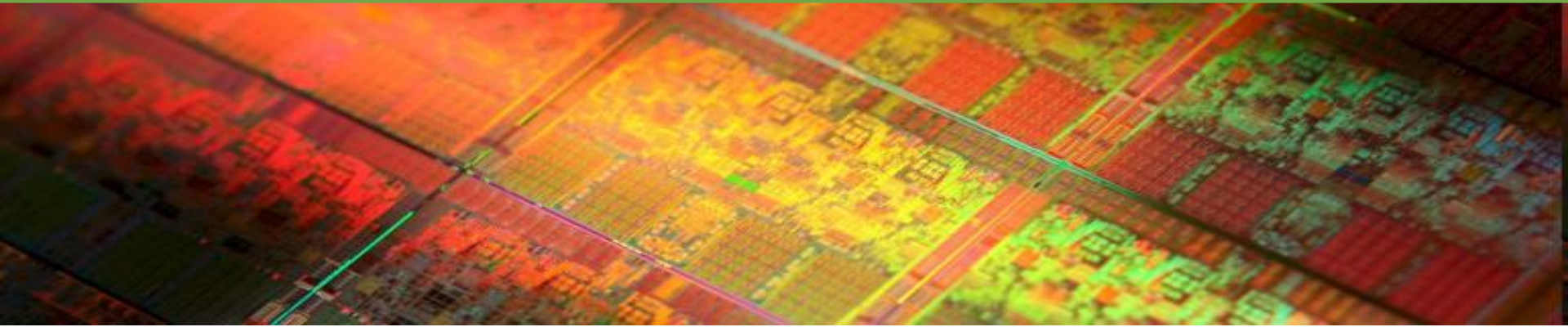


EE 2000

Logic Circuit Design (Fall 2018)



Lecture 4: VHDL Loops

Instructor: Dr. Ray C.C. Cheung

Outline

- Introduction
- VHDL Loops
- VHDL Functions
- Transport and Inertial Delays
- Summary

Introduction

- Often, we need to design systems with some activities in a repetitive manner.
- VHDL loop statements can be used to express this behavior.
- A loop statement is a sequential statement.
- Different kinds of loop statements
 - For Loops
 - While Loops

For Loops, For-Generate, and While Loops

- Allow group of statements to be executed either a number of times or until a condition is met
- For loop uses variables and performs calculations
 - For Loop is used mainly with variables where a calculation is involved.
- For-Generate uses signals and wires parts
- While is a For loop requiring manual setup

For Loop

- Question:
 - Using a For Loop structure, design a model that will output a single bit, P, that is the even parity value of a four-bit input vector A.

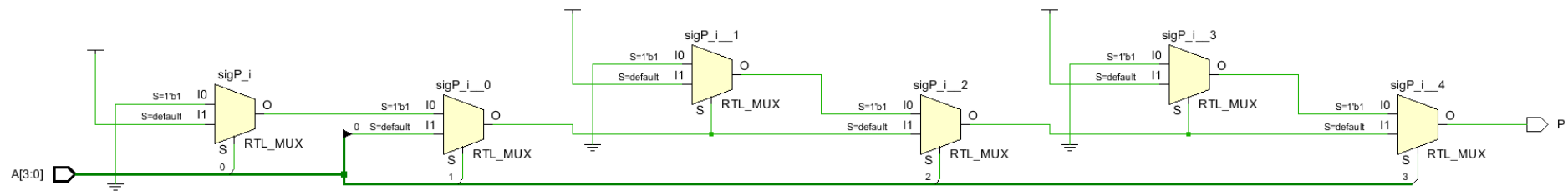
Answer

- Suggested Design using For Loop:

```
21 | library IEEE;
22 | use IEEE.STD_LOGIC_1164.ALL;
23 |
24 | entity ForParity is
25 |     port (
26 |         A : in std_logic_vector (3 downto 0);
27 |         P : out std_logic
28 |     );
29 | end ForParity;
30 |
31 | architecture Behavioral of ForParity is
32 |
33 | begin
34 |     process(A)
35 |         variable sigP : std_logic;
36 |         begin
37 |             sigP := '0';
38 |             for I in 0 to 3 loop
39 |                 if A(I) = '1' then
40 |                     if sigP = '1' then
41 |                         sigP := '0';
42 |                     else
43 |                         sigP := '1';
44 |                     end if;
45 |                 end if;
46 |                 P <= sigP;
47 |             end loop;
48 |         end process;
49 | end Behavioral;
```

