University of Twente

Master Embedded Systems

Design of Digital Systems (192130022)

Reverse a Vector

Author:
Mike Evers (s2165643)

Supervisor:
IR. E. Molekamp

February 27, 2019

UNIVERSITY
OF TWENTE.

Contents

1 Introduction

For the course Design of Digital Systems an extra assignment was given. No grade will be given but the first student whose complete the assignment will earn a lovely Snicker candy bar.

2 Assignment

The goal of the assignment is to design a synthesizable function that can reverse any vector. The function has to be implemented into a package so that it can be easily used. For the assignment two different functions have to be made which do exactly the same. One function is free of choice, the other has to be a recursive function.

3 Designs

3.1 For Loop Function

The first and easiest methode for creating a vector reversing function is a for loop that assigns all elements of the original vector to the new vector in a reversed order.

```
    function reverse_1 (inp_vector: in std_logic_vector) return std_logic_vector is
      variable result: std_logic_vector(inp_vector'REVERSE_RANGE);
    begin
      for i in inp_vector'RANGE loop
             result(i) := inp_vector(i);

      end loop;
return result;

    end function;
```

3.2 Recusive Function

The recursive function was a lot harder to design. I chose to make a function that switches the highest and lowest vector element and then, depending on the vector leghth, the function will remove the highest and lowest element and it will call itself again with the reduced vector. When every element is changed the vector will be restored and the reverse vector will be returned. This design is somewhat based on Figure 1.

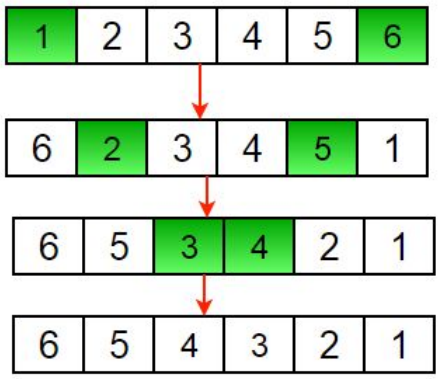

Figure 1: reverse-vector

4 Synthesis Result

The RTL-viewer shows that every vector is indeed reversed. In the three figures below (Figure 2, Figure 3, Figure 4) the screeshots of the RTL viewer are given. Because of the way RTL-viewer works I had to make a screenshot of every input/output separately to show that the vectors are indeed reversed.

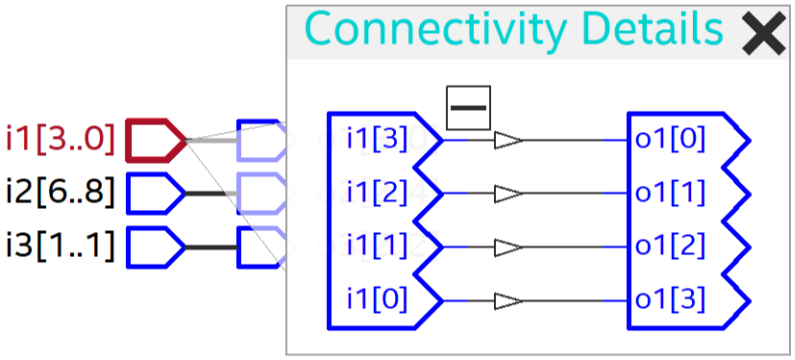

Figure 2: vector1

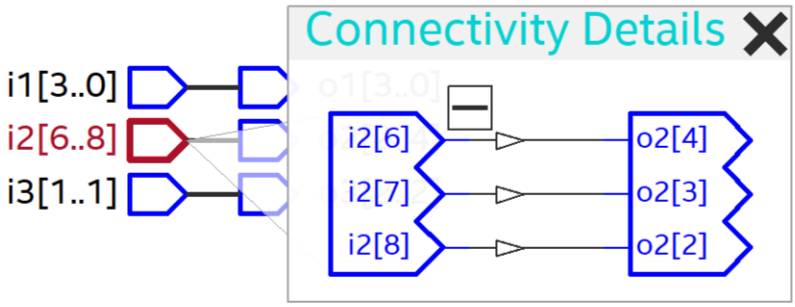

Figure 3: vector2

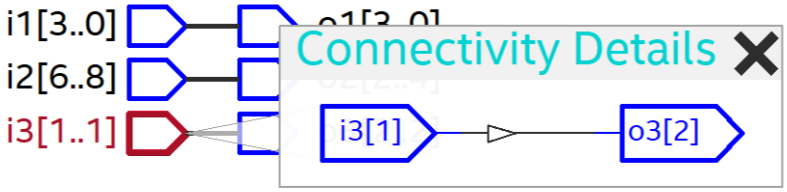

Figure 4: vector3

5 unsigned reversion

The last question of the assignment is:

Assume this function is also required for type unsigned (in package numeric_std), how would you write this function? FUNCTION reverse (a: unsigned) RETURN unsigned;

For this problem I would properly use a for loop again and go through the usigned bit by bit and reverse it.