



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



„Układy reprogramowalne i SoC” „Projektowanie systemów wbudowanych z użyciem EDK”

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





Systemy wbudowane w układach FPGA

- System wbudowany posiada następujące cechy:
 - Jedna funkcja
 - Rygorystyczne wymagania
 - Niski koszt
 - Mało elementów
 - Wystarczająco szybki do założonych celów
 - Zużywający niewiele energii
 - Reagujący na zdarzenia zewnętrzne w czasie rzeczywistym
 - Wymaga współprojektowania sprzętu i oprogramowania

Systemy wbudowane w układach FPGA

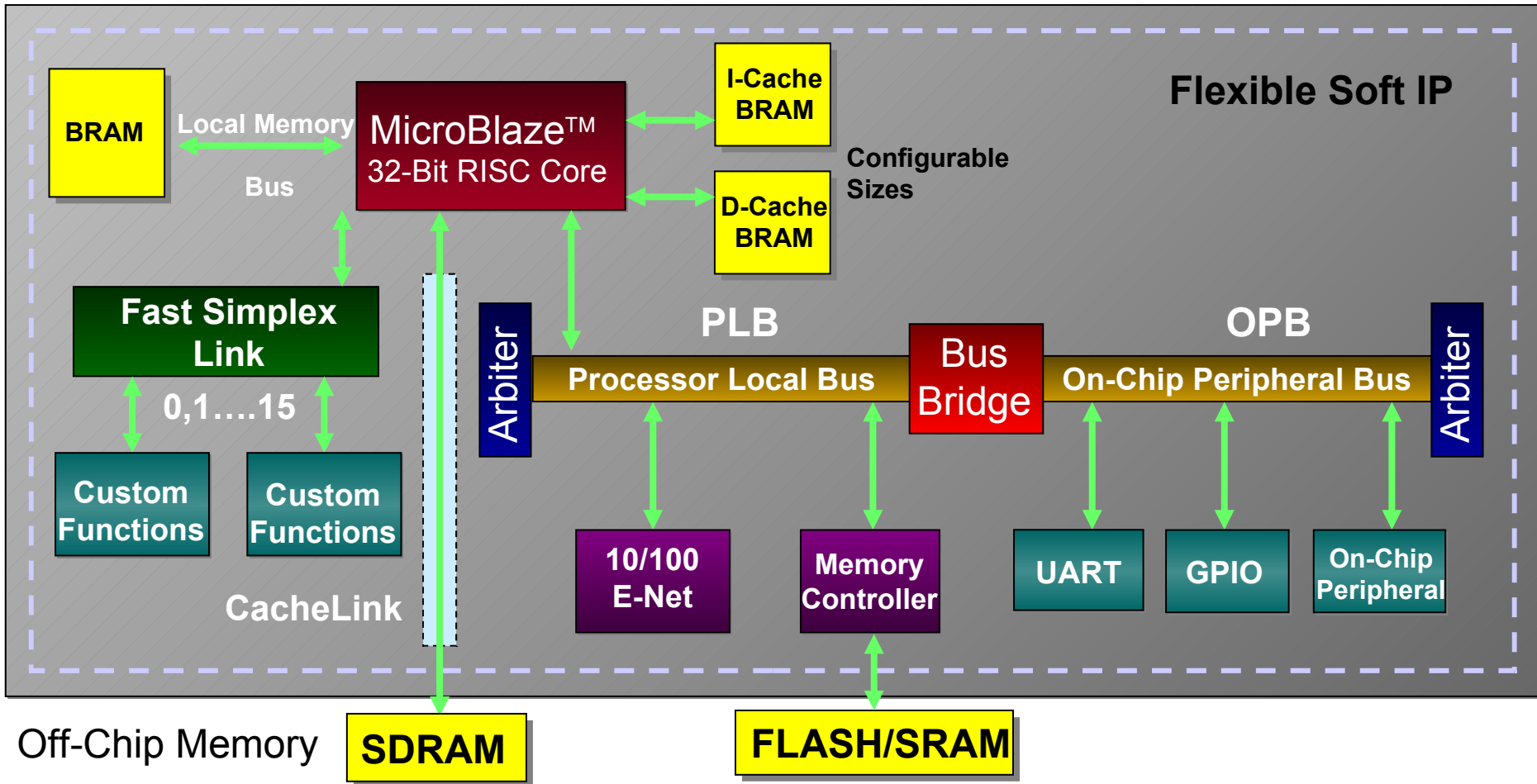
- Wielu producentów FPGA oferuje rozwiązania pozwalające na skonstruowanie zintegrowanego systemu wbudowanego
- Procesor może być zaimplementowany zarówno wykorzystując zasoby FPGA, jak i specjalizowane bloki sprzętowe
 - Altera NIOS
 - Xilinx MicroBlaze
 - Xilinx PowerPC

Systemy wbudowane w układach FPGA

- Projektowanie systemu wbudowanego w układzie FPGA obejmuje następujące czynności:
 - Zaprojektowanie sprzętu implementowanego w FPGA
 - Wygenerowanie bibliotek i sterowników urządzeń
 - Opracowanie oprogramowania
 - Procedury programowe
 - Procedury obsługi przerwań (opcjonalnie)
 - System operacyjny, system operacyjny czasu rzeczywistego (opcjonalnie)