Schematic

- Try to understand the design



Use of While Loop

- A while loop that models a down counter.
- Continue the decrementing process until the stop is encountered or the counter reaches 0.
- The counter is decremented on every rising edge of clk until either the count is 0 or stop is 1.

```
while stop = '0' and count /= 0 loop  
    wait until clk'event and clk = '1';  
    count <= count - 1 ;  
    wait for 0 ns;  
end loop;
```

Loops in VHDL

- Activity occurring in a repetitive way.
- Statements are sequential.
- Kinds of loop statements: **for**, **while**.
- Infinite loop:

- General form:

```
[loop-label:] loop  
    sequential statements  
end loop [loop-label];
```

- Can be terminated using exit statements:

```
exit; or exit when condition;
```

Loops in VHDL

- for loop:

- General form:

- ```
[loop-label:] for loop-index in range loop
 sequential statements
end loop [loop-label];
```

- Loop-index is incremented at the end of each loop.
  - Continues for every value in the range.

- while loop:

- General form:

- ```
[loop-label:] while condition loop  
    sequential statements  
end loop [loop-label];
```

- Condition is tested at the beginning of each loop.
 - Loop is terminated if condition is false.

For Generate

- For Loop has an optional label.
- For-Generate must have a label.
 - For I in 1 to 4 generate
 - For I in 4 downto 1 generate
- Question: We have input A & B, both 8-bit inputs, assign an AND gate to each of the input pair bits, to the 8-bit output vector Z.

