Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных средств

Дисциплина: Проектирование цифровых систем на языках описания аппаратуры

ОТЧЕТ

к лабораторной работе №5

на тему

ФУНКЦИИ И ПРОЦЕДУРЫ

|  |  |
| --- | --- |
| Выполнили:  ст. гр. 850702  Турко В. Д. | Проверил:  Санько Н. С. |
|  |  |

Минск 2020

1. Цель работы

Написать и провести тестирование функции и процедуры поиска максимального элемента mij в заданном столбце j матрицы M, элементами которого являются натуральные числа, выбираемые из множества {0, 1, 2, …, 100}.

1. Функция и процедура поиска

LIBRARY IEEE;

USE IEEE.std\_logic\_1164.ALL;

USE IEEE.numeric\_std.ALL;

PACKAGE matrix\_package IS

    CONSTANT interval : TIME := 50 ns;

    TYPE matrix IS ARRAY (NATURAL RANGE <>, NATURAL RANGE <>) OF NATURAL;

    TYPE matrix\_element IS ARRAY (0 TO 2) OF NATURAL;

    FUNCTION findElementFunc (

        vec : MATRIX;

        rows, column : NATURAL) RETURN matrix\_element;

    PROCEDURE findElementProc(

        vec : IN MATRIX;

        value : INOUT NATURAL;

        y : OUT NATURAL;

        rows, column : IN NATURAL);

END PACKAGE matrix\_package;

PACKAGE BODY matrix\_package IS

    FUNCTION findElementFunc (

        vec : MATRIX;

        rows, column : NATURAL) RETURN matrix\_element IS

        VARIABLE element : matrix\_element := (vec(0, column), 0, column);

    BEGIN

        FOR i IN 0 TO rows - 1 LOOP

            IF (vec(i, column) > element(0)) THEN

                element := (vec(i, column), i, column);

            END IF;

        END LOOP;

        RETURN element;

    END FUNCTION findElementFunc;

    PROCEDURE findElementProc(vec : IN MATRIX; value : INOUT NATURAL; y : OUT NATURAL; rows, column : IN NATURAL) IS

    BEGIN

        value := vec(0, column);

        y := 0;

        FOR i IN 0 TO rows - 1 LOOP

            IF (vec(i, column) > value) THEN

                value := vec(i, column);

                y := i;

            END IF;

        END LOOP;

    END findElementProc;

END PACKAGE BODY matrix\_package;

1. Моделирование
2. Тестирующая программа

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

USE ieee.numeric\_std.ALL;

USE ieee.math\_real.ALL;

USE std.textio.ALL;

USE work.matrix\_package.ALL;

ENTITY TEST\_BENCH IS

END TEST\_BENCH;

ARCHITECTURE MainBehavior OF TEST\_BENCH IS

    FUNCTION to\_string (element : INTEGER) RETURN STRING IS

        VARIABLE str : STRING(1 TO 4);

    BEGIN

        CASE element IS

            WHEN 0 => str := "   0";

            WHEN 1 => str := "   1";

            WHEN 2 => str := "   2";

. . .

            WHEN 100 => str := " 100";

            WHEN OTHERS => str := "????";

        END CASE;

        RETURN str;

    END FUNCTION to\_string;

    CONSTANT column : NATURAL := 5;

    SIGNAL test\_array : MATRIX(0 TO 9, 0 TO 7);

    SIGNAL el\_f, y\_f : INTEGER := 0;

    SIGNAL el\_p, y\_p : INTEGER := 0;

    SIGNAL x : INTEGER := column;

BEGIN

    PROCESS

        VARIABLE arr : MATRIX(0 TO 9, 0 TO 7);

        VARIABLE r : real;

        VARIABLE seed1, seed2 : INTEGER := 999;

    BEGIN

        FOR K IN 0 TO 4 LOOP

            FOR i IN 0 TO 9 LOOP

                FOR j IN 0 TO 7 LOOP

                    uniform(seed1, seed2, r);

                    arr(i, j) := INTEGER(round(r \* real(100 + 1) + real(0) - 0.5));

                END LOOP;

            END LOOP;

            test\_array <= arr;

            WAIT FOR interval;

        END LOOP;

        WAIT;

    END PROCESS;

    MONITOR :

    PROCESS (test\_array)

        VARIABLE oline : line;

        VARIABLE element : MATRIX\_ELEMENT;

        VARIABLE value\_proc, x\_proc, y\_proc : NATURAL;