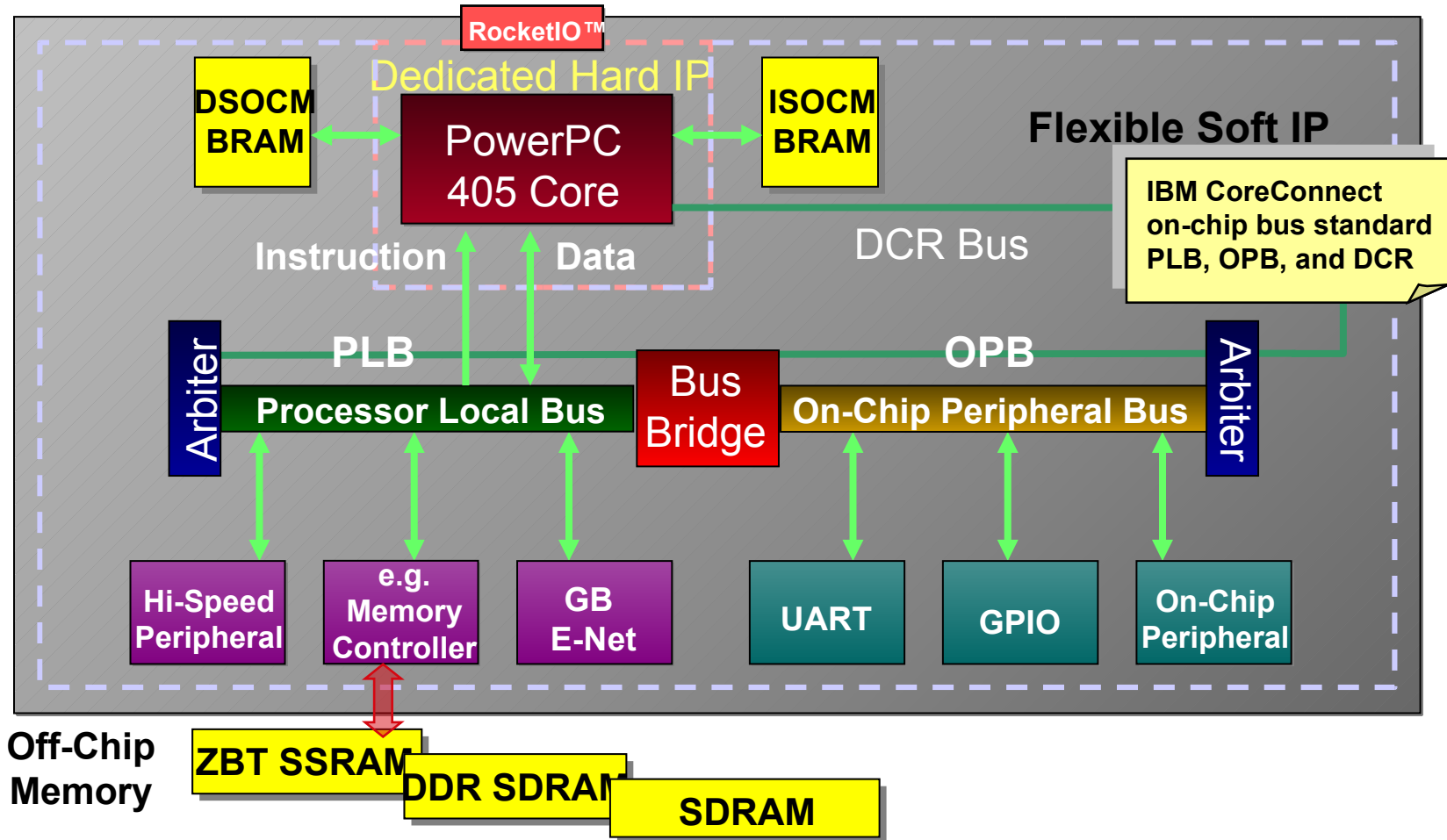


System wbudowany w oparciu o procesor MicroBlaze





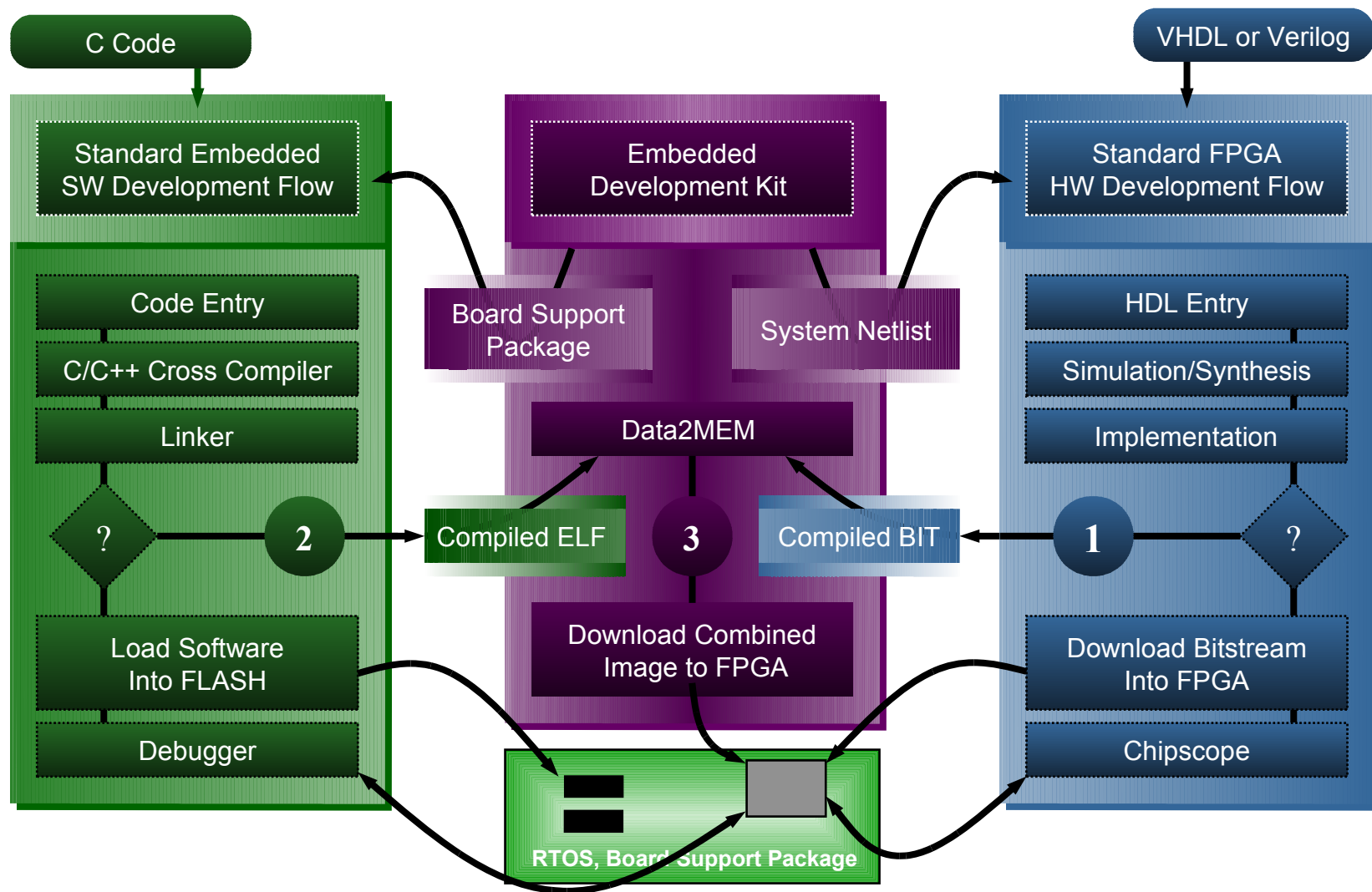
System wbudowany w oparciu o procesor PowerPC



- Co to jest Embedded Development Kit (EDK)?
 - Pakiet oprogramowania firmy Xilinx do projektowania złożonych programowalnych systemów wbudowanych
 - Zawiera narzędzia, dokumentację i bloki IP potrzebne do projektowania systemów zawierających wbudowane procesory IBM PowerPC™ lub Xilinx MicroBlaze™
 - Pozwala na integrację sprzętu i oprogramowania w systemie wbudowanym



