Electronic Systems

Hands-on Activity

Ing. Luca Zulberti – luca.zulberti@phd.unipi.it
Prof. Massimiliano Donati – massimiliano.donati@unipi.it
Prof. Luca Fanucci – luca.fanucci@unipi.it

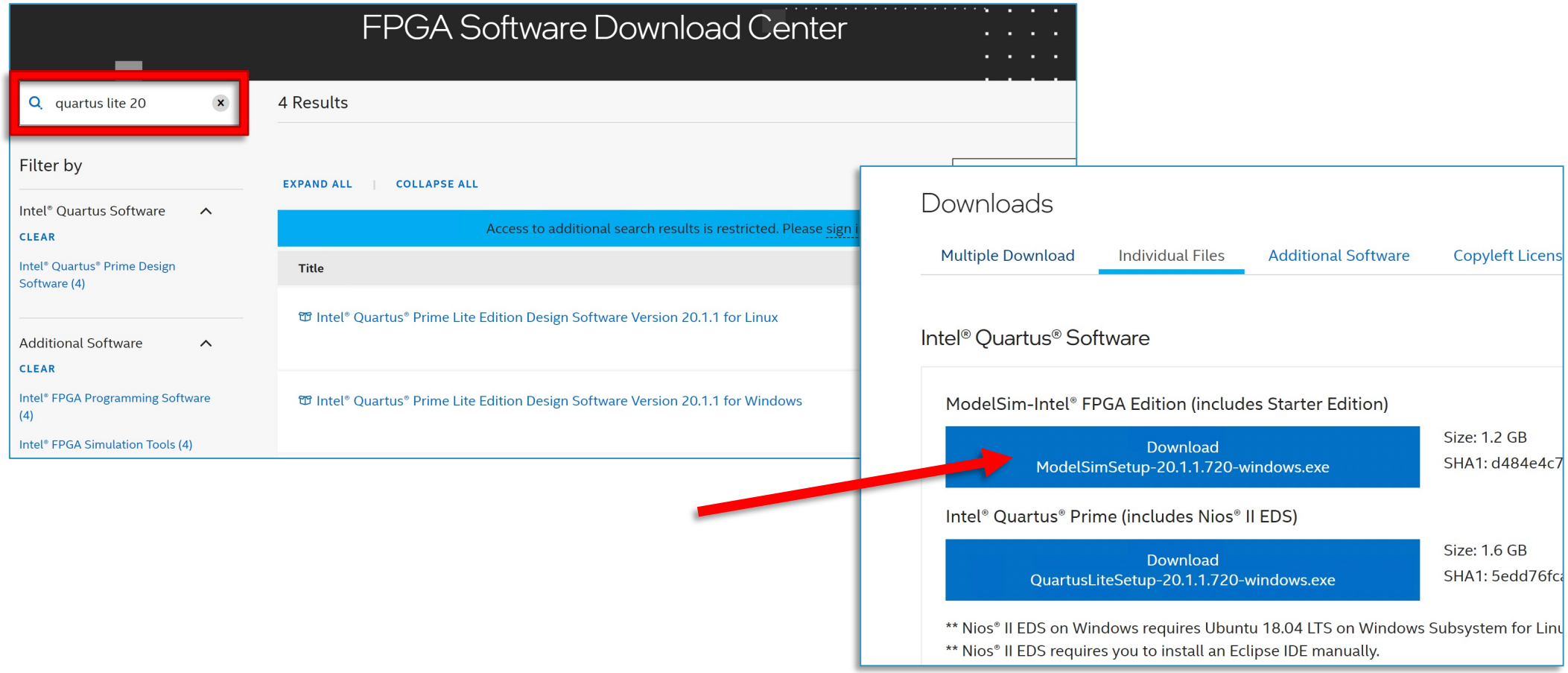
Agenda

1. Tool Installation
2. VHDL Practical Tips
3. VHDL Exercises

Agenda

1. Tool Installation
2. VHDL Practical Tips
3. VHDL Exercises

Modelsim



https://fpgasoftware.intel.com/

Vivado

Downloads

Log4j Vulnerability Patch | Licensing Help | Alveo

Vivado (HW Developer) | Vitis (SW Developer) | Vitis Embedded Platforms | Alveo Packages | PetaLinux

Version

We strongly recommend using the latest releases available.

