

# ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

---

## Sprzętowa implementacja algorytmów

1. Wprowadzenie do obsługi ISE Webpack. Logika kombinacyjna.

---

Lider projektu: Politechnika Warszawska

Autor: Łukasz Mik

Akademia Nauk Stosowanych w Tarnowie

# Declaration

This laboratory instruction has been prepared in the context of the ENGINE project. Where other published and unpublished source materials have been used, these have been acknowledged.

## Copyright

© Copyright 2021 - 2023 the [ENGINE](#) Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

All rights reserved.



This document is licensed to the public under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

## Funding Disclaimer

This project has been funded with support from the European Commission. This report reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

# I. Instalacja oprogramowania ISE Webpack

Oprogramowanie ISE Webpack jest darmowym narzędziem firmy Xilinx, które umożliwia projektowanie systemów cyfrowych od syntezy i symulacji, przez implementację aż po konfigurację układu docelowego za pomocą interfejsu JTAG. Najnowsza wersja jest dostępna na stronie firmy Xilinx i jest dostosowana do systemów Windows oraz Linux.

**KROK 1:** Pobieranie plików instalacyjnych ze strony firmy Xilinx.

Wejź na stronę: <https://www.xilinx.com/products/design-tools/ise-design-suite/ise-webkit.html> i kliknij link o nazwie *Download ISE WebPACK software for Windows and Linux*. Jeśli posiadasz wersję systemu Windows 10 Pro lub Enterprise to możesz pobrać najnowszą wersję środowiska projektowego ISE Design Suite 14.7 ze zintegrowanym oprogramowaniem ISE Webpack, które trzeba uruchamiać z poziomu maszyny wirtualnej. Jest to wersja mniej wygodna dla użytkowników w związku z czym proponuję pozostać przy wersji ISE Design Suite 14.7 dla starszych systemów Windows 7/XP/Server.

Można jej używać w systemie Windows 10 po wprowadzeniu drobnych poprawek w katalogu instalacyjnym aplikacji. Procedura jest opisana na stronie producenta pod adresem [https://support.xilinx.com/s/article/62380?language=en\\_US](https://support.xilinx.com/s/article/62380?language=en_US) oraz omówiona w tej instrukcji.

Do pobrania pliku niezbędne jest posiadanie konta na stronie producenta. Jeśli masz już utworzone konto to możesz się zalogować, w przeciwnym razie musisz je utworzyć podając adres e-mail swojej poczty uczelnianej oraz zaznaczyć pełnioną funkcję na uczelni (w liście rozwijanej *Job Function* wybrać *Student*). Ze strony producenta należy pobrać plik TAR/GZIP o rozmiarze 6.18 GB do którego link jest pokazany na poniższym rysunku.

ISE Design Suite - 14.7 Full Product Installation

Download Type	Full Product Installation
Last Updated	Oct 23, 2013

**⚠ Last Updated October 2013**

As of October 2013, ISE has moved into the sustaining phase of its product life cycle, and there are no more planned ISE releases.

ISE supports the following devices families and their previous generations: Spartan-6, Virtex-6, and Coolrunner. For more information, please visit the [ISE Design Suite](#).

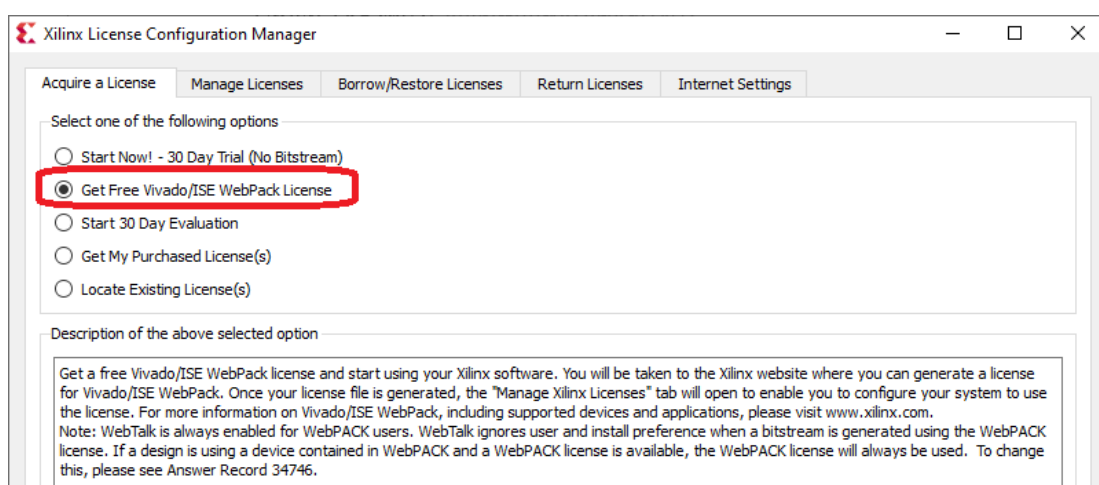
Xilinx recommends [Vivado Design Suite](#) for new design starts with Virtex-7, Kintex-7, Artix-7, and Zynq-7000.