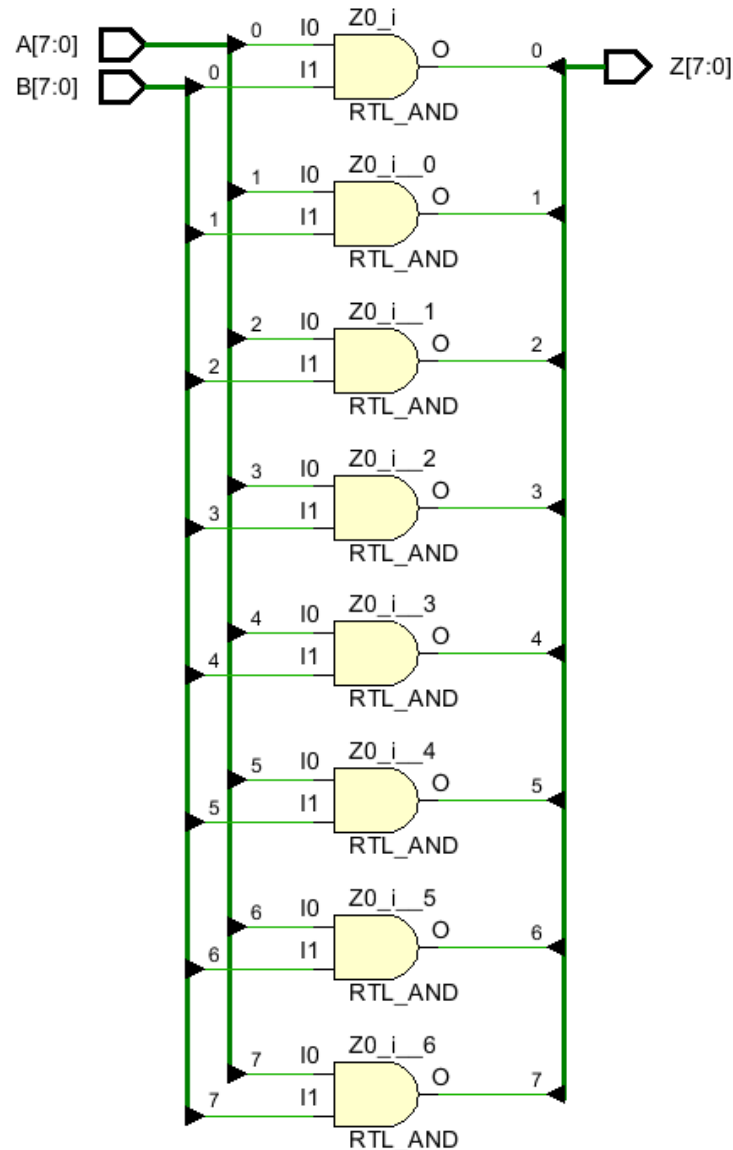
Example

- Remember the label please.

```
34 entity ForGenerateEx is
35     port(
36         A, B: in std_logic_vector (7 downto 0);
37         Z : out std_logic_vector (7 downto 0)
38     );
39 end ForGenerateEx;
40
41 architecture Behavioral of ForGenerateEx is
42
43     begin
44
45     Label_1:
46         for I in 0 to 7 generate
47             Z(i) <= A(i) and B(i);
48         end generate;
49
50 end Behavioral;
```

Schematic

- Can we generate to N-bits?



One more example

- Question: assign the AND gate to the odd elements of the vectors A & B to the output vector Y, assign the OR gate to the even elements of the vectors of A & B to the output vector Z.
 - A, B are 16-bit vectors
 - Y, Z are 8-bit vectors

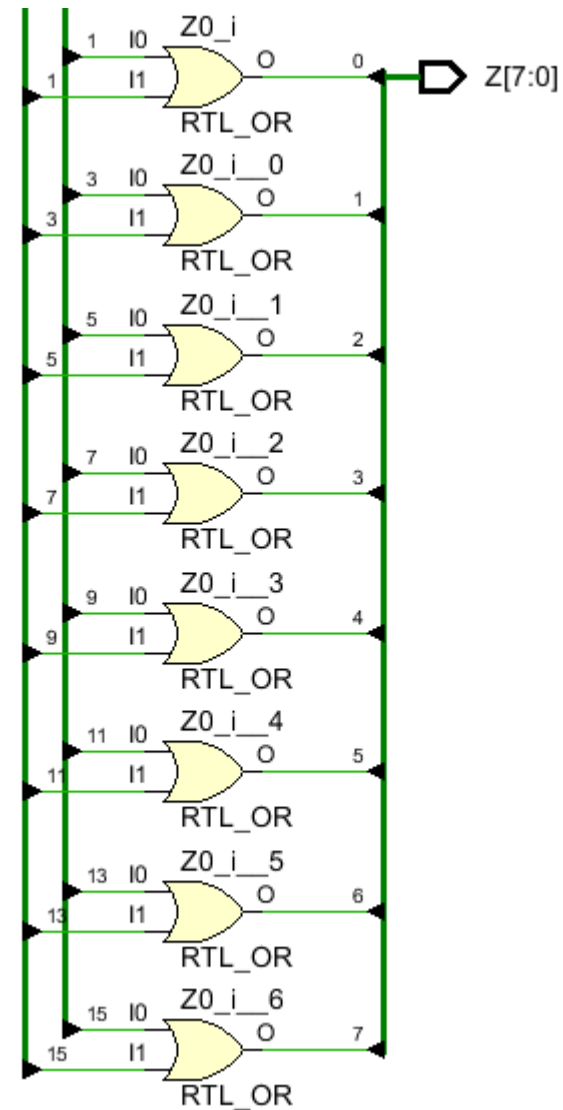
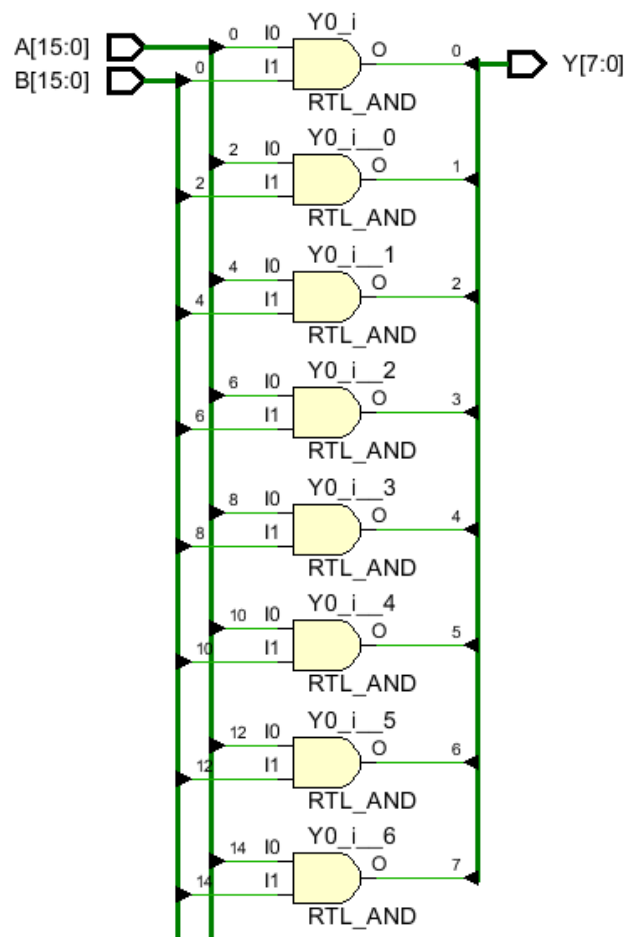
Design

