**Министерствообразования и наукиРоссийскойФедерации**

Федеральноегосударственноебюджетноеобразовательноеучреждение

высшегопрофессиональногообразования

**«Владимирскийгосударственныйуниверситет**

**имениАлександраГригорьевича и НиколаяГригорьевичаСтолетовых»**

**(ВлГУ)**

**Кафедра «Вычислительнаятехника»**

Лабораторнаяработа № 4

подисциплине

«САПР ПЛИС и ИМС»

**Выполнил:**

**ст. гр. ВТм - 112**

**А.Х Муна**

**Приняла:**

**С.С. Гладьо**

**Владимир 2012**

**Вариант No. 13.1**

Написать и провеститестированиедвухфункций и двухпроцедур.

1. Функция (процедура) нахождения в двоичнойматрице B строки J с **минимальным**суммарнымчислом S единиц.

Выдатьномер J строки исуммарноечисло S единиц.

Типматрицыразмерности (**NxM)** определить в пакете.

**Цельработа:**

Изучат функции и процедуры на языке VHDL.

**VHDL code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.all;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

package Lab4\_pkg is

-- type<new\_type> is

-- record

-- <type\_name> : std\_logic\_vector( 7 downto 0);

-- <type\_name> : std\_logic;

-- end record;

-- Declare functions and procedure

--

-- function<function\_name> (signal <signal\_name> : in <type\_declaration>) return <type\_declaration>;

-- procedure<procedure\_name> (<type\_declaration><constant\_name> : in <type\_declaration>);

--

--myMxN array

type M\_ROW is range 1 to 4;

type N\_COL is range 1 to 4;

type MN\_ARRAY is

array(M\_ROW, N\_COL) of std\_logic\_vector(3 downto 0);

--//my functions

**--Функциянахождения в двоичнойматрице B строки J с --минимальнымсуммарнымчислом S единиц.**

**functionfind\_row\_J\_of\_bin\_matrix(checker:bit) return std\_logic\_vector;**

**--Функциянахождениястроки J с минимальнымсуммарнымчислом S единиц**

**functionnumber\_of\_min\_row(row:bit) return integer;**

end Lab4\_pkg;

package body Lab4\_pkg is

---- Procedure Example

-- procedure<procedure\_name> (<type\_declaration><constant\_name> : in <type\_declaration>) is

--

-- begin

--

-- end<procedure\_name>;

**functionfind\_row\_J\_of\_bin\_matrix(checker:bit) return std\_logic\_vector is**

--//variables declarations

--//variables declarations

variabletemp:std\_logic\_vector(3 downto 0);

variable temp1:std\_logic\_vector(3 downto 0);

variable temp2:std\_logic\_vector(3 downto 0);

--variableCIN:std\_logic; --for carry bit

variablematrix:MN\_ARRAY;

variable A: integer:= 0; --init A

variable B: integer:= 0; --init B

variable S1,S2,S3,S4: std\_logic\_vector(3 downto 0);

begin

--ifsomeMtx = "00" then

if checker = '1' then

temp := (others =>'0'); --//check what is being returned

temp1:= (others =>'0');

temp2:= (others =>'0');

--//1st row

matrix(1, 1) :="0001";

matrix(1, 2) :="0010";

matrix(1, 3) :="0011";

matrix(1, 4) := "0100";

--//second row

matrix(2, 1) :="0110";

matrix(2, 2) :="0111";

matrix(2, 3) :="0001";

matrix(2, 4) := "0010";

--//3rd row

matrix(3, 1) :="0001";

matrix(3, 2) :="0001";

matrix(3, 3) :="0001";

matrix(3, 4) := "0000";

--//4th row

matrix(4, 1) :="0100";

matrix(4, 2) :="0101";

matrix(4, 3) :="0110";

matrix(4, 4) := "0111";

--//adding SUMINT <= ('0' & A) + ('0' & B) + ("0000" & CIN);

S1 := ('0' & matrix(1,1))+ ('0'& matrix(1,2))+ ('0'& matrix(1,3))+ ('0' & matrix(1,4));

S2 := ('0' & matrix(2,1))+ ('0'& matrix(2,2))+ ('0'& matrix(2,3))+ ('0' & matrix(2,4));

S3 := ('0' & matrix(3,1))+ ('0'& matrix(3,2))+ ('0'& matrix(3,3))+ ('0' & matrix(3,4));

S4 := ('0' & matrix(4,1))+ ('0'& matrix(4,2))+ ('0'& matrix(4,3))+ ('0' & matrix(4,4));

--comparing 1 and 2nd row

if S1 < S2 then

temp1 := S1;

else

temp1 := S2;

end if;

--comparing 3 and 4th row

if S3 < S4 then

temp2 := S3;

else

temp2:= S4;

end if;

--comparing temps

if temp1 < temp2 then

temp := temp1;

return temp;

else

temp :=temp2;

return temp;

end if;

end if; --end checker

endfind\_row\_J\_of\_bin\_matrix;

--//the function for returning the minimum row

**functionnumber\_of\_min\_row(row:bit) return integer is**

--//variables declarations

variabletemp:std\_logic\_vector(3 downto 0);

variable temp1:std\_logic\_vector(3 downto 0);

variable temp2:std\_logic\_vector(3 downto 0);

variablemin\_row:integer;

variablematrix:MN\_ARRAY;

variable S1,S2,S3,S4: std\_logic\_vector(3 downto 0);

begin

--ifsomeMtx = "00" then

temp := (others =>'0'); --//check what is being returned

temp1:= (others =>'0');

temp2:= (others =>'0');

--//1st row

matrix(1, 1) :="0001";

matrix(1, 2) :="0010";

matrix(1, 3) :="0011";

matrix(1, 4) := "0100";