    BEGIN

        FOR i IN 0 TO 9 LOOP

            write(oline, to\_string(test\_array(i, 0)) & to\_string(test\_array(i, 1)) & to\_string(test\_array(i, 2)) & to\_string(test\_array(i, 3)) & to\_string(test\_array(i, 4)) & to\_string(test\_array(i, 5)) & to\_string(test\_array(i, 6)) & to\_string(test\_array(i, 7)));

            writeline(OUTPUT, oline);

        END LOOP;

        element := findElementFunc(test\_array, 10, column);

        write(oline, "Max element: " & INTEGER'image(element(0)) & " at: (" & INTEGER'image(element(1)) & ", " & INTEGER'image(element(2)) & ")");

        writeline(OUTPUT, oline);

        writeline(OUTPUT, oline);

        el\_f <= element(0);

        y\_f <= element(1);

        findElementProc(test\_array, value\_proc, y\_proc, 10, column);

        el\_p <= value\_proc;

        y\_p <= y\_proc;

    END PROCESS;

END MainBehavior;

1. Временная диаграмма

В результате моделирования получим временную диаграмму, представленную на рис. 4.1.

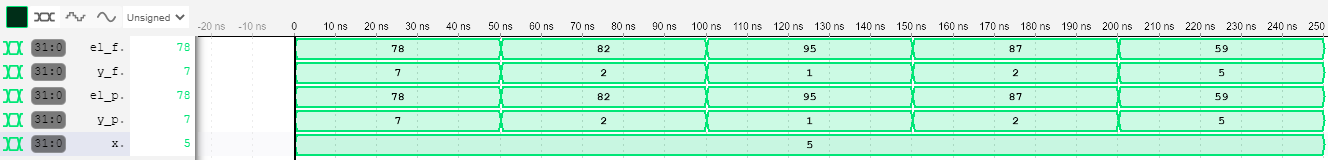


Рисунок 3.1 – Временная диаграмма

1. Вывод результата

|  |  |  |
| --- | --- | --- |
| 100 55 84 75 91 55 99 59  94 14 30 69 17 61 72 18  12 33 72 44 73 6 74 47  43 12 27 88 56 4 42 53  35 30 25 6 24 37 52 42  61 61 61 82 70 63 20 27  74 46 62 94 65 74 5 27  46 76 92 62 71 78 61 64  3 6 36 39 98 25 39 13  61 58 39 27 10 59 30 12  Max element: 78 at: (7, 5) | 55 97 89 72 18 37 68 49  72 55 63 20 92 46 47 97  75 70 65 40 57 82 58 70  15 85 10 63 89 20 60 84  54 81 56 9 33 45 72 0  100 4 46 74 60 11 54 98  59 60 47 34 47 54 13 53  71 76 97 9 59 25 6 31  30 15 97 51 42 69 97 45  31 72 76 92 68 0 97 32  Max element: 82 at: (2, 5) | 11 43 96 4 90 61 67 46  13 89 38 8 66 95 79 72  85 37 88 62 72 2 28 69  68 88 95 42 70 2 85 34  53 50 30 22 65 80 75 13  87 93 21 14 53 73 67 79  12 31 60 18 85 64 56 96  39 70 20 43 57 58 100 94  89 88 36 37 7 22 23 18  38 80 57 56 5 56 28 4  Max element: 95 at: (1, 5) |
| 23 74 94 88 4 81 25 25  63 95 34 32 28 14 89 76  57 17 9 76 50 87 46 73  29 72 34 17 21 23 11 68  28 57 43 79 17 32 73 16  25 71 64 23 46 84 7 85  32 54 35 70 42 78 69 84  40 100 20 36 90 7 15 36  28 28 40 55 99 1 88 64  35 14 92 49 54 33 62 64  Max element: 87 at: (2, 5) | 10 87 23 55 87 48 26 59  42 93 80 13 62 1 51 23  9 99 40 13 55 20 26 78  91 100 40 34 94 45 62 2  33 18 28 10 21 49 73 95  82 41 24 6 56 59 83 64  63 59 5 47 13 58 81 20  98 90 43 78 90 43 71 86  11 13 3 4 16 55 56 84  60 56 0 17 75 49 37 31  Max element: 59 at: (5, 5) |  |

1. вывод

При выполнении лабораторной работы я написал на языке VHDL функцию и процедуру поиска максимального элемента в столбце матрицы, а также выполнил их моделирование.