

Literatura (w zakresie języka VHDL)

Książki:

Ashenden P.: Designers Guide to VHDL, MKP, 2002.

Ashenden P.: The VHDL Cookbook (internet)

Skahill K.: Język VHDL, WNT, Warszawa 2001.

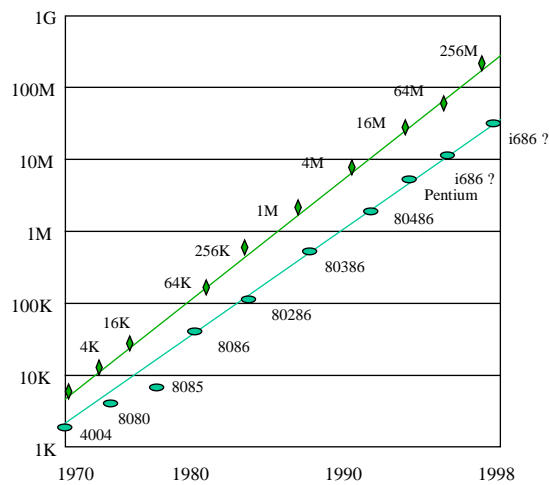
Wrona W.: VHDL język opisu i projektowania układów cyfrowych.

Internet:

???

ale tu warto zacząć: <http://www.vhdl.org> (comp.lang.vhdl -> FAQ)

Wzrost złożoności układów scalonych



Zasoby w projektowaniu UC:

1970 – INTEL 4004	4 projektantów	1 tys. tranzystorów
1982 – INTEL 80286	20 projektantów	100 tys. tranzystorów
1992 – INTEL PENTIUM	100 projektantów	3 mln tranzystorów
2002 – INTEL PENTIUM 4	1000 projektantów	150 mln tranzystorów
200? – ???		