<a href="#">Full DVD Single File Download Image (TAR/GZIP - 7.78 GB)</a>
MD5 SUM Value : bfe4e9c3cd8d2d7024163ca140113d25
<a href="#">Full Installer for Linux (TAR/GZIP - 6.09 GB)</a>
MD5 SUM Value : e8065b2ffb411bb74ae32efa475f9817
<a href="#">Full Installer for Windows 7/XP/Server (TAR/GZIP - 6.18 GB)</a>
MD5 SUM Value : 94f40553a93dfbeca642503e2721b270

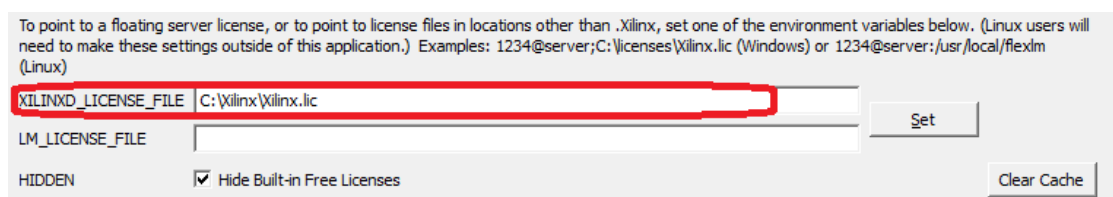
## KROK 2: Instalacja oprogramowania ISE Webpack 14.7.

Po rozpakowaniu pobranego archiwum należy uruchomić plik o nazwie *xsetup.exe* w trybie zgodności z systemem Windows 7. Podczas instalacji z listy produktów do instalacji należy wybrać ISE Webpack. W opcjach instalacji można też odznaczyć *Install WinCap for Ethernet Hardware Co-simulation* oraz *Install Cable Drivers* jeśli nie będziemy korzystać z programatora JTAG firmy Xilinx. Do realizacji zajęć będzie używany zestaw Numato Elbert V2, do którego producent udostępnia zarówno sterowniki jak i program do konfiguracji układu FPGA.

Na końcu procesu instalacji zostanie uruchomiony program menadżera licencji, za pomocą którego należy uzyskać plik licencji po zalogowaniu się na swoje konto Xilinx.



Plik licencyjny o nazwie *Xilinx.lic* należy pobrać ze swojego konta i wgrać do folderu, w którym zostało zainstalowane oprogramowanie. W menadżerze licencji można też ręcznie ustawić ścieżkę do pliku licencyjnego.



W przypadku problemów z uzyskaniem licencji należy poinformować o tym fakcie prowadzącego zajęcia. Po zakończonej instalacji należy zastosować poprawki opisane w nocie Xilinx AR# 62380:

- 1) Należy przejść do folderu `<install_path>\Xilinx\14.7\ISE_DS\ISE\lib\nt64\`
- 2) Zmieniamy nazwę pliku *libPortability.dll* na *libPortability.dll.orig*
- 3) Zmieniamy nazwę pliku *libPortabilityNOSH.dll* na *libPortability.dll*

