



PODSTAWY UKŁADÓW PROGRAMOWALNYCH FPGA WPROWADZENIE DO OBSŁUGI PROGRAMU ISE WEBPACK.

Dariusz Tefelski
dariusz.tefelski@pw.edu.pl

Politechnika Warszawska,
Wydział Fizyki

REGULAMIN PRZEDMIOTU

- 1 Przedmiot *Podstawy Układów Programowalnych FPGA* składa się z 10-ciu zajęć laboratoryjnych po 3h.
- 2 Zajęcia są obowiązkowe, przy czym dozwolone są 2 nieobecności usprawiedliwione i 1 nieobecność nieusprawiedliwiona.
- 3 Ocena wystawiana jest proporcjonalnie do zdobytych punktów trakcie zajęć przy czym nie mniejsza niż 3.0, w przypadku osób uczestniczących aktywnie we wszystkich zajęciach (dozwolone nieobecności w pkt. 2).
- 4 Aktywne uczestnictwo = wykonywanie zadań zgodnie z instrukcją niezależnie od rezultatu finałowego.

PROGRAMOWALNE UKŁADY LOGICZNE (PLD)

- PLD wyglądem przypominają typowy układ scalony (czarna obudowa z wyprowadzonymi pinami). Istotą tych układów jest jednak ich budowa wewnętrzna.
- Układy PLD składają się z ogromnej ilości pojedynczych komórek, zwanych elementami logicznymi, zawierającymi bramami AND, OR, NOT oraz przerzutnik RS.
- Programowanie układów PLD sprowadza się do ustawienia połączeń między odpowiednimi bramkami/przerzutnikami w komórkach.

PORÓWNANIE MIKROKONTROLERÓW I UKŁADÓW PLD

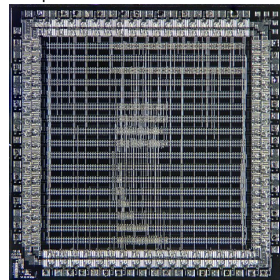
	Mikrokontrolery	PLD
Możliwość wykonywania zadań równoległych	nie	tak
Kod wykonywany linia po linii	tak	nie
Cena	mniejsza	większa

NAJPOPULARNIEJSZE RODZAJE PLD

- **FPGA** (*Field-Programmable Gate Array*) - zawiera w sobie matrycę programowalnych bloków logicznych i konfigurowalnych połączeń między nimi.
- **CPLD** (*Complex Programmable Logic Devices*) - posiadają uproszczoną strukturę wewnętrzną i mniejszą liczbę komórek w przeciwieństwie do FPGA. W zamian posiadają większą liczbę pinów wejściowych. Układy te są dedykowane do zadań typowych dla układów kombinacyjnych (stan wyjść zależy wyłącznie od stanu wejść).
- **ASIC** (*Application Specific Integrated Circuit*) - to układy scalone, w których nie ma możliwości ich rekonfiguracji. Natomiast wykonują funkcje szybciej i z mniejszym użyciem zasobów w porównaniu do mikrokontrolerów. Układy ASIC zawierają w sobie funkcjonalność nie tylko mikrokontrolerów ale i zewnętrznych układów, które normalnie trzeba by dołączyć do mikrokontrolera.



FPGA z rodziny Spartan. Źródło: Wikipedia



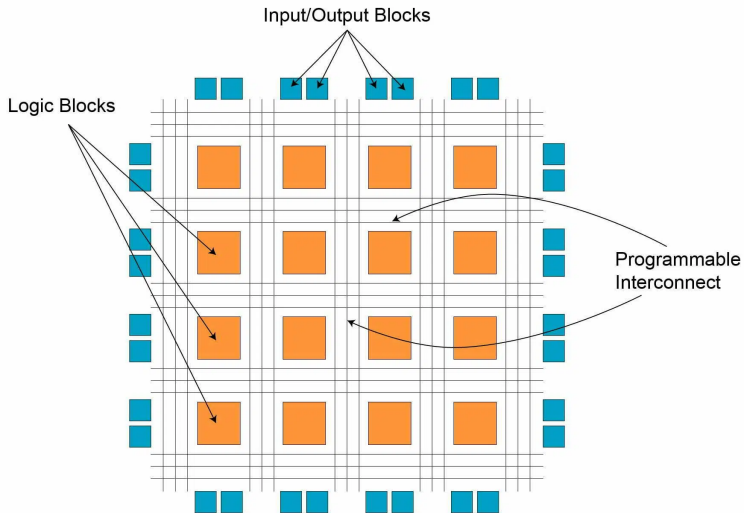
Wnętrze układu ASIC. Źródło: Wikipedia

HISTORIA UKŁADÓW FPGA

- 1985 - firma Xilinx wyprodukowała pierwszy komercyjny FPGA - XC2064. Firmę założyli: Ross Freeman (fizyk) i Bernard Vonderschmitt.
- XC2064 miał programowalne bramki i programowalne połączenia między bramkami.
- XC2064 posiadał zaledwie 64 konfigurowalnych bloków logicznych (CLB), z dwoma 3-wejściowymi tablicami typu look-up table (LUT).
- W latach 80-tych, firma Xilinx była niekwestionowanym liderem na rynku i szybko rośnie od 1985 do połowy 1990, kiedy konkurencja dopiero powstaje. Xilinx ma znaczny udział w rynku.
- Lata 90-te to okres dużej popularności układów FPGA.
- Na początku lat 1990, układy FPGA były wykorzystywane głównie w telekomunikacji i zastosowań sieciowych. Pod koniec dekady, FPGA zaczynają być stosowane w urządzeniach konsumenckich, samochodowych i w przemyśle.