Producenci narzędzi do projektowania

Aldec

Cadence

Compass

Mentor Graphics

Synopsys

Synplicity

Viewlogic

Model Technology

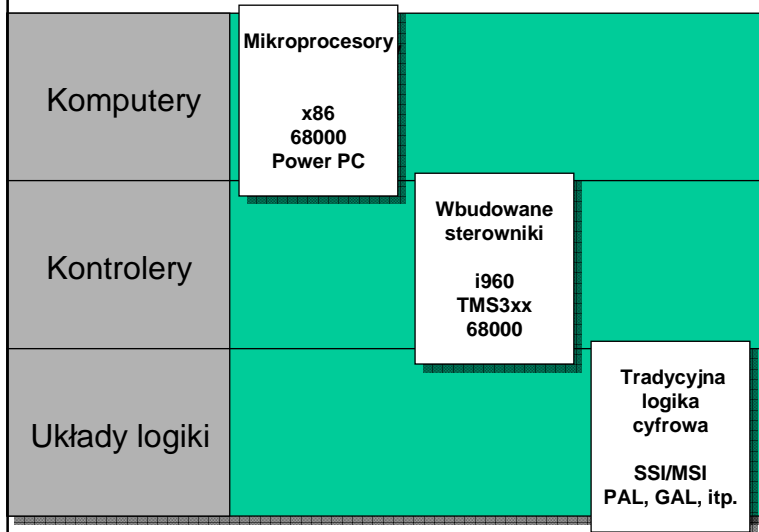
Cypress Semiconductor

Exemplar

Xilinx

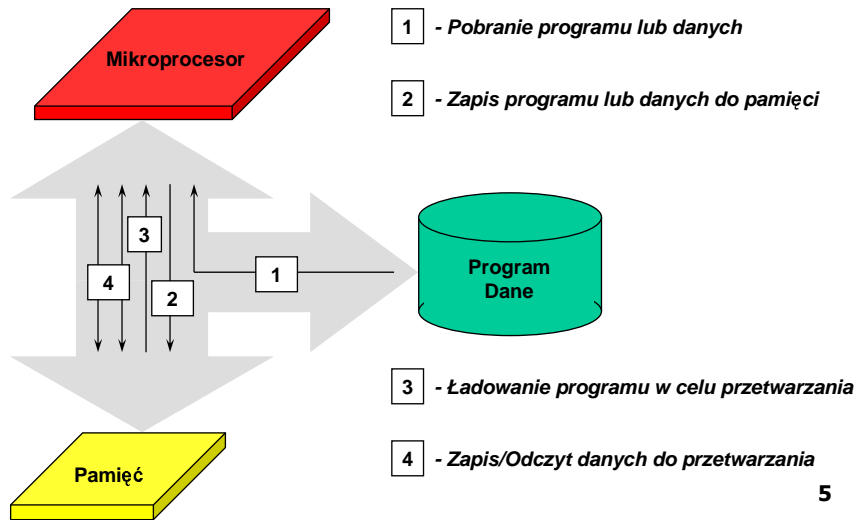
3

Urządzenia o ustalonej strukturze sprzętowej



4

Uproszczona zasada działania tradycyjnego komputera

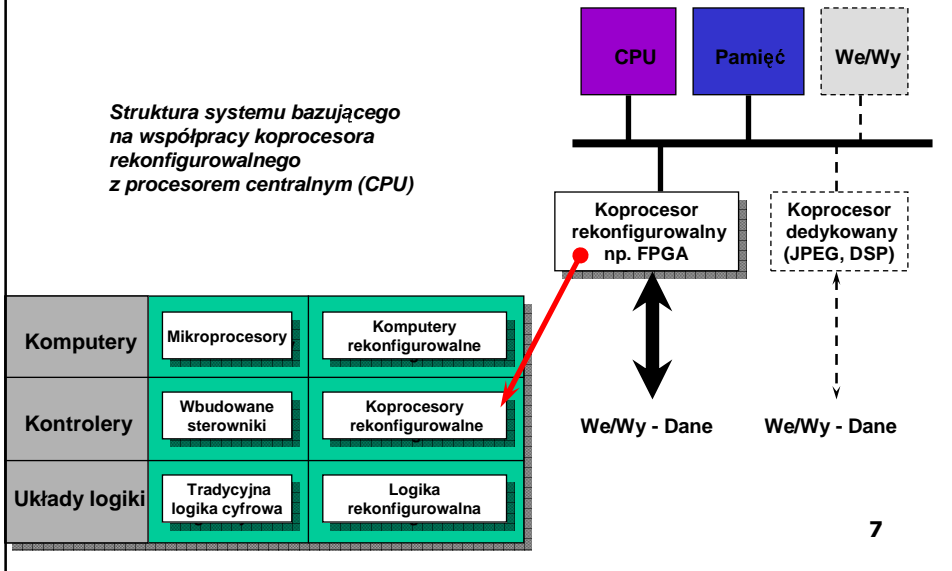


Urządzenia o rekonfigurowalnej strukturze sprzętowej

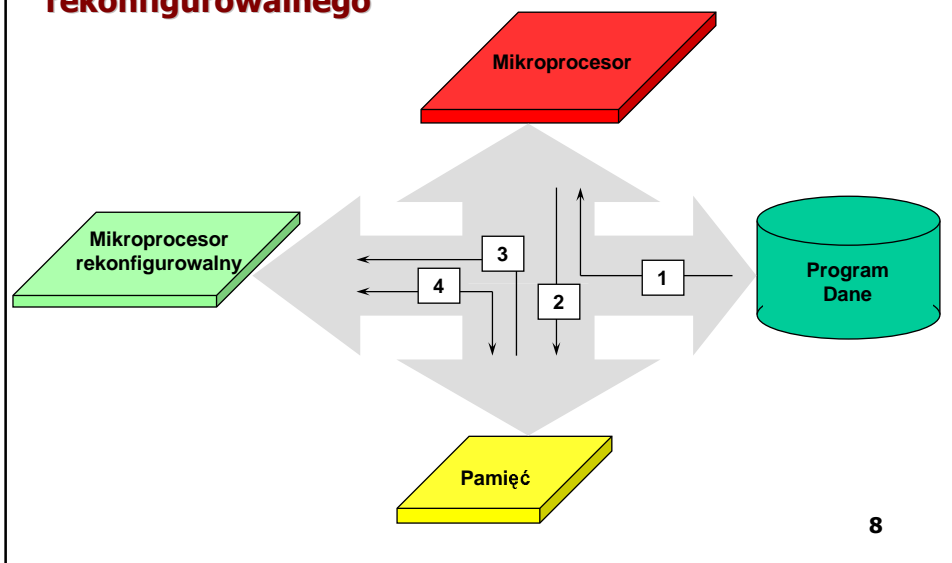
Komputery	Mikroprocesory	Komputery rekonfigurowalne
Kontrolery	Wbudowane sterowniki	Koprocesory rekonfigurowalne
Układy logiki	Tradycyjna logika cyfrowa	Logika rekonfigurowalna

Koprocesory rekonfigurowalne

Struktura systemu bazującego na współpracy koprocesora rekonfigurowalnego z procesorem centralnym (CPU)



Przykładowa architektura koprocesora rekonfigurowalnego



Zastosowania układów rekonfigurowalnych

Zakres zastosowań

Komputery	Mikroprocesory	Komputery rekonfigurowalne
Kontrolery	Wbudowane sterowniki	Koprocesory rekonfigurowalne
Układy logiki	Tradycyjna logika cyfrowa	Logika rekonfigurowalna

Aplikacje

- Układy wspomaganie obliczeń
- Przyspieszanie aplikacji programowych
- Rzeczywistość wirtualna
- Diagnostyka
- Logika sterująca
- ...