- How to define odd and even in the bits?

```
34 entity ForGenerateEx is
35     port(
36         A, B: in std_logic_vector (15 downto 0);
37         Y, Z : out std_logic_vector (7 downto 0)
38     );
39 end ForGenerateEx;
40
41 architecture Behavioral of ForGenerateEx is
42
43     begin
44
45     --Label_1:
46     --    for I in 0 to 7 generate
47     --        Z(i) <= A(i) and B(i);
48     --    end generate;
49
50     Label_2:
51         for I in 0 to 7 generate
52             Y(i) <= A(i*2) and B(i*2);
53             Z(i) <= A(i*2 + 1) or B(i*2 + 1);
54         end generate;
55
56 end Behavioral;
```

Schematic

- Very long design



More Attributes: 'HIGH, 'LOW, and 'LENGTH

- When attached to a vector name, value of highest number bit, lowest number bit, or number of bits is returned
- 'LOW and 'HIGH are very useful in subroutines that can be called with different length vectors

Example

- If we have a port vector defined as:
 - A: in std_logic_vector (7 downto 0);
- A'high = 7
- A'low = 0
- A'length = 8

How to use it?

- See what is the difference?

```
34 entity ForGenerateEx is
35     port(
36         A, B: in std_logic_vector (15 downto 0);
37         Y, Z : out std_logic_vector (7 downto 0)
38     );
39 end ForGenerateEx;
40
41 architecture Behavioral of ForGenerateEx is
42
43 begin
44
45 --Label_1:
46 --     for I in 0 to 7 generate
47 --         Z(i) <= A(i) and B(i);
48 --     end generate;
49
50 Label_2:
51     for I in 0 to 7 generate
52         Y(i) <= A(i*2) and B(i*2);
53         Z(i) <= A(i*2 + 1) or B(i*2 + 1);
54     end generate;
55
56 end Behavioral;
```

Outline

- Introduction
- VHDL Loops
- VHDL Functions
- Transport and Inertial Delays
- Summary