Na podstawie: <http://mysinski.wieik.pk.edu.pl>

ARCHITEKTURA UKŁADÓW FPGA



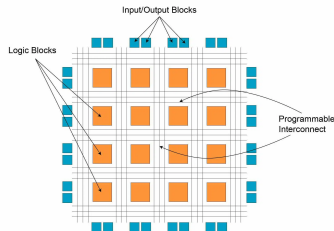
Źródło: <https://sii.pl/>

ELEMENTY UKŁADÓW FPGA

Podstawowe elementy:

- Konfigurowalne bloki logiczne (*Configurable Logic Blocks*) - grupy przerzutników, bramek logicznych i multiplexerów potrzebnych do realizacji podstawowych układów kombinacyjnych i sekwencyjnych.
- Sieć połączeń - fizyczne ścieżki pomiędzy wszystkimi CLB, których połączenie jest konfigurowane za pomocą kluczy.
- Klucze programowalne - są to elementy pozwalające na sterowanie połączeniami pomiędzy różnymi CLB w celu budowania większych struktur z pojedynczych bramek i przerzutników.
- Sieć doprowadzenia zegarów - doprowadza do każdego CLB zegar w sposób zapewniający jak najmniejsze różnice w czasach propagacji.
- Bloki wejść/wyjść - umożliwiają komunikację z urządzeniami zewnętrznymi.

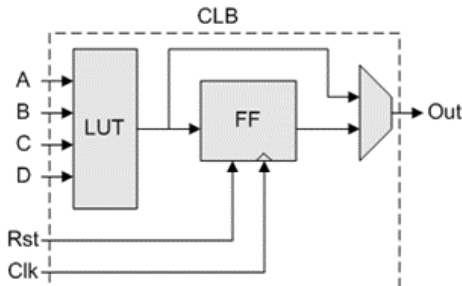
Źródło: <https://sii.pl/>



KONFIGUROWALNE BLOKI LOGICZNE

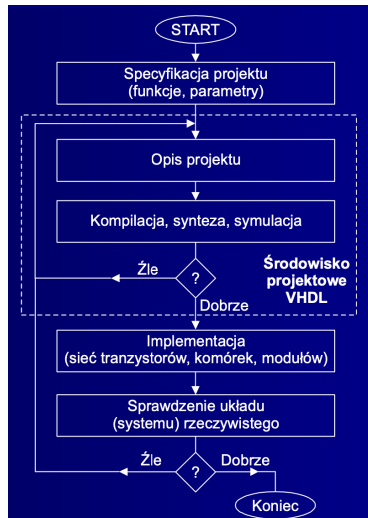
Składają się z:

- części kombinacyjnej (lewa strona rysunku) złożonej z bramek logicznych (LUT - *Look-Up Table*).
- przerzutnik posiadający, poza wejściem i wyjściem, połączenia do sieci zegarowej i sygnału resetu (środkowa część rysunku).
- multiplexer (prawa strona rysunku), który wybiera sygnał wyjściowy w zależności od tego czy wykorzystano tylko część kombinacyjną CLB czy również sekwencyjną.



PROGRAMOWANIE UKŁADÓW FPGA

- Aby zdefiniować zachowanie układu FPGA używa się języka opisu sprzętu takiego jak VHDL czy Verilog.
- Następnie przy pomocy narzędzi syntezy generuje się listę połączeń, która potem w procesie implementacji jest odwzorowywana w konkretnym układzie.
- Należy zwrócić uwagę, że proces syntezy dopuszcza tworzenie układów logicznych dowolnych rozmiarów, podczas gdy proces implementacji jest próbą wpisania go do konkretnego układu FPGA, gdzie może zabraknąć zasobów do realizacji zadanej logiki.
- Do zaprogramowania układu FPGA służy plik binarny, który zawiera informacje o konfiguracji układu.



Źródło: <http://wmii.uwm.edu.pl/>

Na podstawie: <http://mysinski.wieik.pk.edu.pl>

- Język VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) jest językiem opisu sprzętowego używany w komputerowym projektowaniu układów cyfrowych typu FPGA i ASIC.
- Różni się zasadniczo od języków programowania takich jak C, C++ czy Python tym, że komendy nie są wykonywane linia po linii.
- Program w VHDL opisuje kolejność i połączenia pojedynczych elementów logicznych.
- Język powstał w latach 80. w ramach projektu finansowanego przez Departament Obrony USA. Miał być używany do dokumentowania zachowania układów ASIC.
- W 1987 roku stał się standardem IEEE.
- W języku VHDL nie rozróżniamy wielkości liter!