9

Cechy ogólne układów FPGA* rodziny XC4000

- pamięć typu Select-RAM™:
 - ultra szybka pamięć z synchronicznym zapisem
 - pamięć dwuportowa
- gwarancja wymagań PCI
- przejrzysty generator funkcji
- wewnętrzna magistrala trójstanowa
- dedykowane linie szybkiego przeniesienia
- hierarchiczna organizacja linii połączeń
- możliwość uzyskania wewnętrznych magistral trójstanowych
- duża liczba przerzutników

* Field Programmable Gate Array

10

Cechy ogólne układów FPGA rodziny XC4000

- Wydajność systemu przekraczająca 80 MHz
- Przejrzysta struktura matryc
- Segmentacja połączeń o niskim poborze mocy
- Cechy zorientowane systemowo
 - implementacja standardu IEEE 1149.1
 - programowalna szybkość narastania napięcia wyjściowego
 - programowane wykorzystanie wejściowych rezystorów podciągających
- Konfiguracja poprzez załadowanie pliku binarnego
 - nieograniczona liczba cykli rekonfigurowania

11

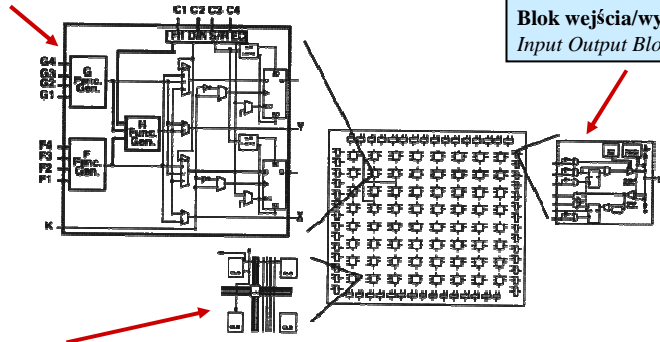
Cechy ogólne układów FPGA* rodziny XC4000

- Możliwość zwrotnego odczytu po konfiguracji układu
 - weryfikacja programu
 - testowanie węzłów wewnątrz układu
- Jednolite środowisko projektowe
 - interfejsy do popularnych narzędzi projektowych
 - automatyczny proces mapowania, rozmieszczania i łączenia
 - interaktywny edytor projektu umożliwiający jego optymalizację

12

Architektura:

Konfigurowalny blok logiczny -
Configurable Logic Block (CLB)

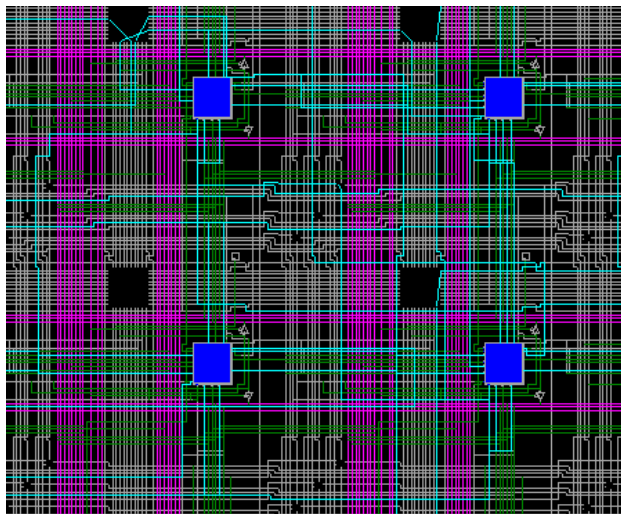


Blok wejścia/wyjścia -
Input Output Block (IOB)

Blok programowalnych połączeń -
Programmable Interconnect

13

A tak to wygląda od środka:



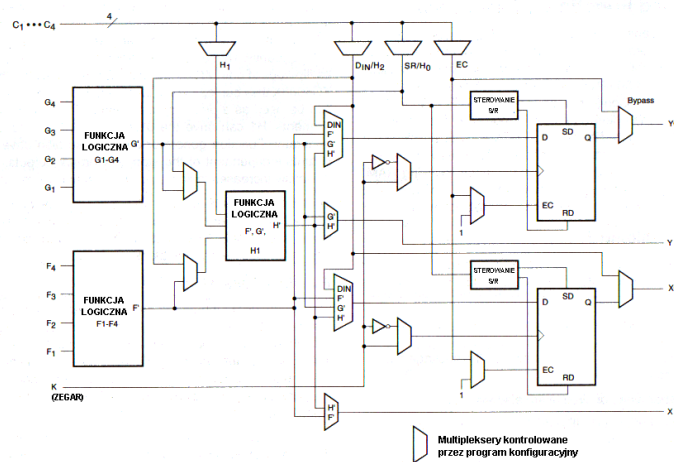
Fragment układu po zaimplementowaniu

14

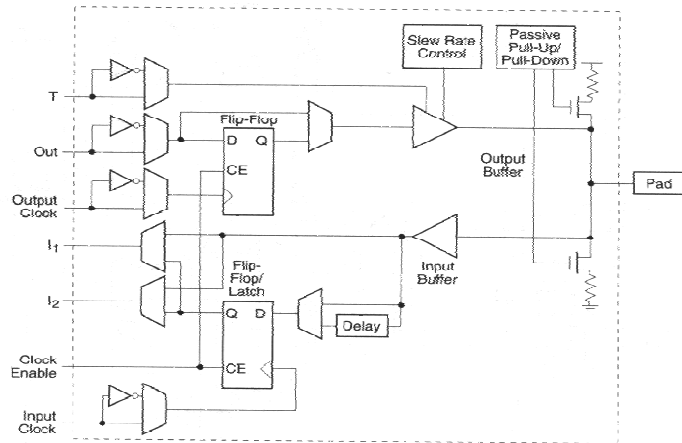
Przegląd układów rodziny XC4000

Urządzenie	Ilość komórek logicznych	Maks. liczba bramek	Maks. ilość bitów RAM	Rozmiar matrycy komórek	Liczba bloków logiki	Maks. liczba we/wy
XC4003E	238	3,000	3,200	10x10	100	80
XC4005E/XL	466	5,000	6,272	14x14	196	112
XC4006E	608	6,000	8,192	16x16	256	128
XC4008E	770	8,000	10,368	18x18	324	144
XC4010E/XL	950	10,000	12,800	20x20	400	160
XC4013E/XL	1368	13,000	18,432	24x24	576	192
XC4020E/XL	1862	20,000	25,088	28x28	784	224
XC4025E	2432	25,000	32,768	32x32	1,024	256
XC4028EX/XL	2432	28,000	32,768	32x32	1,024	256
XC4036EX/XL	3078	36,000	41,472	36x36	1,296	288
XC4044XL	3800	44,000	51,200	40x40	1,600	320
XC4052XL	4598	52,000	61,952	44x44	1,936	352
XC4062XL	5472	62,000	73,728	48x48	2,304	384
XC4085XL	7448	85,000	100,352	56x56	3,136	448

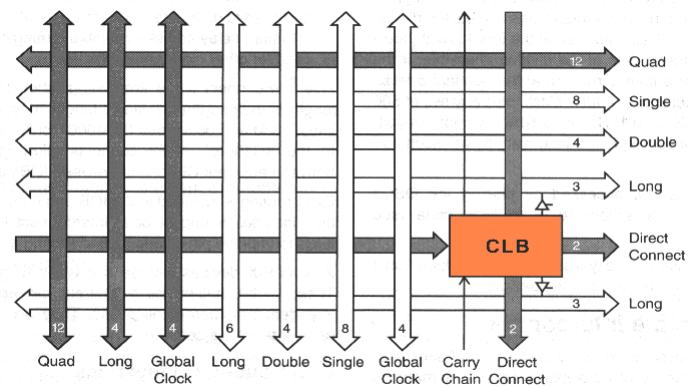
Budowa bloku logicznego (CLB):



Budowa bloku wejścia/wyjścia (IOB):



Programowalna sieć połączeń



Spartan 3 – przegląd

Low-cost, high-performance logic solution for high-volume, consumer-oriented applications

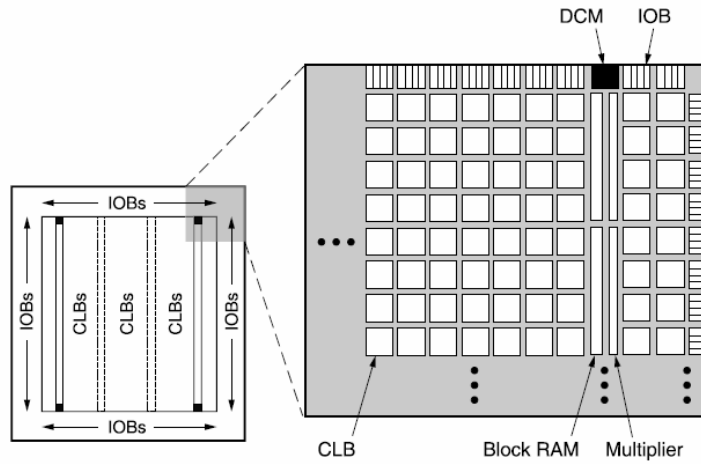
- Densities up to 74,880 logic cells
- SelectIO™ signaling
- Up to 784 I/O pins
- 622 Mb/s data transfer rate per I/O
- 18 single-ended signal standards
- 8 differential I/O standards including LVDS, RSDS
- Termination by Digitally Controlled Impedance
- Signal swing ranging from 1.14V to 3.45V
- Double Data Rate (DDR) support
- DDR, DDR2 SDRAM support up to 333 Mbps
- Logic resources
- Abundant logic cells with shift register capability
- Wide, fast multiplexers
- Fast look-ahead carry logic
- Dedicated 18 x 18 multipliers
- JTAG logic compatible with IEEE 1149.1/1532