Subroutines

- Can only contain sequential statements
- Can declare and use variables
- Reasons for using subroutines:
 - Perform same manipulation on several signals or at several places in VHDL code
 - Keep lengthy manipulation out of main code so it is easier to follow
 - Intend on using subroutine in several different VHDL designs

Subroutines

- Function
 - Can be called into use with inputs in its parameter list
 - Will return a single value
- Procedure
 - Can be called into use with inputs and variable type outputs in its parameter list
 - Can modify outputs
 - Will not return a value

Example Question

- Design a circuit with a four-bit input vector A, eight-bit input vector B, three single-bit output PA, PB, PC.
- PA is the odd parity of A
- PB is the odd parity of B
- PC is the even parity of bit[0] of A, and bit[1] of B

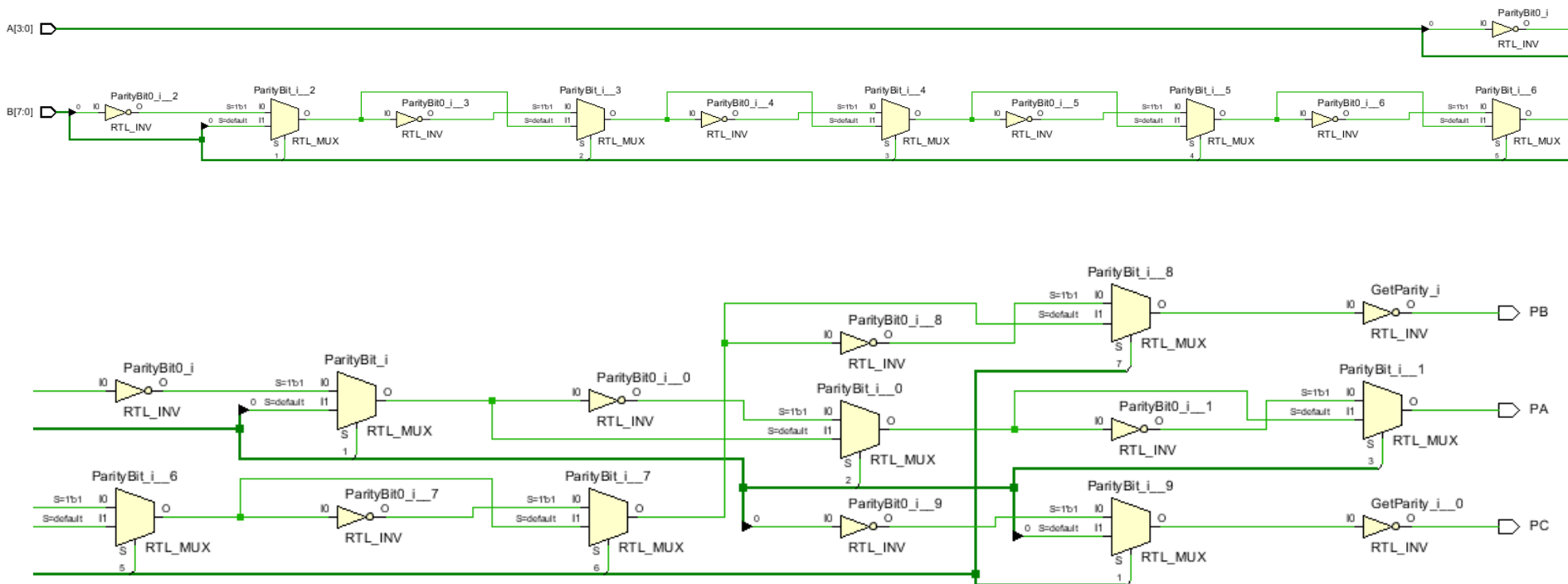
Design

- Let's go through the code together

```
25 entity FunctionEx is
26     port(
27         A: in std_logic_vector(3 downto 0);
28         B: in std_logic_vector(7 downto 0);
29         PA, PB, PC: out std_logic
30     );
31 end FunctionEx;
32
33 architecture Behavioral of FunctionEx is
34     function GetParity
35     (
36         Vect: in std_logic_vector;
37         odd: boolean
38     ) return std_logic is
39         variable ParityBit : std_logic := '0';
40     begin