Przebieg projektowania z użyciem EDK





Przebieg projektowania z użyciem EDK

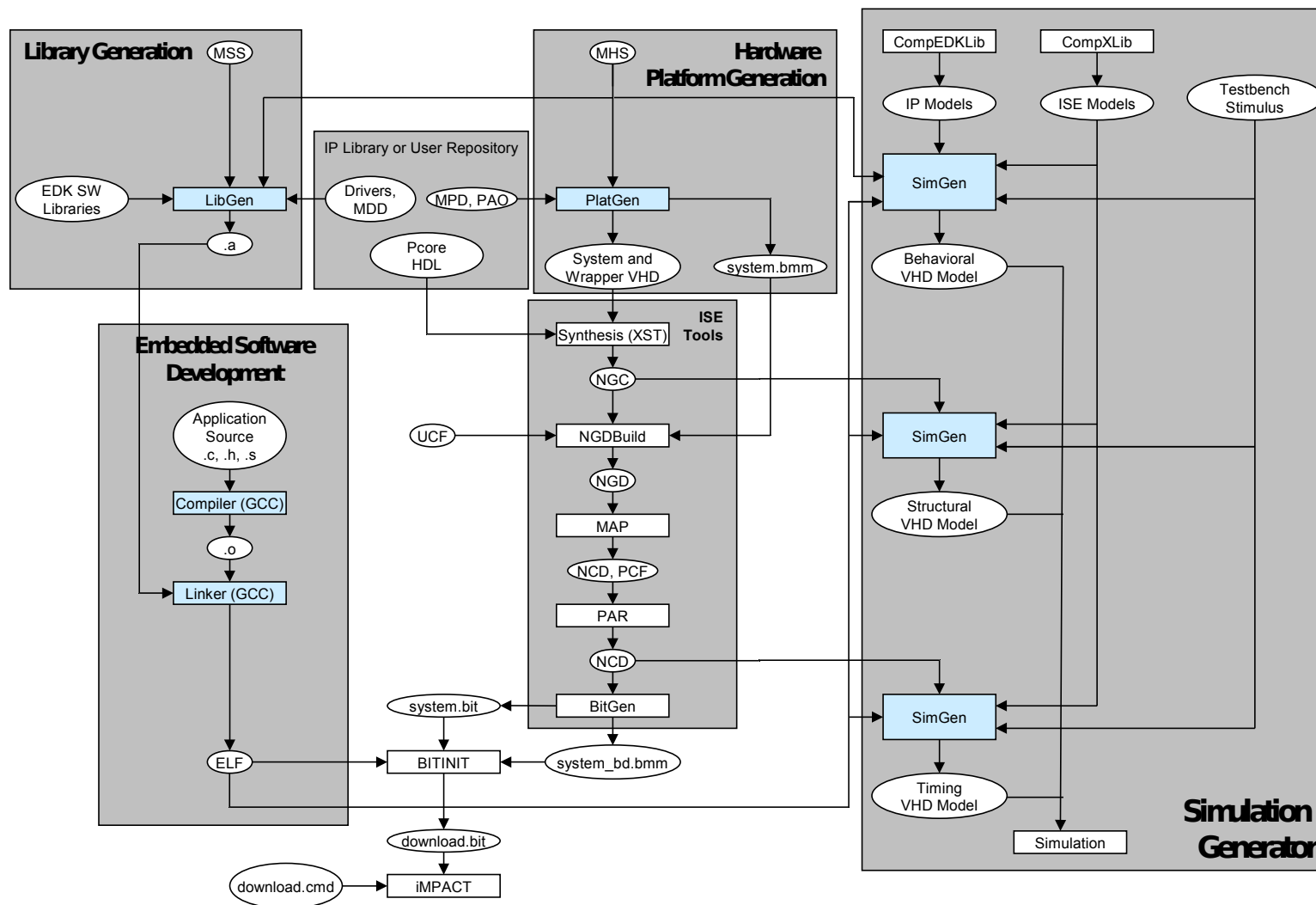
- Projektowanie sprzętu
 - Zaprojektowanie sprzętu z użyciem Base System Builder Wizard
 - Rozbudowa sprzętu
 - Dodanie urządzeń peryferyjnych z katalogu IP
 - Zaprojektowanie i dodanie własnych urządzeń peryferyjnych z użyciem Create/Import Peripherals Wizard
 - Wstawienie modułów ChipScope™ Pro do systemu z użyciem Debug Configuration Wizard
 - Generacja netlisty HDL z użyciem PlatGen
 - Symulacja HDL z użyciem symulatora HDL simulator
 - Generacja modeli symulacyjnych z użyciem SimGen
- Projektowanie oprogramowania
 - Wygenerowanie bibliotek i sterowników z użyciem LibGen
 - Projektowanie i uruchamianie oprogramowania w XPS
 - Kompilacja z użyciem kompilatora GNU C/C++ (gcc)
 - Połączenie z płytą używając Xilinx Microprocessor Debug (XMD)
 - Uruchamianie z użyciem debugera GNU (gdb)
- Praca ze sprzętem
 - Generacja bitstreamu i konfiguracja FPGA
 - Uaktualnienie pamięci programu w FPGA z użyciem BitInit
 - Inicjalizacja zewnętrznej pamięci flash
 - Zapis do zewnętrznej pamięci flash z użyciem programu Flash Writer
 - Generacja pliku konfiguracyjnego dla karty Compact Flash z użyciem GenACE



Narzędzia projektowe w systemie EDK

- Biblioteki
 - lwIP Library – biblioteka sieciowa
 - LibXil MFS – Memory File System
 - LibXil FATfs – FAT file system
 - LibXil Flash – obsługa pamięci flash
 - Standardowe biblioteki C (libc, libm)
- Operating System Board Support Packages (BSPs)
 - Standalone operating system
 - Xilinx MicroKernel (XMK)

Narzędzia projektowe w systemie EDK





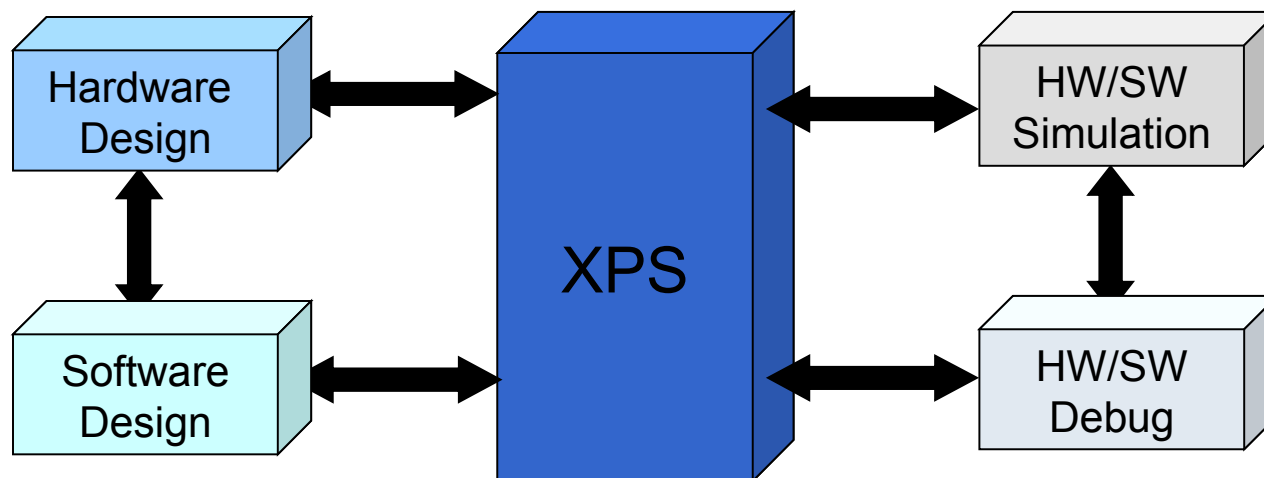
Xilinx Platform Studio (XPS)