- SelectRAM™ hierarchical memory
- Up to 1,872 Kbits of total block RAM
- Up to 520 Kbits of total distributed RAM
- Digital Clock Manager (up to four DCMs)
- Clock skew elimination
- Frequency synthesis
- High resolution phase shifting
- Eight global clock lines and abundant routing
- Fully supported by Xilinx ISE and WebPACK development systems
- MicroBlaze™ and PicoBlaze™ processor, PCI, PCI Express PIPE endpoint, and other IP cores
- Pb-free packaging options
- Automotive Spartan-3 XA Family variant

Spartan 3 – przegląd

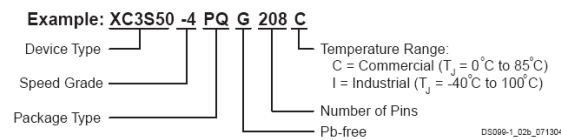
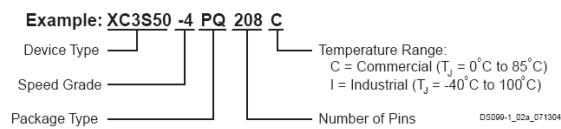
Device	System Gates	Equivalent Logic Cells ¹	CLB Array (One CLB = Four Slices)			Distributed RAM Bits (K=1024)	Block RAM Bits (K=1024)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs						
XC3S50 ²	50K	1,728	16	12	192	12K	72K	4	2	124	56
XC3S200 ²	200K	4,320	24	20	480	30K	216K	12	4	173	76
XC3S400 ²	400K	8,064	32	28	896	56K	288K	16	4	264	116
XC3S1000 ²	1M	17,280	48	40	1,920	120K	432K	24	4	391	175
XC3S1500	1.5M	29,952	64	52	3,328	208K	576K	32	4	487	221
XC3S2000	2M	46,080	80	64	5,120	320K	720K	40	4	565	270
XC3S4000	4M	62,208	96	72	6,912	432K	1,728K	96	4	712	312
XC3S5000	5M	74,880	104	80	8,320	520K	1,872K	104	4	784	344