2021.2

2021.1

2020

2020.2

Vivado Archive

CAE Vendor Libraries Archive

Vivado Design Suite - HLx Editions - 2020.2 Full Product Installation

Important Information

Download

Xilinx Unified Installer 2020.2: Windows Self Extracting Web Installer (EXE - 248.44 MB)

MD5 SUM Value : 102bb67c6806a6667dc7176be7997475

Download Verification

Digests | Signature | Public Key

Xilinx Unified Installer 2020.2: Linux Self Extracting Web Installer (BIN - 354.08 MB)

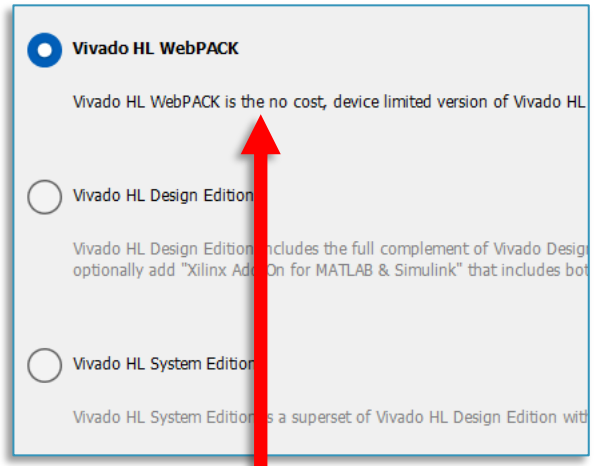
MD5 SUM Value : 0c74a74cbef649dceea34774c5bca490