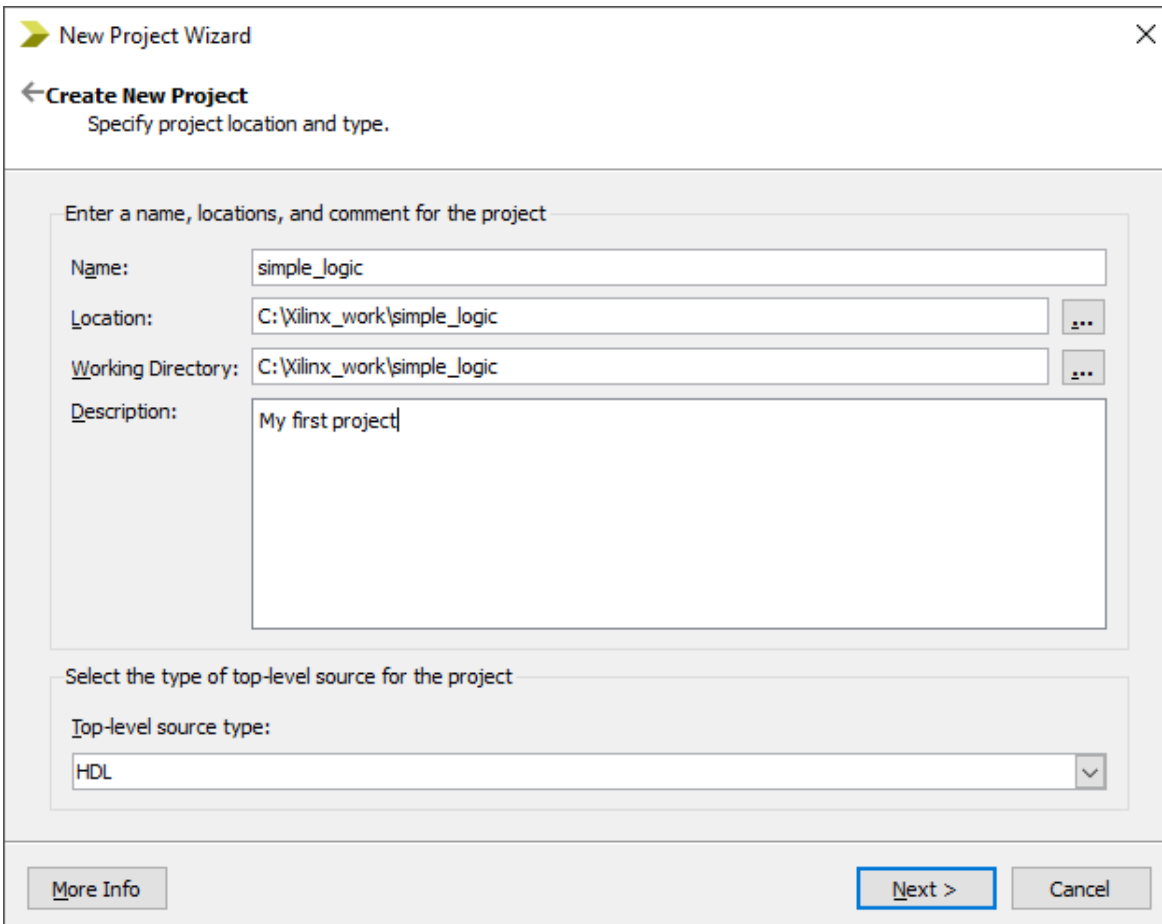
Kroki od 1 do 3 powtarzamy dla tych samych plików w folderze: `<install_path>\Xilinx\14.7\ISE_DS\common\lib\nt64\`

## II. Tworzenie nowego projektu w programie ISE WebPack

Uruchom program ISE Design Suite 14.7, korzystając ze skrótu na pulpicie lub z menu Start. Po uruchomieniu pojawi się główne okno programu. Nowy project można utworzyć za pomocą przycisku New Project... w oknie startowym lub wybierając opcję File → New Project ...