Spartan 3 – przegląd



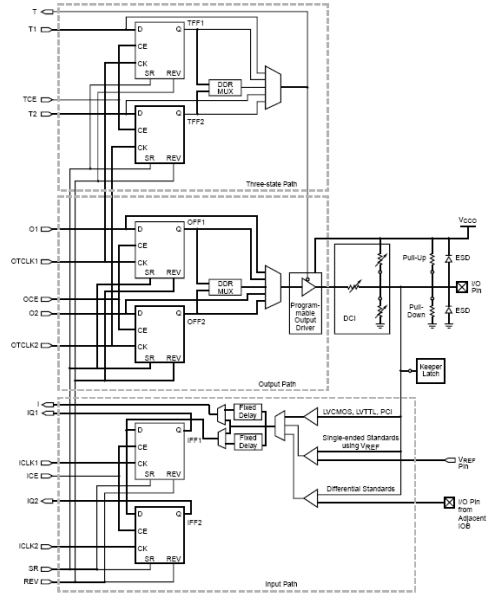
21

Spartan 3 – przegląd

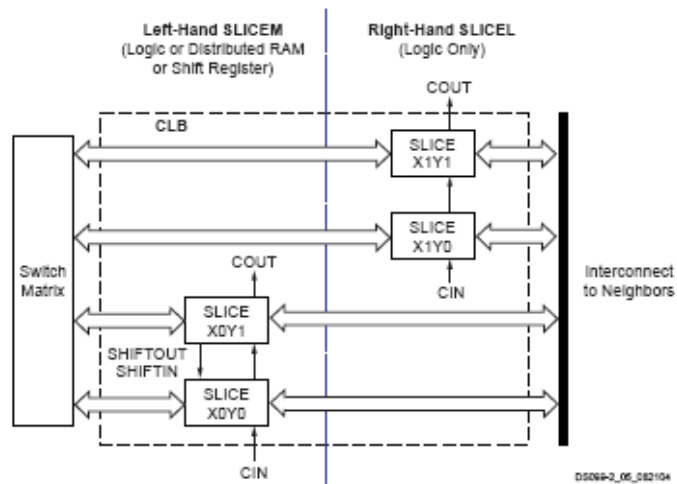


22

Spartan 3 – przegląd

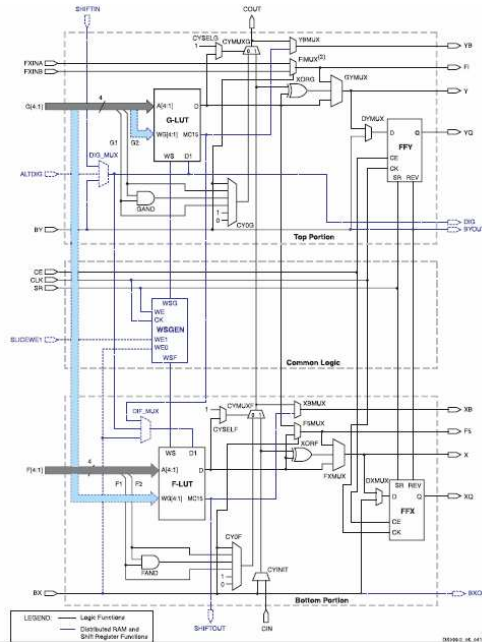


Spartan 3 – przegląd

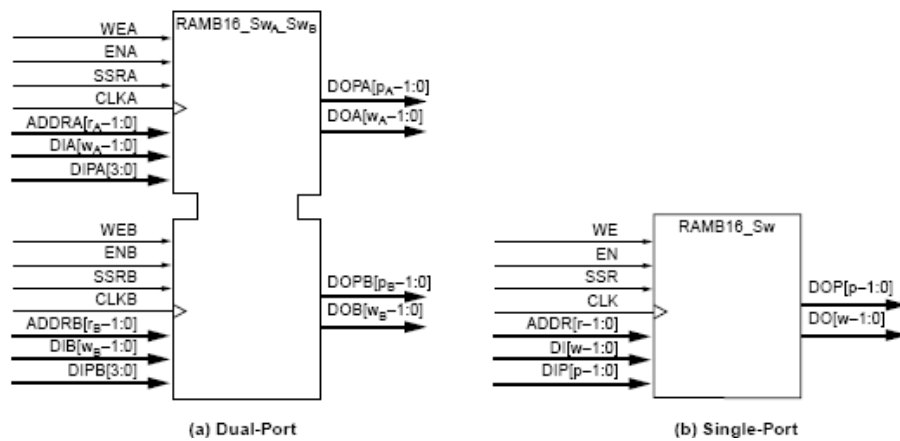


DS099-2_06_S2104

Spartan 3 – przegląd



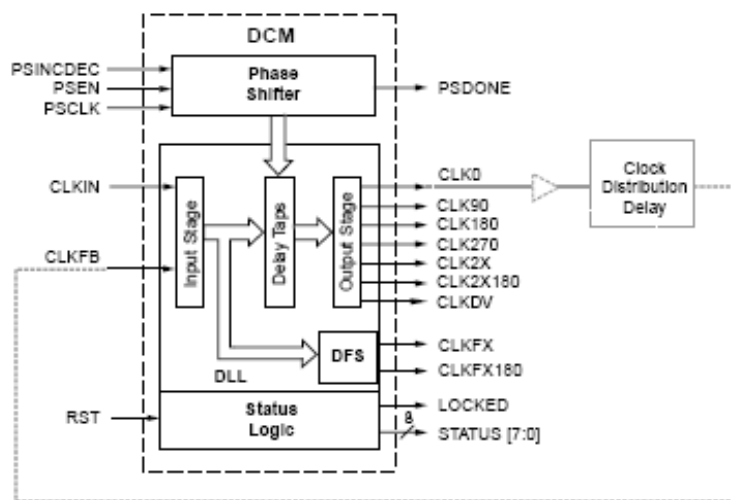
Spartan 3 – przegląd



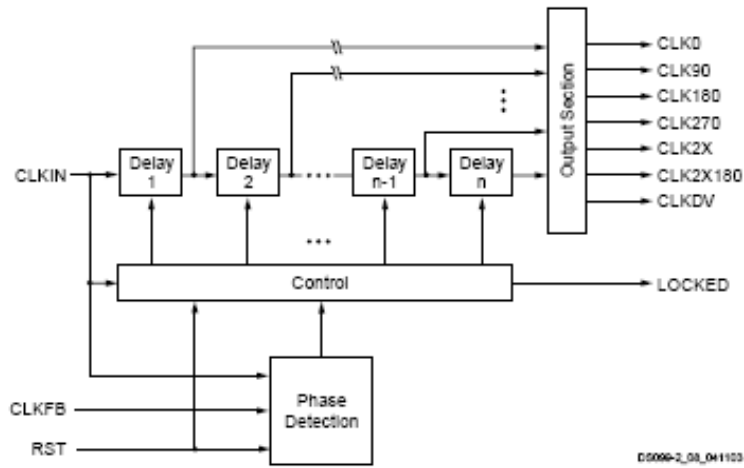
Spartan 3 – przegląd



Spartan 3 – przegląd