STRUKTURA PROGRAMU W JĘZYKU VHDL

- Na początku programu należy dołączyć bibliotekę:

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

- Następnie określa się pojedynczy układ (*entity*), w którym definiowane są wyprowadzenia układu FPGA. W poniższym przykładzie mamy 2 wejścia *a* i *b* oraz jedno wyjście *y*, a jednostka została nazwana *simple_logic*:

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

- Tryby portów:
 - **in** - sygnał wejściowy,
 - **out** - sygnał wyjściowy,
 - **inout** - sygnał wejściowy i wyjściowy,
 - **buffer** - port wyjściowy z możliwością odczytu.

STRUKTURA PROGRAMU W JĘZYKU VHDL

- W kolejnym kroku określa się architekturę:

```
architecture nazwa_architektury1 of nazwa_jednostki is
    begin
        -- instrukcje przypisania wartości do sygnałów,
        -- procesy,
        -- komponenty
end Behavioral;
```

np:

```
architecture Behavioral of simple_logic is
    begin
        y <= a and b; -- funkcja logiczna AND
end Behavioral;
```

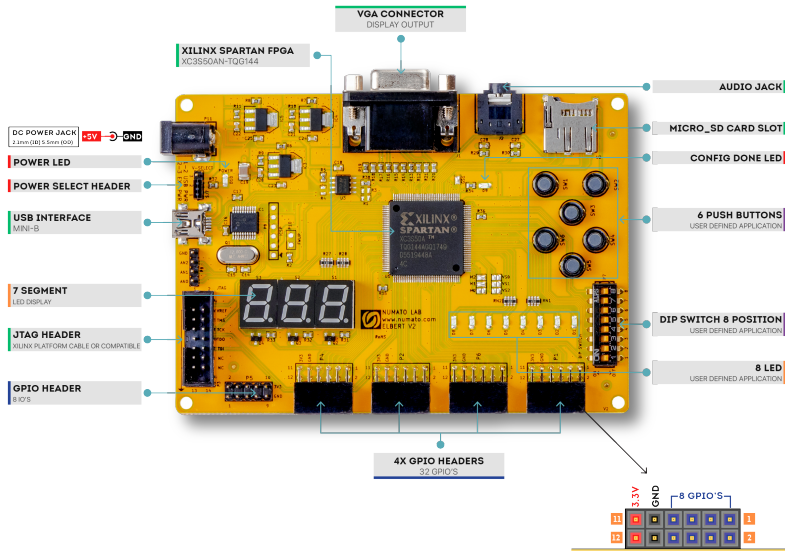
- Podwójny znak minus oznacza komentarz.

NUMATO ELBERT V2 - SPECYFIKACJA TECHNICZNA

- Sercem zestawu jest układ FPGA firmy Xilinx z serii Spartan XC3S50A, który posiada 144 wyprowadzenia, z czego 104 są do dyspozycji użytkownika.
- Układ pracuje z maksymalną częstotliwością dochodzącą do 280 MHz.
- W jego wnętrzu znaleźć można 1728 elementów logicznych.
- Konfiguracja FPGA i komunikacja zestawu odbywa się poprzez kabel USB 2.0. Od strony zestawu znajduje się złącze Mini USB.
- Zestaw posiada 16 MB pamięci Flash która przechowuje konfiguracje układu FPGA.
- Na płytce umieszczono dodatkowo: 8 kontrolki LED, 6 przycisków typu micro-switch, 8 przełączników DIP, wyjście VGA (stosowane jako wyjście grafiki dla monitorów komputerowych), wyjście Audio Stereo, 3 wyświetlacze 7-segmentowe, gniazdo dla kart MicroSD.

Źródło: <https://forbot.pl>

NUMATO ELBERT V2 - PŁYTKA



Źródło: <https://numato.com>

Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR																																																																																																
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A + B$	$\overline{A + B}$	$A \oplus B$	$\overline{A \oplus B}$																																																																																																
Symbol																																																																																																							
Truth Table	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table><tr><th>B</th><th>A</th><th>X</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	1
A	X																																																																																																						
0	1																																																																																																						
1	0																																																																																																						
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	1																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	0																																																																																																					
0	1	1																																																																																																					
1	0	1																																																																																																					
1	1	0																																																																																																					
B	A	X																																																																																																					
0	0	1																																																																																																					
0	1	0																																																																																																					
1	0	0																																																																																																					
1	1	1																																																																																																					

Źródło: <https://majsterkowo.pl>

Dziękuję za uwagę!

Prezentacja powstała w oparciu o materiały z:

- "Układy FPGA w przykładach" Majewski, Zbysiński, BTC.
- <https://numato.com>
- <http://mysinski.wieik.pk.edu.pl>
- <http://wmii.uwm.edu.pl/>
- <http://www.cs.put.poznan.pl/>
- <https://forbot.pl/>
- <https://sii.pl/>