



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



## **„Układy reprogramowalne i SoC” „Implementacja w układach FPGA”**

Prezentacja jest współfinansowana przez  
Unię Europejską w ramach  
Europejskiego Funduszu Społecznego w projekcie pt.

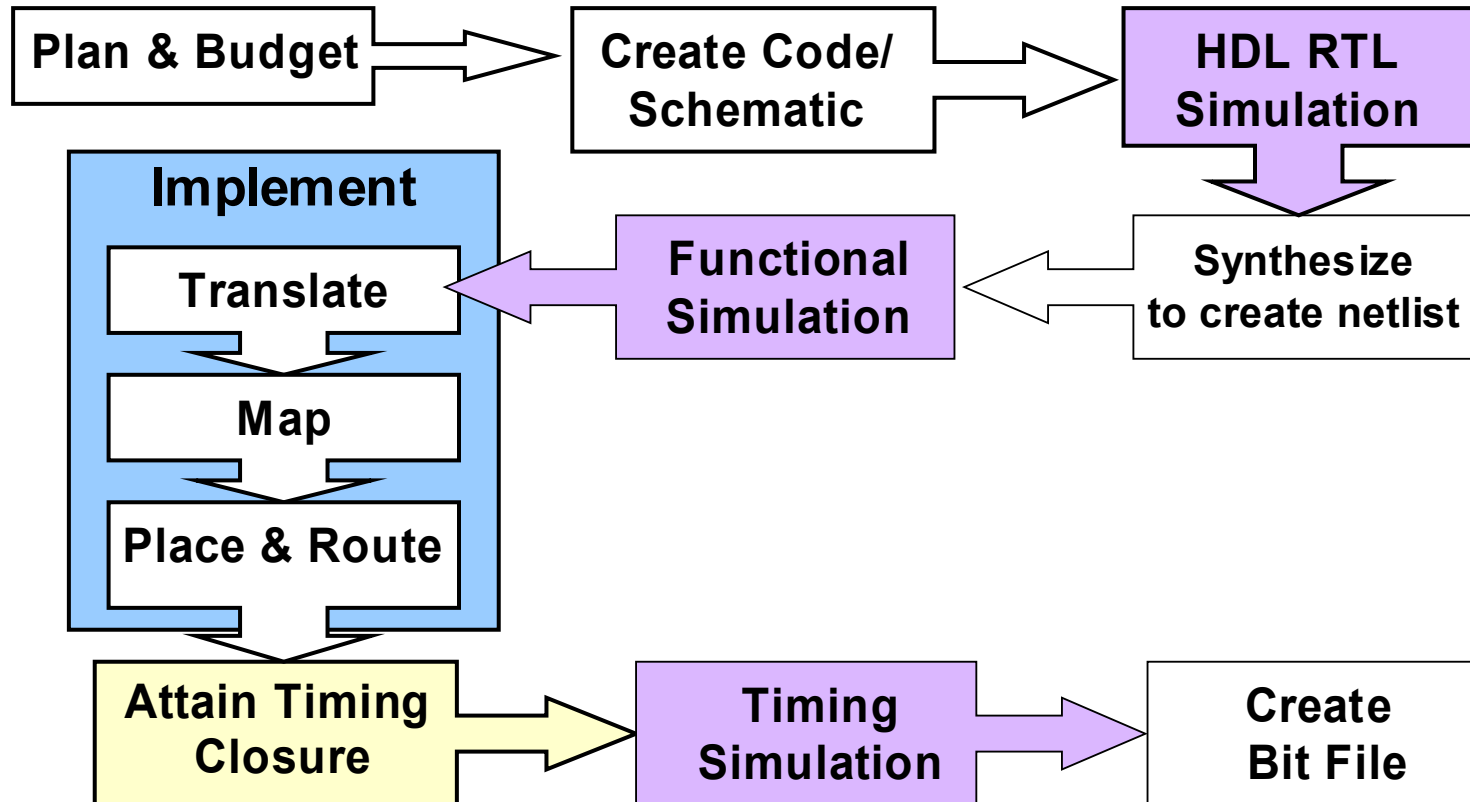
*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -  
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do  
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie