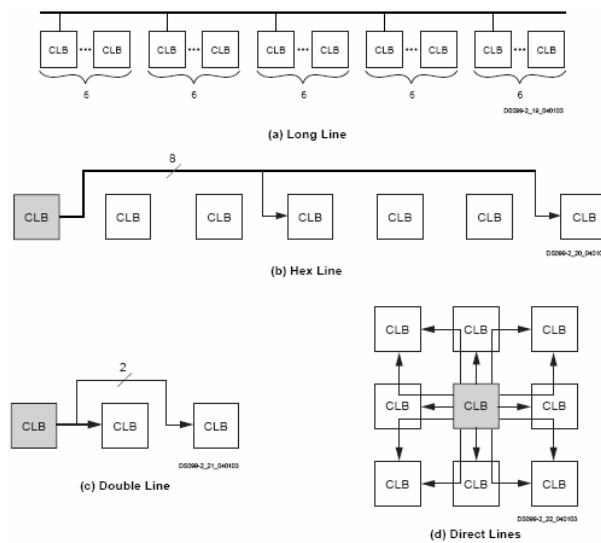


Spartan 3 – przegląd



29

Spartan 3 – przegląd



30

Co to jest VHDL?

Very High Speed Integrated Circuits

Hardware

Description

Language

Do czego służy?

- Niezależna od technologii specyfikacja projektu
- Możliwość współpracy pomiędzy różnymi producentami
- Łatwość zmian funkcjonalnych
- Automatyzacja projektowania (redukcja czasu i kosztów)
- Ułatwienie weryfikacji funkcjonalnej oraz implementacji