The screenshot shows the Xilinx Platform Studio (XPS) interface. The top window title is "Xilinx Platform Studio - Workshops/courses/v92Embedded/sp3ekit/solutions...". The interface includes a menu bar (File, Edit, Debug, Simulation, Window, Help), a toolbar, and several panels:

- Project Information Area:** Contains tabs for Project, Applications, and IP Catalog. A callout "Access project files" points to the Project tab.
- IP Catalog:** A list of IP blocks. A callout "Select cores from the IP catalog" points to this area.
- System Assembly View:** A central diagram showing the hardware system with callouts "Connect the hardware system" and "View a block diagram of the system".
- Table of IP Blocks:** A table listing various IP blocks and their connections.
- Console:** A window at the bottom left showing output text: "(0x84400000-0x8440ffff) debug module mb_plb".

Name	Bus Connection	IP Type	IP Ver
microblaze_0		microblaze	7.00.a
ilmb		lmb_v10	1.00.a
dilmb		lmb_v10	1.00.a
mb_plb		plb_v46	1.00.a
dilmb_cntrl		lmb_bram_if_cntrl	2.10.a
ilmb_cntrl		lmb_bram_if_cntrl	2.10.a
BRAM_PORT	ilmb_port		
SLMB	ilmb		
lmb_bram		bram_block	1.00.a
dip		xps_gpio	1.00.a
push		xps_gpio	1.00.a
RS232_DCE		xps_uartlite	1.00.a
LEDs_8Bit		xps_gpio	1.00.a
debug_module		mdm	1.00.a

- Zarządzanie projektem
 - Utworzenie plików Microprocessor Hardware Specification (MHS) i Microprocessor Software Specification (MSS)
 - Plik Xilinx Microprocessor Project (XMP)
- Zarządzanie programami
- Zarządzanie platformą projektową
 - Konfiguracja narzędzi
 - Konfiguracja platformy programowej
 - Uruchamianie narzędzi
 - Uruchamianie i symulacja





Utworzenie/otwarcie projektu

- Utworzenie nowego projektu
 - File → New Project
 - Opcja Base System Builder
 - Base System Builder (BSB) wizard pomaga zbudować działający system dla danej płyty uruchomieniowej
 - Wybieramy Select Blank XPS Project
- Otwarcie istniejącego projektu
 - File → Open Project
 - Przechodzimy do katalogu z projektem i wybieramy plik xmp
 - File → New Project
 - Wybieramy opcję Open a Recent Project option a następnie jeden z ostatnio otwartych projektów
- Dane o projekcie są zapisywane w pliku Xilinx Microprocessor Project (XMP)

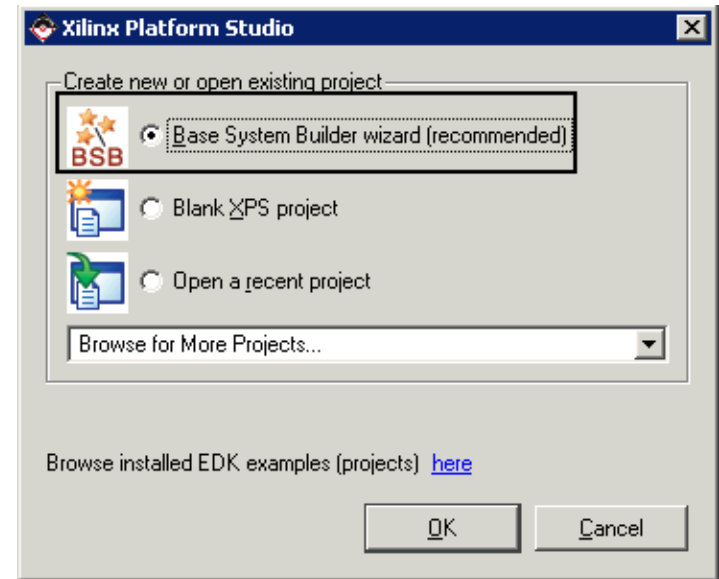
Używanie Base System Builder

1. Wybieramy płytę docelową
2. Wybieramy procesor
3. Konfigurujemy procesor
4. Wybieramy i konfigurujemy interfejsy I/O
5. Dodajemy wewnętrzne peryferia
6. Generujemy programy systemowe i skrypt linkera
7. Generujemy projekt

– Generowane pliki:

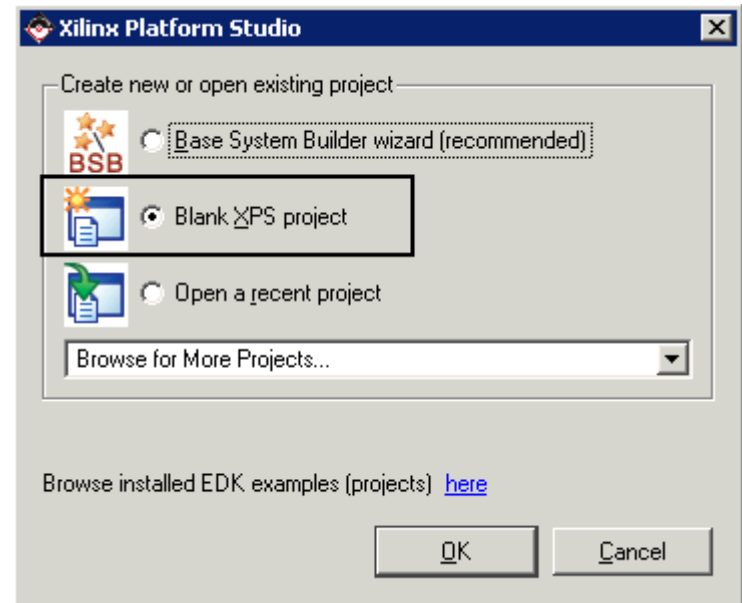
- system.mhs
 - system.xmp
 - etc/fast_runtime.opt
 - katalog pcore (pusty)
- opcjonalny katalog TestApp_Memory/src zawierający
- TestApp_Memory.c
- opcjonalny katalog TestApp_Peripheral/src zawierający
- TestApp_Peripheral.c

system.mss
data/system.ucf
etc/download.cmd
system.bsb (opcjonalnie)
TestApp_Memory_LinkScr.ld
TestApp_Peripheral_LinkScr.ld



Utworzenie projektu od podstaw

- Określamy położenie nowego projektu XPS
 - Wybieramy docelowy układ FPGA, opcjonalnie importujemy plik MHS i/lub katalog repozytorium użytkownika
- Z użyciem katalogu IP dodajemy procesor i peryferia
- Konfigurujemy procesor i peryferia
- Tworzymy plik UCF
- Konfigurujemy oprogramowanie dla komponentów sprzętowych
- Projektujemy oprogramowanie
- Generujemy bitstream
- Ładujemy i uruchamiamy bitsteram





Projektowanie sprzętu

- Platform Generator – PlatGen
 - Plik wejściowy → MHS i MPD
 - Plik MHS definiuje konfigurację wbudowanego systemu mikroprocesorowego, m.in. architekturę magistral, urządzenie peryferyjne i procesory, priorytety przerwań i przestrzenie adresowe
 - Plik MPD definiuje interfejsy, konfigurowalne parametry i ich domyślne wartości oraz dostępne porty urządzenia peryferyjnego
 - Pliki wyjściowe → netlista systemu, netlisty urządzeń peryferyjnych, plik BMM
 - Tworzy katalogi dla syntezy, plików HDL i implementacji
 - Generuje pliki HDL dla urządzeń peryferyjnych
 - Generuje plik HDL opisujący cały system
 - Pobiera netlisty urządzeń peryferyjnych z katalogu instalacyjnego EDK
 - Generuje plik BMM
 - Wywołuje program XST aby zsyntezować plik opisujący system



Implementacja sprzętu

- Netlisty opisujące sprzęt muszą być zaimplementowane z użyciem narzędzi Xilinx
- Możliwe dwa podejścia: ręczne (ISE™ Project Navigator) lub automatyczne (Xflow batch tool)
 - ISE Project Navigator GUI umożliwia bezpośredni dostęp do narzędzi Xilinx
 - Xflow jest programem wsadowym, ukrywającym szczegóły implementacyjne
 - Niezależny od układu FPGA i posiada prosty, elastyczny i rozszerzalny interfejs do narzędzi Xilinx



Generacja bibliotek

- Library Generator – LibGen
 - Pliki wejściowe → MSS
 - Pliki wyjściowe → libc.a, libXil.a, libm.a
- LibGen to narzędzie do konfiguracji bibliotek i sterowników urządzeń
 - Plik MSS definiuje sterowniki skojarzone z peryferiami, standardowe urządzenia wejścia/wyjścia i inne opcje oproramowania
- LibGen konfiguruje biblioteki i sterowniki według tych informacji i generuje biblioteki:
 - libc.a - Standard C library
 - libXil.a - Xilinx library
 - libm.a - Math functions library



Projektowanie oprogramowania

- Kompilacja plików źródłowych
 - Pliki wejściowe → *.c, *.c++, *.h, libc.a, libXil.a, libm.a
 - Pliki wyjściowe → executable.elf
 - Dla każdego programu wywoływany jest kompilator i tworzone są pliki wykonywalne dla każdego z procesorów
 - Cztery etapy:
 - Pre-processor: Zastępuje makrodefinicje, włącza pliki
 - Kompilator specyficzny dla procesora i języka: kompiluje kod C/C++
 - Asembler: konwertuje kod na język maszynowy i generuje pliki wynikowe
 - Linker: łączy pliki wynikowe i biblioteki z użyciem domyślnego lub zdefiniowanego przez użytkownika skryptu linkera



Połączenie sprzętu i oprogramowania

- BitInit – uaktualnienie bitstreamu
 - Pliki wejściowe → system_bd.bmm, system.bit, executable.elf
 - Plik wyjściowy → download.bit
 - Wywoływane jest narzędzie data2MEM, inicjalizujące pamięć programu procesora w bitstreamie
 - Pamięć programu może być zainicjalizowana przez pustą pętlę (bootloop), bootloader lub właściwy program
 - Na tym etapie łączy się projekt sprzętu i oprogramowania.