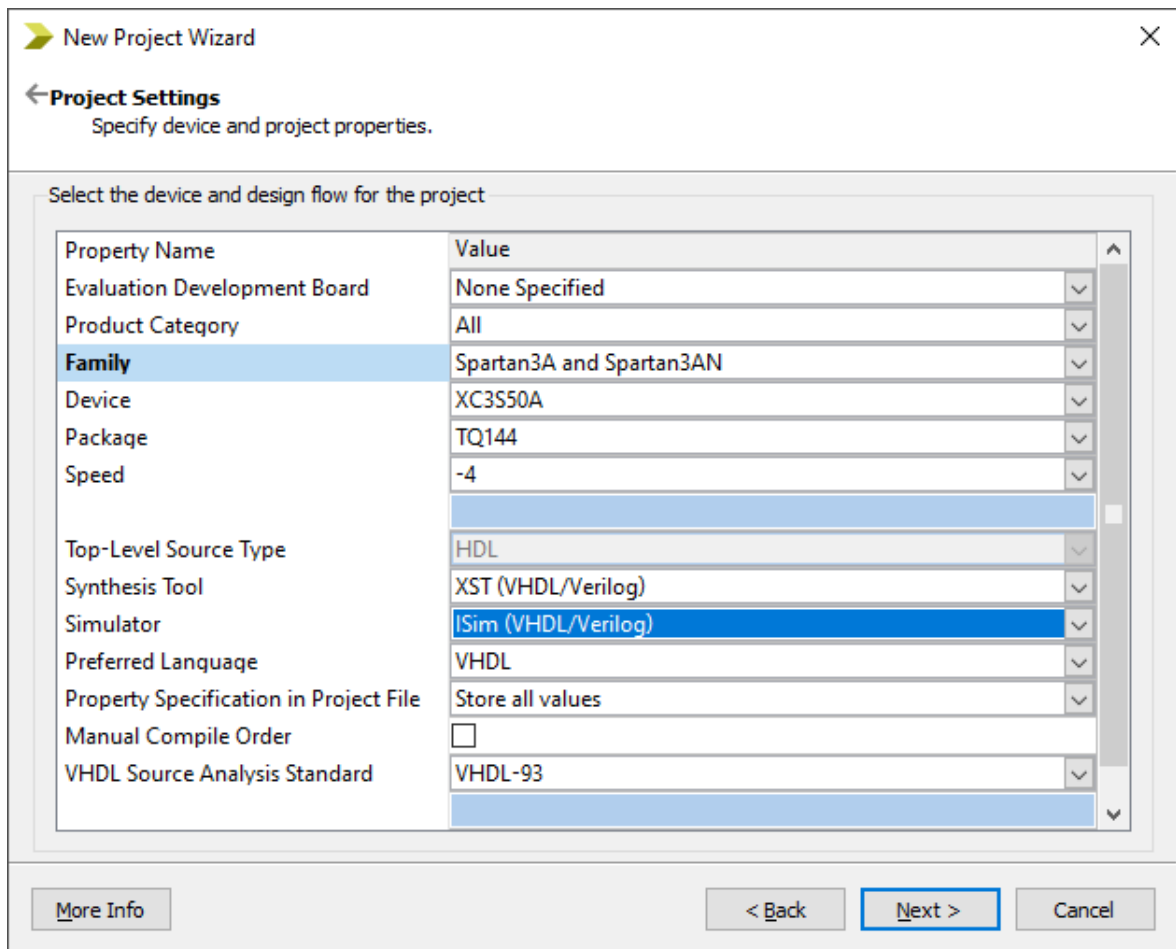
Przed utworzeniem projektu należy utworzyć folder roboczy np. *Xilinx\_work*, w którym będą zapisywane nasze projekty.

W oknie tworzenia projektu należy wpisać nazwę projektu np. *simple\_logic* i wybrać utworzony wcześniej folder roboczy, w którym zostanie on zapisany.