# Proces projektowania dla FPGA firmy Xilinx





## Etapy implementacji

- Synthesize: zamiana opisu np. w języku VHDL na odpowiedni układ złożony z bramek logicznych i przerzutników
- Translate: połączenie wielu plików projektu w pojedynczą netlistę
- Map: odwzorowanie logicznych elementów netlisty (bramki, przerzutniki) na komponenty fizyczne (CLB, IOB)
- Place & Route: rozmieszczenie komponentów na powierzchni układu scalonego i połączenie ich ze sobą oraz ekstrakcja informacji o opóźnieniach w układzie
- Timing closure: spełnienie założeń co do wymaganej częstotliwości zegara, opóźnień itp.
- Timing (Sim): Wygenerowanie netlisty z informacją o opóźnieniach dla narzędzi symulacyjnych
- Configure: wygenerowanie pliku konfiguracyjnego dla układu FPGA





## Ograniczenia projektowe

- Różne wymagania dla projektu są zadawane w pliku UCF (User Constraints File)
  - wyprowadzenia układu (który sygnał na której końcówce)
  - wymagania czasowe
    - minimalna częstotliwość zegara
    - maksymalne opóźnienia
  - wymagania na rozmieszczenie komponentów



- Xilinx ISE to zintegrowane środowisko do implementacji układów cyfrowych w układach FPGA i CPLD tej firmy
- Wszystkie operacje wykonywane za pomocą IDE można również wykonać z linii poleceń
- Pierwszy etap: tworzymy nowy projekt (File -> New project) i wybieramy typ układu FPGA, w którym będziemy go implementować
  - Family: Spartan 3A and Spartan 3AN
  - Device: XC3S700A
  - Package: FG484
  - Speed: -4
- Drugi etap: dodajemy do projektu pliki VHDL i UCF



## Przykład: sumator

### adder.vhd

```
ENTITY full_adder IS
PORT (a, b, cin: IN BIT;
      s, cout: OUT BIT);
END full_adder;
-----
ARCHITECTURE dataflow OF full_adder IS
BEGIN
    s <= a XOR b XOR cin;
    cout <= (a AND b) OR (a AND cin) OR
            (b AND cin);
END dataflow;
```

### s3starter-adder.ucf

```
NET "s"          LOC = "R20" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;
NET "cout"       LOC = "T19" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;

NET "a"          LOC = "T16" | IOSTANDARD = LVCMOS33 | PULLDOWN ;
NET "b"          LOC = "T14" | IOSTANDARD = LVCMOS33 | PULLDOWN ;
NET "cin"        LOC = "T15" | IOSTANDARD = LVCMOS33 | PULLDOWN ;
```





## Implementacja i programowanie

- Implementacja
  - Okno "Processes"
    - Dwuklik na "Generate programming file"
- Programowanie FPGA
  - Podłączenie płyty do komputera kablem USB
  - Włączenie zasilania płyty
  - Dwuklik na "Configure target device" w oknie "Processes"
    - Zgoda na otwarcie programu IMPACT
    - Wybór pliku full\_adder.bit dla pierwszego układu w łańcuchu
    - Wybór "Bypass" dla drugiego układu w łańcuchu
    - Dwuklik na "Program" w oknie "Configuration operations"



## Przykład: migacz

```
ENTITY blinker IS
PORT (clk: IN BIT;
      led0, led1: OUT BIT);
END blinker;
-----
ARCHITECTURE rtl OF blinker IS
BEGIN

  process (clk)
    variable cnt: integer range 0 to 50_000_000;
  begin
    if clk'event and clk='1' then
      if cnt = 0 then
        led0 <='0';
        led1 <='0';
      end if;
      if cnt = 25_000_000 then
        led0 <='1';
      end if;
      if cnt = 40_000_000 then
        led1 <='1';
      end if;
      cnt := cnt + 1;
      if cnt=50_000_000 then
        cnt:=0;
      end if;
    end if;
  end process;
END architecture rtl;
```

```
NET "led0" LOC = "R20" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;
NET "led1" LOC = "T19" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;

NET "CLK" LOC = "E12"| IOSTANDARD = LVCMOS33 ;
# Define clock period for 50 Mhz oscillator
NET "CLK" PERIOD = 20.0ns HIGH 50% ;
```







**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



## **„Układy reprogramowalne i SoC” „Implementacja w układach FPGA”**

Prezentacja jest współfinansowana przez  
Unię Europejską w ramach  
Europejskiego Funduszu Społecznego w projekcie pt.

*„Innowacyjna dydaktyka bez ograniczeń - zintegrowany rozwój Politechniki Łódzkiej -  
zarządzanie Uczelnią, nowoczesna oferta edukacyjna i wzmacniania zdolności do  
zatrudniania osób niepełnosprawnych”*

Prezentacja dystrybuowana jest bezpłatnie