- Załadowanie bitstreamu do FPGA
 - Plik wejściowy → download.bit
 - Plik download.bit jest ładowany na płytę z użyciem narzędzia Xilinx iMPACT w trybie wsadowym
 - XPS używa pliku etc/download.cmd w celu załadowania bitstreamu
 - Plik download.cmd zawiera takie informacje, jak typ kabla i pozycja układu FPGA w łańcuchu JTAG



Zarządzanie platformą przez XPS

- Zadania zarządzania platformą to:
 - Generacja sprzętu (PlatGen)
 - Konfiguracja bibliotek i sterowników urządzeń (LibGen)
 - Generacja modelu symulacyjnego (SimGen)
 - Implementacja (Xflow lub ISE™)
 - Kompilacja (GNU Compiler)
 - Inicjalizacja bistreamu (Data2MEM)
- XPX umożliwia następujące procesy specyfikacji i konfiguracji systemu i oprogramowania:
 - Dodawanie modułów IP, edycja parametrów modułów, połączenie magistral i portów z użyciem System Assembly view
 - Generacja i modyfikacja pliku MSS za pomocą Software Platform Settings



Zarządzanie oprogramowaniem

- XPS umożliwia tworzenie aplikacji testowych i zarządzanie skryptami linkera przy pomocy BSB
- XPS pozwala na specyfikację wielu projektów oprogramowania w oknie Applications
- XPS posiada wbudowany edytor dla plików w języku C i plików nagłówkowych
- Pliki źródłowe są pogrupowane według egzemplarzy procesorów i można nimi zarządzać dla każdego procesora niezależnie
- Wszystkie pliki źródłowe dla danego procesora są kompilowane kompilatorem wyspecyfikowanym dla tego procesora
- XPS śledzi zmiany w plikach C/C++ i rekompiluje je w miarę potrzeby
- XPS może uruchomić Platform Studio Software Development Kit (SDK)

Modyfikacja sprzętu

- Dodajemy moduły, edytujemy ich parametry i podłączamy do nich porty i magistrale z użyciem System Assembly view

- Wybieramy tab IP Catalog aby dodać peryferia

1

- Wybieramy moduł i przeciągamy go na okno System view lub podwójnie klikamy na nim aby dodać go do systemu

- W System View wybieramy moduł, klikamy prawym klawiszem i wybieramy Delete Instance

2

- Zmieniamy parametry z użyciem odpowiednich filtrów i wybieramy

3

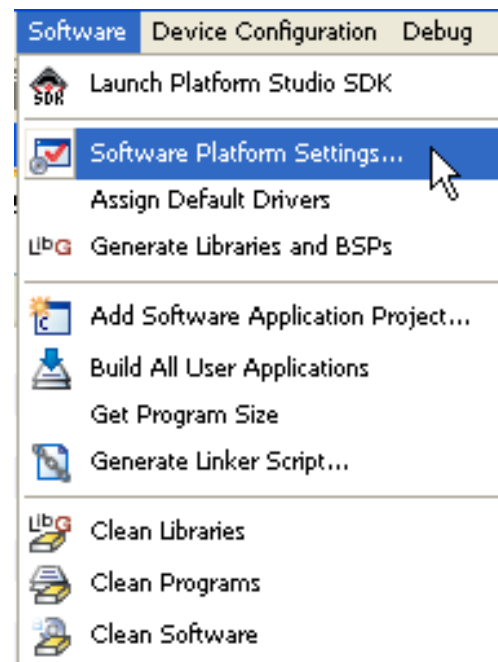
- adresy początkowe i końcowe
- parametry
- porty

The screenshot displays the Xilinx IDE interface. On the left, the 'IP Catalog' window shows a tree view of IP blocks, with 'MicroBlaze' (version 7.10.b) selected. In the center, the 'System Assembly view' shows a schematic diagram with components labeled 'P', 'L', 'M', 'B' and their respective bus connections. On the right, the 'IP Configuration' table lists various IP blocks and their versions. A context menu is open over the 'microblaze_0' instance, showing options like 'Delete Instance...'. The table below is a representation of the IP Configuration window.