--//second row

matrix(2, 1) :="0110";

matrix(2, 2) :="0111";

matrix(2, 3) :="0001";

matrix(2, 4) := "0010";

--//3rd row

matrix(3, 1) :="0001";

matrix(3, 2) :="0000";

matrix(3, 3) :="0001";

matrix(3, 4) := "0011";

--//4th row

matrix(4, 1) :="0100";

matrix(4, 2) :="0101";

matrix(4, 3) :="0110";

matrix(4, 4) := "0111";

S1 := matrix(1,1)+ matrix(1,2)+ matrix(1,3)+ matrix(1,4);

S2 := matrix(2,1)+ matrix(2,2)+ matrix(2,3)+ matrix(2,4);

S3 := matrix(3,1)+ matrix(3,2)+ matrix(3,3)+ matrix(3,4);

S4 := matrix(4,1)+ matrix(4,2)+ matrix(4,3)+ matrix(4,4);

--comparing 1 and 2nd row

if S1 < S2 and S1 < S3 and S1 < S4 then

min\_row := 1;

returnmin\_row;

else

if S2 < S3 and S3 < S4 then

min\_row := 2;

returnmin\_row;

returnmin\_row;

else

if S3 < S4 then

min\_row := 3;

returnmin\_row;

else

min\_row := 4;

returnmin\_row;

end if;

end if;

end if;

**end function number\_of\_min\_row;**

end Lab4\_pkg;

**Вызов Функцию:**

use work.lab4\_pkg.all

--//

sum\_min := **find\_row\_J\_of\_bin\_matrix(X);**

**min\_row := number\_of\_min\_row(Y);**

**VHDL module Code:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use work.Lab4\_pkg.ALL;

entity LAB4 is

Port ( CLK : in STD\_LOGIC;

DI : in BIT;

MIN\_SUM : out STD\_LOGIC\_VECTOR(3 downto 0);

ROW\_NUMBER : out INTEGER);

end LAB4;

architecture Behavioral of LAB4 is

--variable

variabletemp\_sum: std\_logic\_vector(3 downto 0);

variabletemp\_row\_number:integer;

begin

process(CLK,DI)

begin

if(CLK'event and CLK = '1') then

temp\_sum := find\_row\_J\_of\_bin\_matrix(DI);

temp\_row\_number := number\_of\_min\_row(DI);

end if;

end process;

MIN\_SUM <= temp\_sum;

ROW\_NUMBER <= temp\_row\_number;

end Behavioral;

**Xilinx TestBench**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY LAB4TB IS

END LAB4TB;

ARCHITECTURE behavior OF LAB4TB IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT LAB4

PORT(

CLK : IN std\_logic;

DI : IN BIT;

MIN\_SUM : OUT std\_logic\_vector(3 downto 0);

ROW\_NUMBER : OUT INTEGER

);

END COMPONENT;

--Inputs

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

signal DI : BIT := '0';

--Outputs

signal MIN\_SUM : std\_logic\_vector(3 downto 0);

signal ROW\_NUMBER : INTEGER;

-- Clock period definitions

constantCLK\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: LAB4 PORT MAP (

CLK => CLK,

DI => DI,

MIN\_SUM => MIN\_SUM,

ROW\_NUMBER => ROW\_NUMBER

);

-- Clock process definitions

CLK\_process :process

begin

CLK <= '0';

wait for CLK\_period/2;

CLK <= '1';

wait for CLK\_period/2;

end process;

-- Stimulus process

DI\_process: process

begin

-- hold reset state for 100 ns.

-- wait for 100 ns;

DI <= '0';

wait for CLK\_period\*10;

DI <= '1';

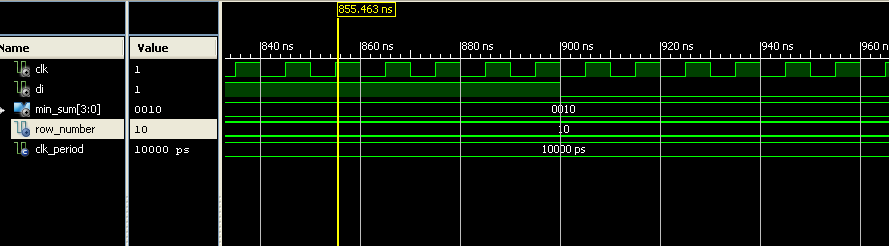
wait for CLK\_period\*10;

end process;

END;

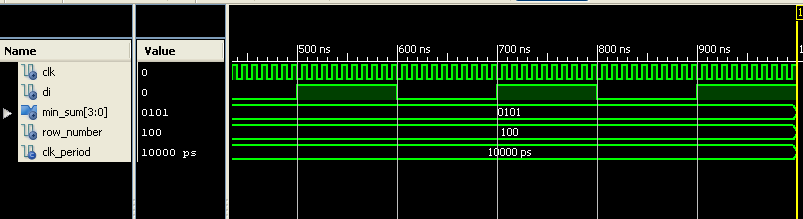
Тест:

|  |  |  |  |
| --- | --- | --- | --- |
| 1111 | 0110 | 1011 | 1000 |
| 0001 | 0000 | 0000 | 0001 |
| 0111 | 0101 | 1101 | 0000 |
| 0101 | 0101 | 0111 | 0111 |

****

**Sum\_min = 2 ,Row\_min = 2**

|  |  |  |  |
| --- | --- | --- | --- |
| 0001 | 0010 | 0011 | 1000 |
| 0110 | 0111 | 0001 | 0010 |
| 0110 | 0101 | 0011 | 0001 |
| 0010 | 0001 | 0000 | 0010 |

****

Sum\_min = 5, Row\_min = 4

Вывод:

В заданном лаборатории работа изучил как реализовать функции на языке VHDL.