31

Historia VHDL

1980 Początek programu rozwijania technologii układów VHSIC (Very High Speed Integrated Circuits)

1981 Konferencja na temat założeń przyszłego standardu HDL

1987 IEEE wydaje opis VHDL 1076

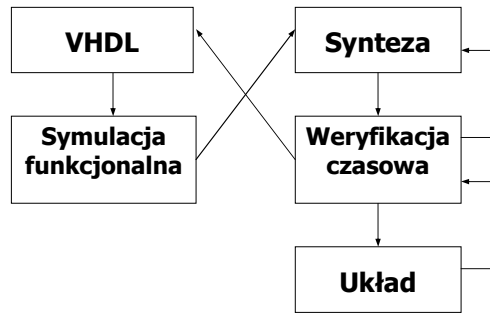
1987 Oficjalny standard VHDL IEEE 1076

1992 Nowelizacja VHDL IEEE 1076-1993

2000 Errata IEEE 1076a-1993

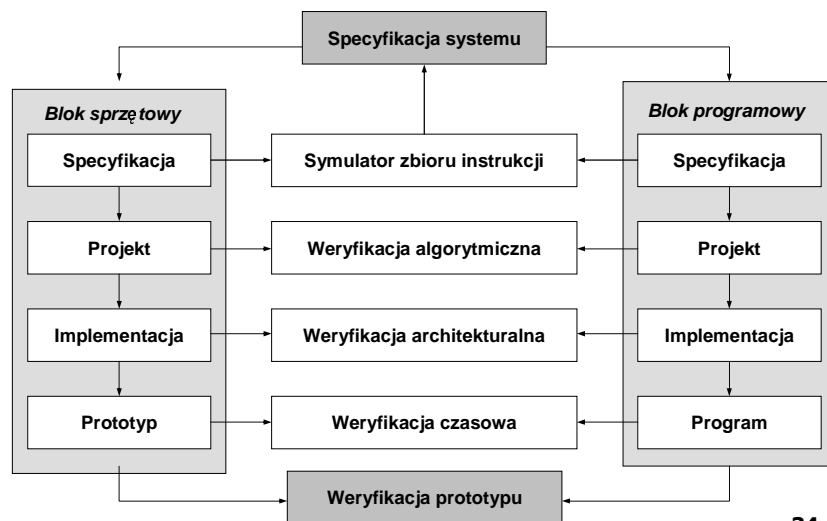
32

Metodyka projektowania z wykorzystaniem VHDL



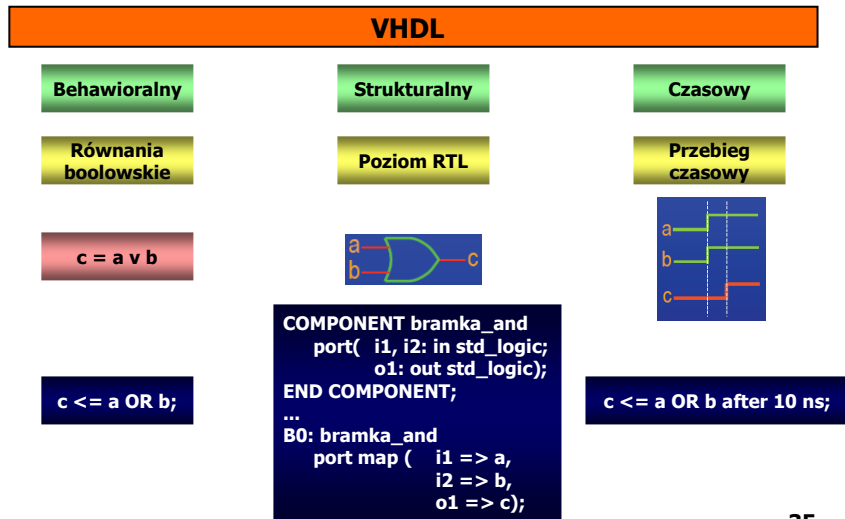
33

Metodyka projektowania systemu



34

Style opisu VHDL

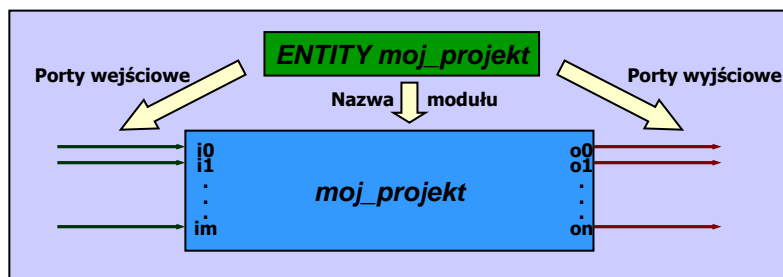


35

Entity – jednostka projektowa

Każdy moduł projektu w VHDL musi zawierać jednostkę projektową *entity*.

Jednostka projektowa (entity) w VHDL oznacza konstrukcję służącą do specyfikacji interfejsu komunikacji ze światem zewnętrznym.



36

Entity – jednostka projektowa

Każda jednostka projektowa (*entity*) w projekcie musi posiadać unikalną nazwę

Wejścia i wyjścia w projekcie noszą nazwę portów (PORTS)

Typy portów:

in – port wejściowy

out – port wyjściowy

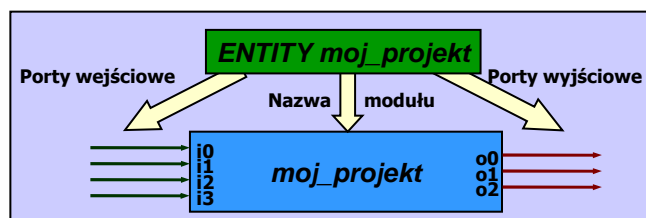
inout – port dwukierunkowy, wykorzystywany w sytuacji, gdy dane przekazywane są przez moduł bez ich zmiany

37

Entity – jednostka projektowa

Deklaracja entity

```
entity moj_projekt is
  port(
    i0, i1, i2, i3: in bit;
    o0, o1, o2: out bit;
  );
end moj_projekt;
```



38

Generic – parametry ogólne

Wykorzystywana do parametryzacji projektu – jednostki projektowej

Składa się z ogólnej listy interfejsu (*generic interface list*), w której zdefiniowane są parametry jednostki projektowej (*formal generic constants*)

Elementy składowe ogólnej listy interfejsu mogą być jedynie stałymi rodzaju *in* (rodzaj jest pomijany w części deklaracyjnej)

Użycie generic

Składnia generic

Brak określenia portu

```
entity projekt_testowy is
  generic(N: integer);
  port(
    a_in, b_in: in bit_vector(N-1 downto 0);
    stat: in bit;
    y_out: out bit_vector (N-1 downto 0);
  );
end projekt_testowy;
```

Deklaracja *generic* jest widoczna dla wszystkich architektur opartych na danej jednostce projektowej (*entity*)

Architecture – implementacja jednostki projektowej

Ciało architektury (*architecture*) definiuje sposób działania jednostki projektowej od pobrania danych z portów wejściowych do wygenerowania danych na portach wyjściowych.

Istnieje możliwość przyporządkowania wielu architektur do jednej jednostki projektowej:

- cecha użyteczna w podejściu top down – dedykowana architektura opisuje kolejne stopnie redefinicji na różnych poziomach abstrakcji
- Potrzeba opisu różnych wariantów tego samego modułu

41

Architektura

Składnia architecture

```
architecture behav of projekt_testowy is
  signal a, b, c: bit;

  begin
    suma <= a XOR b;
  end behav;
```

42

Przykład prostego projektu

```
entity and2_gate is
  port(in_1, in_2: in bit;
        o_slow:  out bit;
        o_fast:  out bit);
end and2_gate;

architecture gates of and2_gate is
  begin
    o_slow <= in_1 and in_2 after 5;
    o_fast <= in_1 and in_2;
  end gates;
```