Download Verification

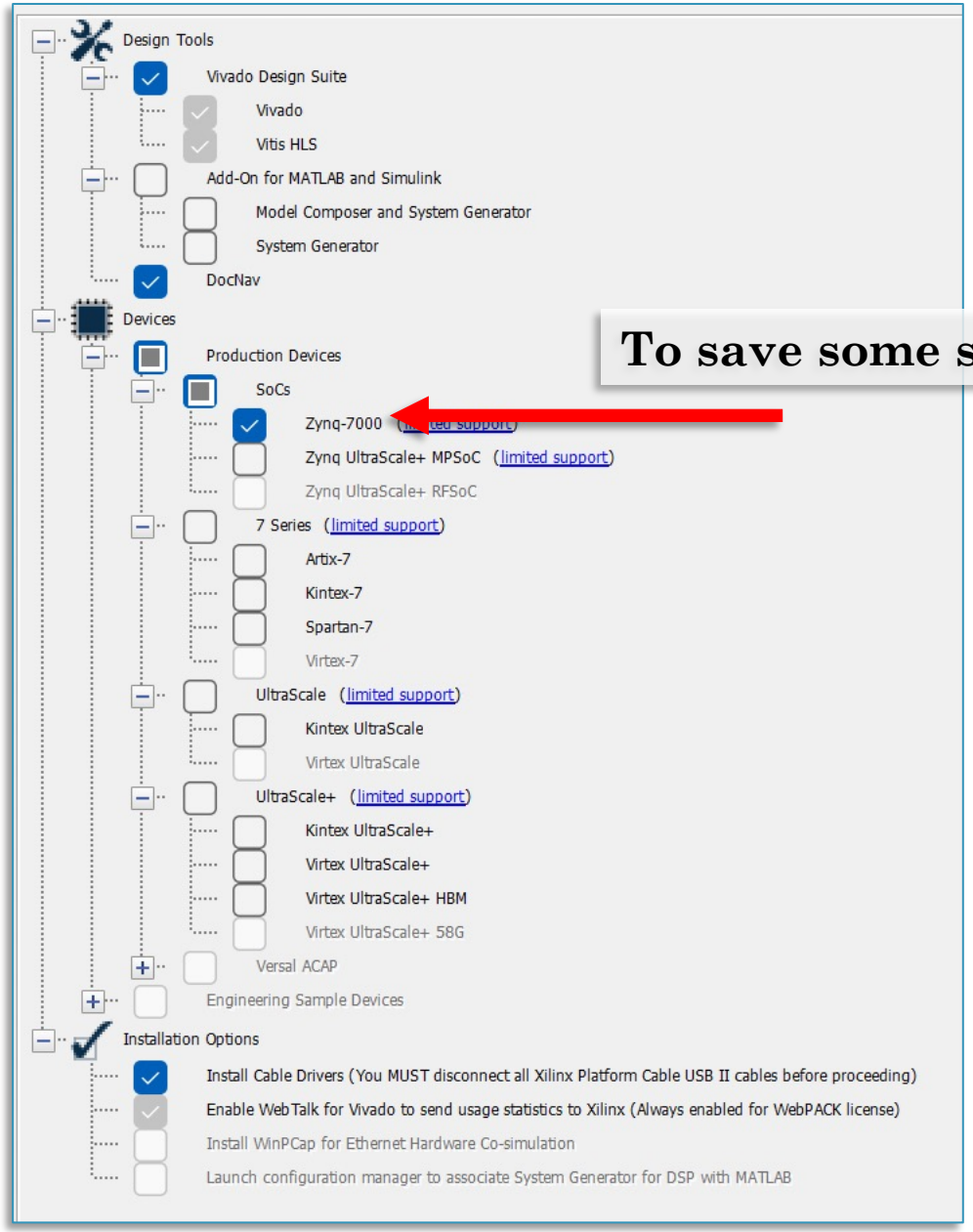
Digests | Signature | Public Key

https://www.xilinx.com/support/download.html

Vivado



Free Version



To save some space

Agenda

1. Tool Installation
2. VHDL Practical Tips
3. VHDL Exercises

VHDL Practical tips: Directory

- Directory structure
  - Project_name → Project directory, everywhere but avoid paths with spaces
    - src → Save here all your synthesizable .vhd source files. You can edit them on a generic text editor (Notepad++, Visual Studio Code, …).
    - tb → Save here all your simulation-only .vhd files (test benches). You may have only one of them or more than one, anyway it is important to keep them separated from the source files.
    - modelsim → Directory for Modelsim Project. The tool writes the project file and all the auxiliary files only inside this directory.
    - vivado → Same as Modelsim, but for the Vivado tool.

    - Each project with its own directory!

Name	Date modified	Type	Size
modelsim	08/04/2019 11:49	File folder	
src	04/04/2019 11:31	File folder	
tb	04/04/2019 11:31	File folder	
vivado	23/10/2017 17:04	File folder	

VHDL Practical tips: File Name

- Entity and source file must have the same name.
  - No spaces
  - Can be case sensitive

- The name of the file and the architecture are used to link different entities in the library. Depending on the tool it may not be able to link the entity compiled in your library to its instances it if the names do not match.

VHDL Practical tips: Port Map

- To recap, components port map can be done:

```
component fullAdder is
    port(-- Input of the full-adder
         a   : in std_logic;
         -- Input of the full-adder
         b   : in std_logic;
         -- Carry input
         c_i : in std_logic;
         -- Output of the full-adder
         o   : out std_logic;
         -- Carry output
         c_o : out std_logic
    );
end component;
```

```
i_full_adder : fullAdder
    port map(
            a    =>  a(i),
            b    =>  b(i),
            c_i  =>  c_int(i),
            o    =>  o(i),
            c_o  =>  c_int(i + 1)
        );
```

Explicit

```
i_full_adder : fullAdder
    port map(a(i), b(i), c_int(i), o(i), c_int(i + 1));
end generate;
```

Positional

- However, even if equivalent, **NEVER** use the positional. It can easily induce in error hard to debug.