41         for I in Vect'LOW to Vect'HIGH loop
42             if Vect(I) = '1' then
43                 ParityBit := not ParityBit;
44             end if;
45         end loop;
46
47         if odd then
48             ParityBit := not ParityBit;
49         end if;
50
51         return (ParityBit);
52     end GetParity;
53 begin
54
55     PA <= GetParity(A, FALSE);
56     PB <= GetParity(B, TRUE);
57     PC <= GetParity(A(0) & B(1), TRUE);
58
59 end Behavioral;
```

Schematic

- How do you read it?



Transport and Inertial Delays

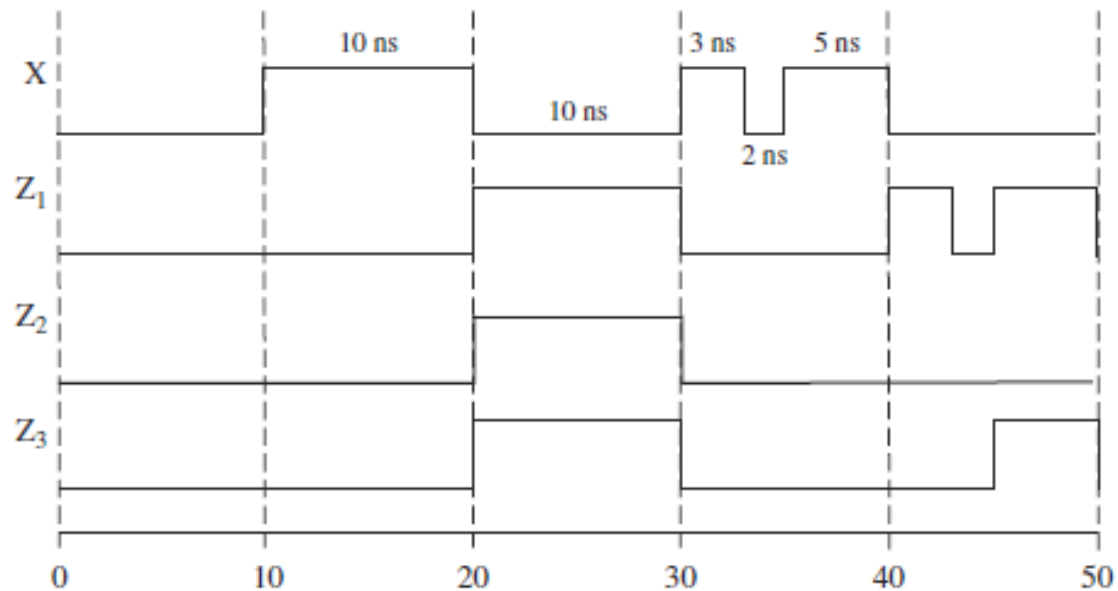
- **Inertial delay:** intended to model gates and other devices that do not propagate short pulses from the input to the output.
- **Transport delay:** intended to model the delay introduced by wiring; it simply delays an input signal by the specified delay time.

Transport and Inertial Delays

- Example of inertial and transport delays:

```
Z1 <= transport X after 10 ns; -- transport delay
Z2 <= X after 10 ns; -- inertial delay
Z3 <= reject 4 ns inertial X after 10 ns;
-- inertial delay with specified rejection pulse width
```

FIGURE 2-23: Transport and Inertial Delays



Summary

- We have covered the following VHDL concept in this module.
- Please continue to read and write more VHDL code, this is the only way to fully understand the important concept.
 - VHDL Loops
 - VHDL Functions
 - Transport and Inertial Delays