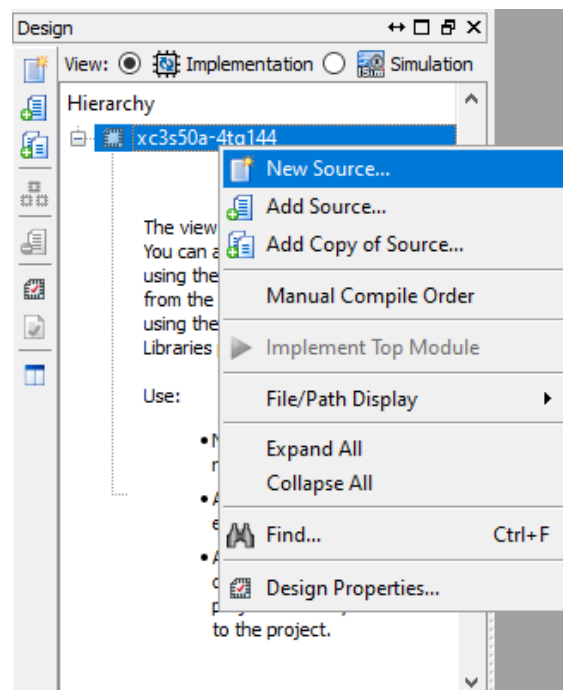


Musimy ustawić też typ źródła znajdującego się najwyżej w hierarchi projektu → *HDL*.

W kolejnym oknie ustawień projektu podajemy parametry układu docelowego oraz typ języka opisu sprzętu. Wybieramy parametry na podstawie oznaczeń układu FPGA, znajdującego się na płytce Numato Elbert V2. Zostały one przedstawione na kolejnym obrazku. Jako preferowany język wybieramy VHDL (wersja VHDL-93).



Po utworzeniu projektu dodajemy do niego odpowiednie źródła.



W pierwszej kolejności dodajemy plik typu *VHDL Module* o nazwie *simple\_logic*.

W kolejnym oknie, które się pojawi ustawiamy 2 porty wejściowe o nazwach *a* i *b* oraz jeden wyjściowy o nazwie *y*.

Port Name	Direction	Bus	MSB	LSB
a	in	<input type="checkbox"/>		
b	in	<input type="checkbox"/>		
y	out	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

Po dodaniu pliku do projektu otworzy się okno edycyjne. Należy zwrócić uwagę, że plikowi źródłowemu zostało przypisane rozszerzenie *\*.vhd*.

Uzupelnimy teraz opis architektury, tak aby projekt realizował prostą operację iloczynu logicznego 2 bitów.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity simple_logic is
    Port ( a : in  STD_LOGIC;
          b : in  STD_LOGIC;
          y : out STD_LOGIC);
end simple_logic;

architecture Behavioral of simple_logic is

begin

y <= a and b;      -- funkcja logiczna AND

end Behavioral;

```

Trzeba pamiętać o tym, że przypisanie sygnałów odbywa się za pomocą operatora “<=”

W kolejnym kroku należy utworzyć plik źródłowy typu *Implementation Constraints File*, który będzie zawierał przypisanie portów z jednostki projektowej do odpowiednich pinów układu FPGA. Jako nazwę pliku również wpisujemy *simple\_logic*. Plikowi zostanie automatycznie przypisane rozszerzenie *\*.ucf*.

Dla zestawu Numato Elbert V2 wszystkie piny są opisane w pliku *elbertv2.ucf*, dostępnym pod adresem:

<https://productdata.numato.com/assets/downloads/fpga/elbertv2/elbertv2.ucf>

Z tego pliku do naszego pliku UCF kopiujemy tylko 3 linijki:

```
NET "DPSwitch[0]" LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;  
NET "DPSwitch[1]" LOC = P69 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;  
NET "LED[0]"      LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
```

Modyfikujemy nazwy sygnałów na następujące:

```
NET "a"  LOC = P70 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;  
NET "b"  LOC = P69 | PULLUP | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;  
NET "y"  LOC = P46 | IOSTANDARD = LVCMOS33 | SLEW = SLOW | DRIVE = 12;
```

Powyzsza konfiguracja łączy porty a, b oraz y z odpowiednimi pinami P70, P69 i P46. Piny P70, i P69 mają włączone wewnętrzne podciąganie do zasilania (PULLUP), ustawiony standard napięciowy jako LVCMOS33 (parameter IOSTANDARD), maksymalny prąd na poziomie 12mA (parameter DRIVE) oraz wolno narastające i opadające zbocza sygnału (parameter SLEW).

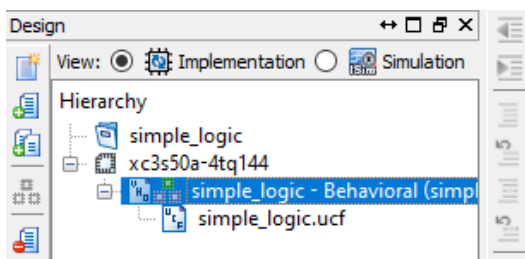
Pin P46 ma ustawiony standard napięciowy LVCMOS33, wolne zbocza sygnału oraz maksymalny prąd 12mA.

Po zapisaniu wszystkich plików w projekcie przechodzimy do kolejnego etapu, czyli syntezy logicznej.

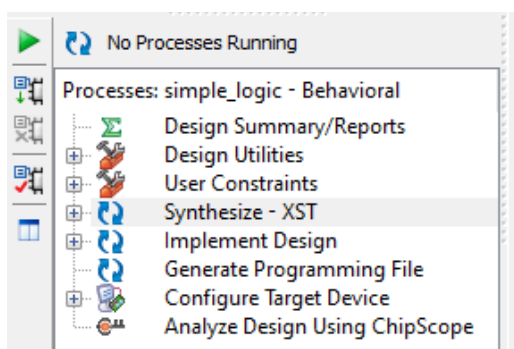


### III. Synteza logiczna.

Po zaznaczeniu pliku źródłowego *simple\_logic.vhd* w oknie widoku projektu:

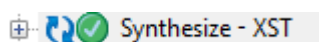


w dolnym oknie procesowym pojawią się opcje, które odpowiadają za kolejne etapy kompilacji projektu.



W pierwszej kolejności uruchamiamy opcję *Synthesize – XST*, która spowoduje uruchomienie sprawdzenia składni oraz procesu syntezy logicznej.

Jeśli składnia kodu VHDL nie zawiera błędów a synteza logiczna przebiegła pomyślnie to wówczas w oknie procesowym pojawi się znak prawidłowego ukończenia tego etapu:



nastomiast w oknie konsoli zostanie wygenerowany raport o pozytywnym zakończeniu procesu syntezy.

```
=====
*                               Final Report                               *
=====

Clock Information:
-----
No clock signals found in this design

Asynchronous Control Signals Information:
-----
No asynchronous control signals found in this design

Timing Summary:
-----
Speed Grade: -4


    Minimum period: No path found
    Minimum input arrival time before clock: No path found
    Maximum output required time after clock: No path found
    Maximum combinational path delay: 7.000ns

=====

Process "Synthesize - XST" completed successfully
```

## IV. Implementacja projektu w układzie docelowym. Konfiguracja układu przy użyciu pliku \*.bit.

Kolejnym etapem kompilacji projektu jest uruchomienie procesu implementacji w układzie docelowym. Służy do tego opcja *Implement Design* w oknie procesowym. Po prawidłowym przebiegu implementacji pojawi się znak prawidłowego ukończenia tego etapu:

 *Implement Design* a w oknie konsoli raport o pozytywnym zakończeniu procesu implementacji.

```
Process "Place & Route" completed successfully

Started : "Generate Post-Place & Route Static Timing".
Running trce...
Command Line: trce -intstyle ise -v 3 -s 4 -n 3 -fastpaths -xml simple_logic.twx
Loading device for application Rf_Device from file '3s50a.nph' in environment
C:\Xilinx\14.7\ISE_DS\ISE\
  "simple_logic" is an NCD, version 3.2, device xc3s50a, package tq144, speed
-4

Analysis completed Fri Jun 17 10:55:24 2022
-----

Generating Report ...

Number of warnings: 0
Total time: 2 secs

Process "Generate Post-Place & Route Static Timing" completed successfully
```

Ostatnim etapem przed zaprogramowaniem układu docelowego jest wygenerowanie pliku konfiguracyjnego przez uruchomienie opcji *Generate Programming File*. Jeśli plik zostanie wygenerowany prawidłowo to w oknie konsoli pojawi się raport jak na poniższym obrazku.