VHDL Practical tips: Initialisation

- Initialisation of signals to be done only inside test benches

```
architecture bhv of counter_tb is -- Ts
    ----------------------------------
    constant T_CLK   : time := 10 ns;
    constant T_RESET : time := 25 ns;
    constant N_TB    : natural := 3;
    ----------------------------------
    signal clk_tb : std_logic := '0';
    signal rst_tb  : std_logic := '0';
```

- No initialisation on VHDL source files (src/*):
  - It will be ignored by the synthesizer
  - It may initialise the circuit in a “specific” state

- To “initialise” internal signal, always assign default value in combinatory process and reset value in sequential process. Always assert a Reset at the beginning of every Testbench.

```
ddf_n_proc: process(clk, rst_n)
begin
    if(rst_n = '0') then
        q <= (others => '0');
    elsif(rising_edge(clk)) then
```

```
ddf_n_proc: process(a, b)
    begin
        a <= '0'; --default
        b <= '1'; --default
```

→ To **avoid** inferred **latches**

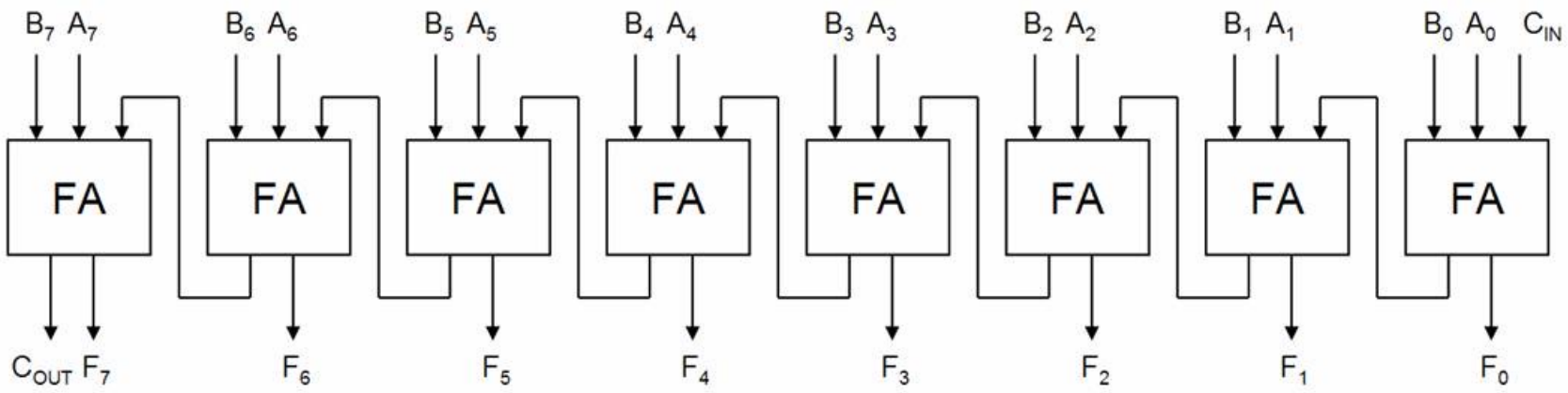
Agenda

1. Tool Installation
2. VHDL Practical Tips
3. VHDL Exercises

Ex 1 - Design and test of:

- RCA - Ripple Carry Adder with generic operands width designed in structural way

