DAT093
Introduction to Electronic System Design

Introductory lab assignment

Sven Knutsson
svenk@chalmers.se
Dept. Of Computer Science and Engineering
Chalmers University of Technology
Gothenburg
Sweden

Goals

Update your knowledge of VHDL.

The assignments are quit simple and only intended to be the first steps into writing VHDL code.

If you have been writing some VHDL code before you will finish the assignments in just a few hours.

For those of you who are not that familiar with VHDL we hope that this will give you some basic knowledge.

Don´t be shy or afraid of asking the assistants a lot of questions. You are there to learn not to show skills from before.

Consult the assistants on suitable ways to write the code.

Assignments

The assignments will be both pure logic (asynchronous design) and clocked designs (syncronous design)

- We start with a simple one bit full adder
- We expand it to a four bit adder
- We generalize it to a configurable number of bits
- We design a continuous up counter

Testing your designs

You should test your designs using simulation in the simulation tool QuestaSim

You can find a Quick start tutorial on QuestaSim on the PingPong homepage. There is also a more extensive introduction to QuestaSim there but you can leave that for later.

At the homepage you will also find testbenches that you should use to test that your designs work as expected. The Quick start tutorial tell you how to use the testbenches

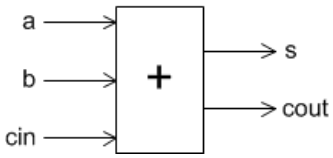
A testbench is a top level design where you use your design as a component

The testbench will assign values to the inputs of your design and check if the outputs are as expected

Full adder

To get to a multi bit adder we will start with a one bit full adder.

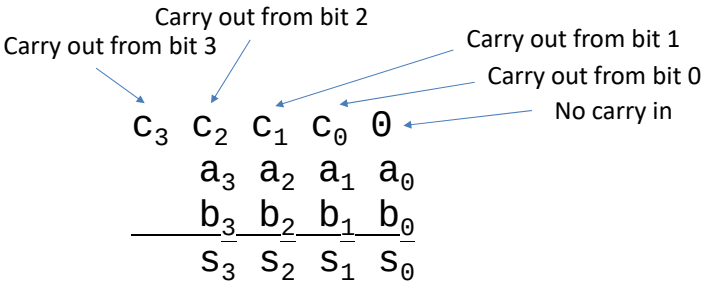
A one bit full adder adds two data bits and a carry in bit and the resultant output is a sum bit and a carry out bit



a	*b*	*cin*	*s*	*cout*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Four bit ripple adder

We extend the design to four bits with the following calculation



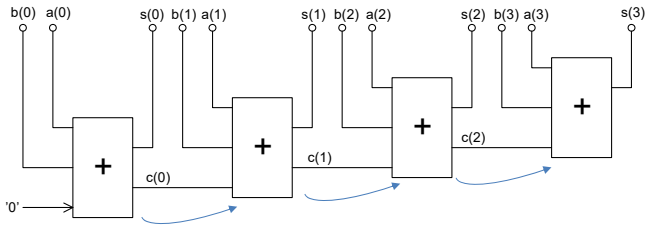
We can see that we have four one bit adders in parallel.

Notice that the calculation must go from LSB to MSB to get the correct result. The carry will ripple from bit to bit.

Notice also that the first one bit addition has zero (0) as carry in

Four bit ripple carry adder cont.

If we instantiate the design using four one bit adders as components we get



The carry bit ripples from bit to bit (LSB to MSB)

Once again notice that carry in for LSB is zero and that we don´t use the carry out from MSB

Generic ripple carry adder

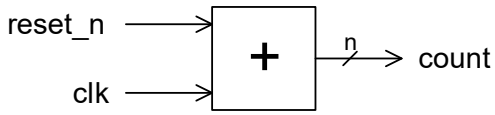
it would simplify things if we could design an adder component where we through a simple configuration could set the number of bits to any desired value.

We will change the ripple adder to a generic design where we can modify the number of bits used for the adder by setting a GENERIC parameter.

If you don´t know how to use GENERICS when you implement the adder try looking at page 78 in VHDL basics presentation in the Reading section of the course homapage.

Up counter

In the last assignment we design a continuous up counter that counts from zero to N-1 and then starts all over again.



What it counted is the number of clock pulses coming through the `clock` input of the design.

The counter should be set to zero by setting `reset_n` low.

Try to use your four bit ripple adder as a component in the counter design