```
Started : "Generate Programming File".
Running bitgen...
Command Line: bitgen -intstyle ise -f simple_logic.ut simple_logic.ncd

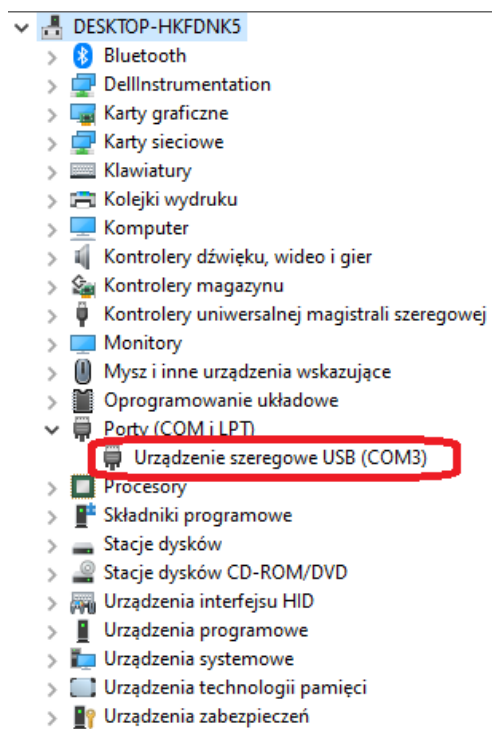
Process "Generate Programming File" completed successfully
```

Przed zaprogramowaniem układu docelowego należy pobrać sterowniki ze strony firmy Numato Lab: <https://numato.com/wp-content/uploads/2021/06/numatocdcdriver.zip> Sterowniki należy rozpakować do dowolnego folderu i wskazać je w przypadku, gdy system sam nie znajdzie odpowiednich sterowników.

Do programowania układu docelowego (konfiguracji układu FPGA) trzeba użyć programu dostarczonego przez producenta:

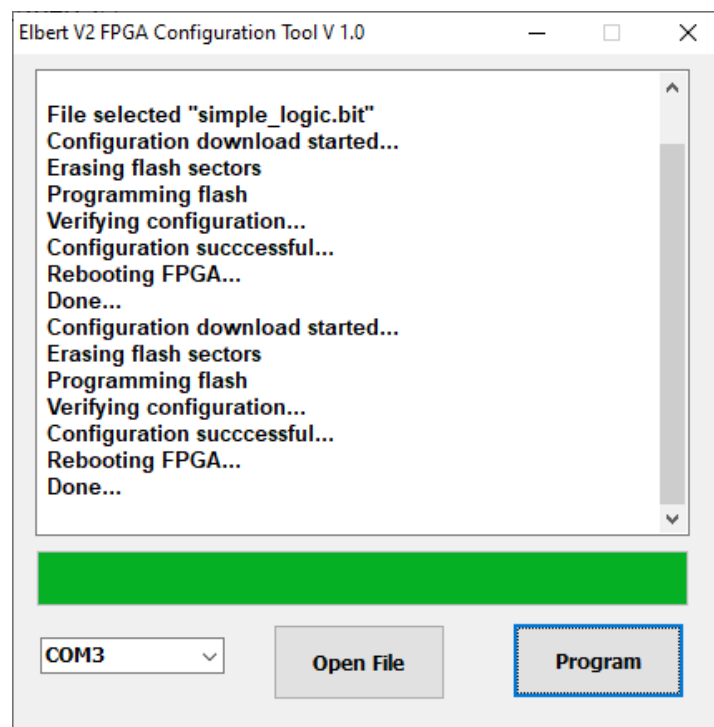
<https://productdata.numato.com/assets/downloads/fpga/elbertv2/ElbertV2Config.exe>

Po uruchomieniu programu należy wskazać numer portu COM w systemie, przypisany do zestawu Elbert V2.

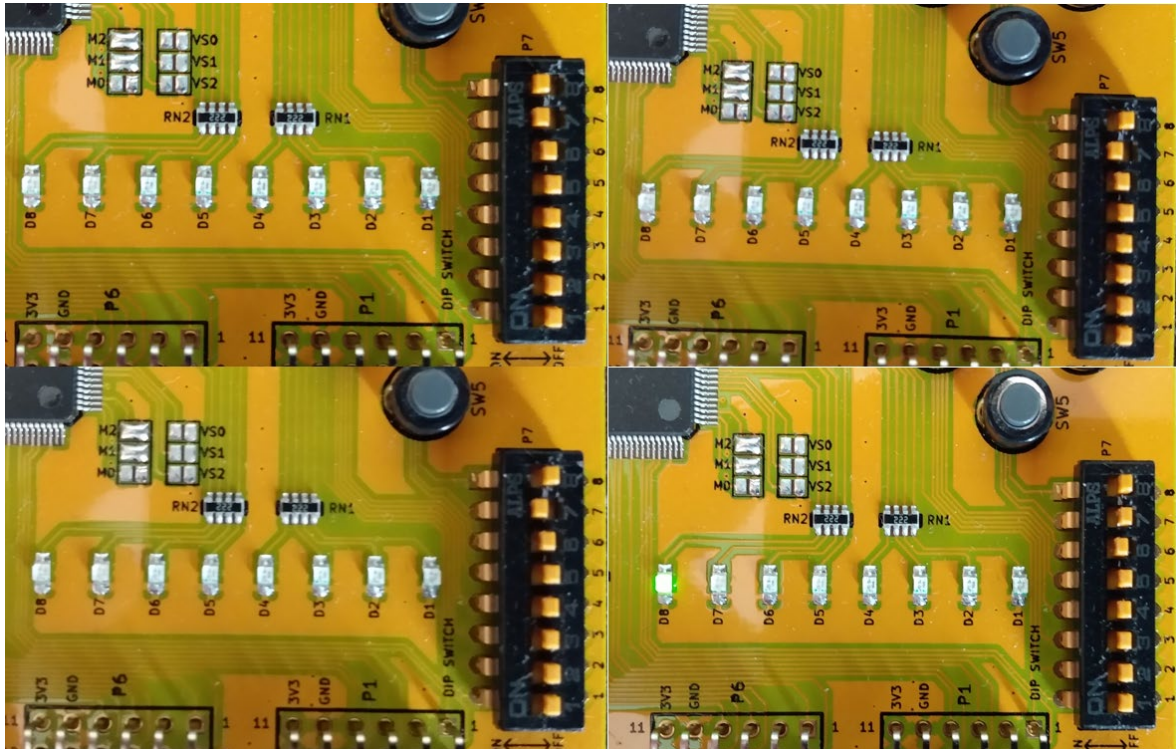


Następnie otwieramy plik o rozszerzeniu *\*.bit*, który znajduje się w folderze projektu i programujemy układ docelowy.