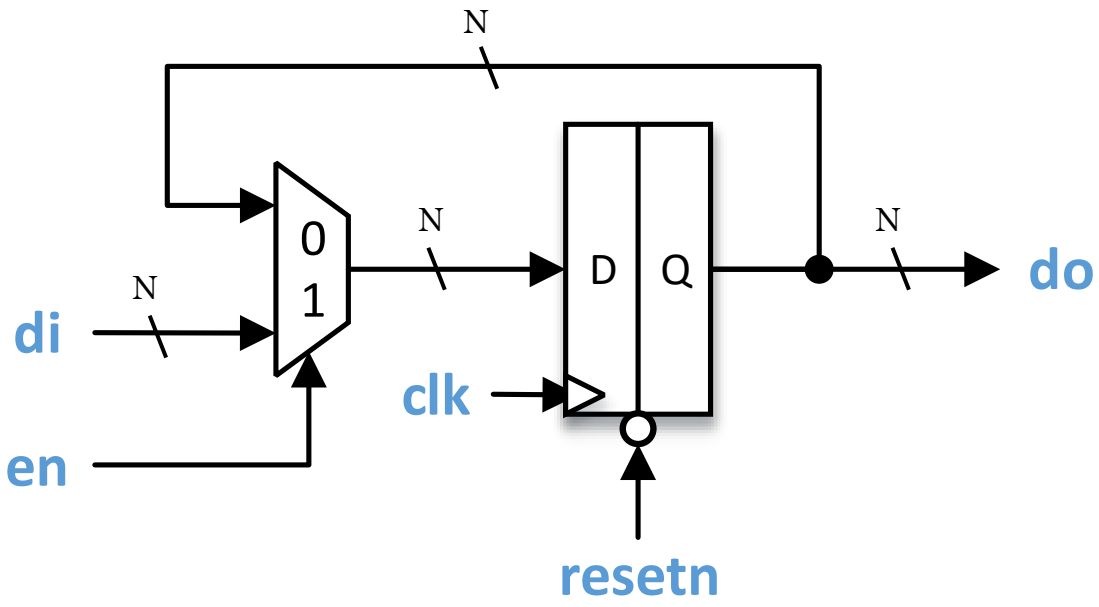
- HINTS:
- for … generate statement;
- FullAdder component
-



Ex 2 - Design and test of:

- DFF_N - Parallel D Flip-Flop with generic number of bits for input/output, asynchronous reset active low, with enable

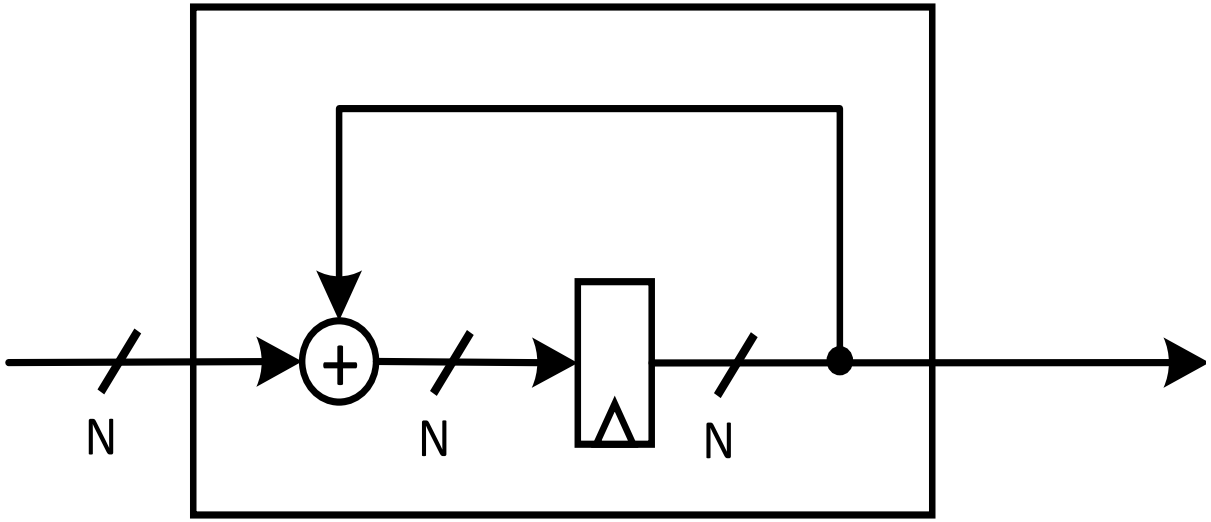
- HINTS:
- std_logic_vector type



Ex 3 - Design and test of:

- Counter

- HINTS:
- Ripple carry adder component:
- DFF_N component



Ex 4 - Design and test of:

- FIFO Buffer: First In First Out module with configurable depth (M) and data width (N), asynchronous reset active low.

- HINTS:
- for … generate statement;
- Two generics (M and N);
- Use DDF_N component;
-