**Laboration 1**

**Adders**

The laboratory should teach structural VHDL as well as provide training in simulation and seemed to RTL level and prototyping to hardware. At the same time, the laboratory should provide an overview of some different types of adder used in digital constructions.

**1. Preparation tasks**

Preparation tasks should be reported when the laboratory starts!

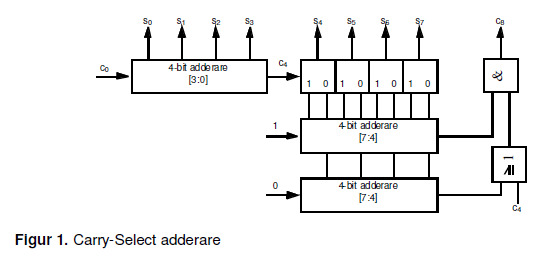
Study Chapter 5.10 in Wakerly: Digital Design - Principles & Practices and Understand the differences between different architectures for adder. When a designer chooses a ripple ladder and when is a carry-lookahead adder preferred?

1. Create a functional VHDL model for a whole adder. The whole adder should have one delay from inputs to 5 ns outputs.

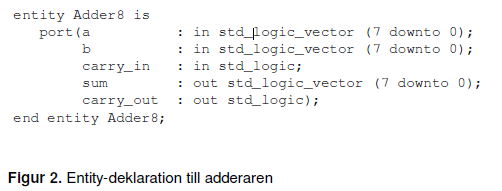
2. Create a structural VHDL model for a 4-bit ripple adder. Used the whole adder from the last assignment.

3. Create a functional VHDL model for a 4-bit carry-lookahead adder. The adder should not have a carry-generate or carry-propagate output.

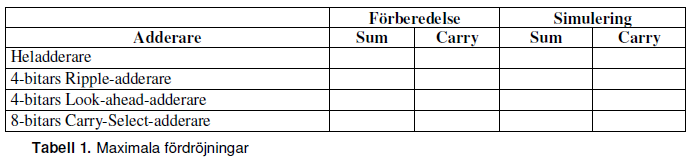
4. Figure 1 shows a carry-select adder. Explain the function! Create a structural VHDL model for an 8-bit carry-select adder built up of 4-bit ripple chargers. Muxarna should have 4 ns, AND gate 3 ns and OR gate 3 ns delay



Use the following entity declaration:



Calculate the maximum delay for the sum and carry outputs of the adder in Preparation 1 - 4. Fill in the values in Table 1. What additions occur the maximum delay?



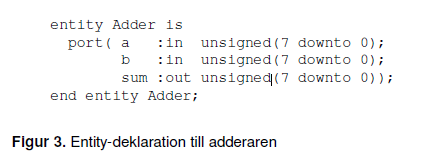
**2. Course of work**

1. Simulate the adder using the VHDL simulator and compare the delay with them Estimated values (Enter the result in Table 1).

2. Synthesise and download the Carry-look-ahead adder on an FPGA and report one functioning construction.

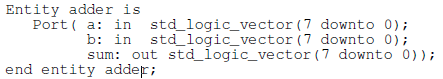
3. Synthesize the Ripple-carry adder. Compare performance (delay and number of logical cells) between the two different adder.

4. In the VHDL library, *ieee* there is a package *std\_logic\_arith*. In it there is Two types of vectors: *signed* and *unsigned*. For both types, they are the most common arithmetic and logical operations predefined. Use *std\_logic\_arith* and implement an RTL model for an 8-bit adder without carry with the following entity description.



Check the function by simulation.

4. Another way to make the addition is to use the package *std\_logic\_unsigned* or *std\_logic\_signed*. In these, they are the most common arithmetic and logical operations predefined for the type *std\_logic\_vector* and makes the number in the vector either interpreted as a *natural* or an *integer*. Used *std\_logic\_unsigned* and implement an RTL model for an 8-bit adder without carry with the following entity description.



**3. Accounting**

For an approved laboratory the following should be shown:

• Well-documented VHDL models for all adder.

• simulation result (where the maximum delays are shown).

• working hardware.