Po prawidłowym przebiegu konfiguracji układu FPGA zaświeci się dioda D9 na płycie ewaluacyjnej (przypisana do pinu DONE układu FPGA). W oknie programu pojawi się komunikat *Done...*

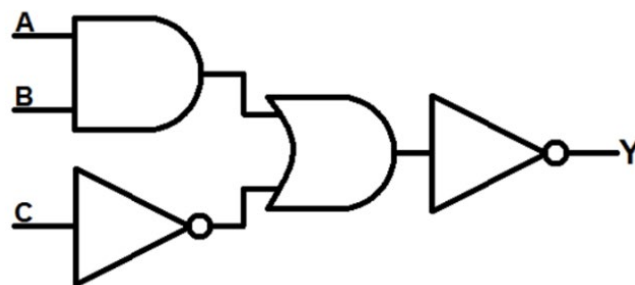


Efekt działania utworzonego w ramach projektu przedstawiony został na kolejnych zdjęciach. Zmiana stanów na wejściach *a* i *b* odbywa się za pomocą przełączników suwakowych o numerach 7 i 8. Stan wyjścia *y* jest sygnalizowany za pomocą diody *D8*.



**ZADANIE:**

Zrealizuj projekt w języku VHDL i przypisz do niego odpowiednie elementy na płytce Elbert V2. Schemat układu kombinacyjnego do realizacji został przedstawiony na poniższym rysunku.



## References

- *User manual for Elbert V2 - Spartan 3A FPGA Development Board.*  
<https://numato.com/docs/elbert-v2-spartan-3a-fpga-development-board/>
- *62380 - ISE Install - Installing and Running ISE 10.1 or 14.7 on a Windows 8.1 or Windows 10 machine.*  
[https://support.xilinx.com/s/article/62380?language=en\\_US](https://support.xilinx.com/s/article/62380?language=en_US)