

end\_simulation:=false;

wait for 16 us; --âðåìÿ ìîäåëèðîâàíèÿ

end\_simulation:=true;

wait;

end process;

end TB\_ARCHITECTURE;

configuration TESTBENCH\_FOR\_risc\_cpu of risc\_cpu\_tb is

for TB\_ARCHITECTURE

for UUT : risc\_cpu

use entity work.risc\_cpu(risc\_cpu\_synt);

end for;

end for;

end TESTBENCH\_FOR\_risc\_cpu;

**Результати симуляції:**