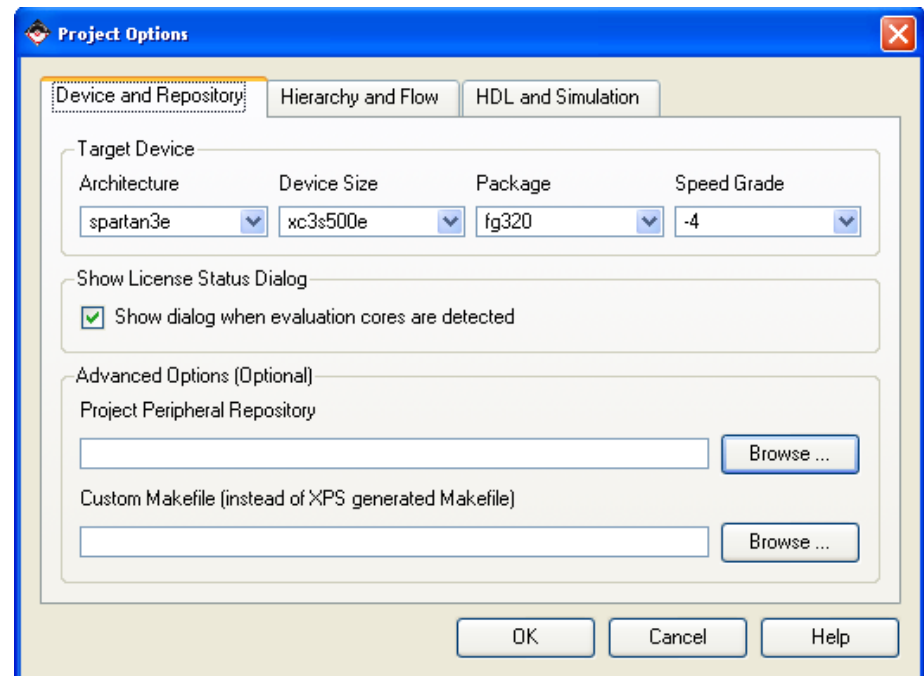
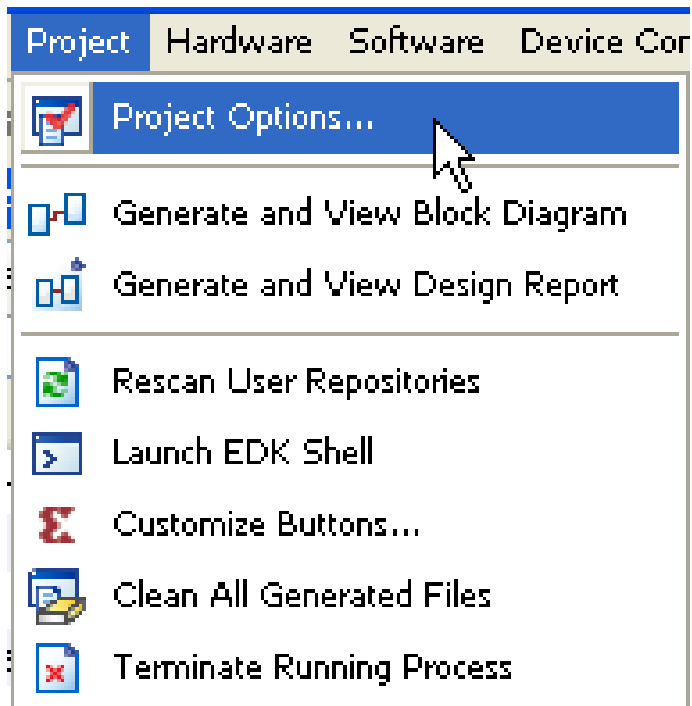
Name	Bus Connection	IP Type	IP Version
microblaze_0		microblaze	7.10.b
dmb		lmb_v10	1.00.a
ilmb		lmb_v10	1.00.a
lmb_plb		plb_v46	1.02.a
dmb_cntlr		lmb_bram_if_cntlr	2.10.a
ilmb_cntlr		lmb_bram_if_cntlr	2.10.a
DDR_SDRAM		mPMC	4.01.a
lmb_bram		bram_block	1.00.a
debug_module		mdm	1.00.b
LEDg_8Bif		xps_gpio	1.00.a
AS232_DCE		xps_uartlite	1.00.a
clock_generator_0		clock_generator	2.01.a
proc_sys_reset_0			

Edycja parametrów oprogramowania










- Ustawia wszystkie opcje związane z platformą programową w projekcie
 - Formularze umożliwiają ustawienie parametrów:
 - Software Platform
 - Sterownik CPU
 - SO i wersja SO
 - Wybór bibliotek
 - OS and Libraries
 - Okreslenie standardowych urządzeń wejścia i wyjścia
 - Konfiguracja SO i bibliotek
 - Drivers
 - Wybór sterowników i ich wersji



Ustawienie opcji projektu



Struktura katalogów projektu

-  project_directory
 -  data - created by default upon project creation
 -  etc - created by default upon project creation
 -  pcores - created by default upon project creation
 -  __xps - created by default upon project creation
 -  ppc405_0/microblaze_0 - created by default upon project creation
 -  implementation - created during PlatGen
 -  synthesis - created during PlatGen
 -  hdl - created during PlatGen



Struktura katalogów projektu

- ppc405_0 / microblaze_0
 - include
 - **.h header files*
 - libsrc
 - *BSP, drivers, etc.*
 - lib
 - *libc.a file*
 - *libm.a file*
 - *libxil.a file*
 - *boot.o file*
 - code (default repository if user application is not defined)
 - *executable.elf file*



Struktura katalogów projektu

- pcores
 - Peripheral IP files
- data
 - *system.ucf file*
- code
 - user-defined software application
 - **.c / *.cpp, *.h files*
- etc
 - *download.cmd file*
 - *fast_runtime.opt file*
 - *BSDL files*



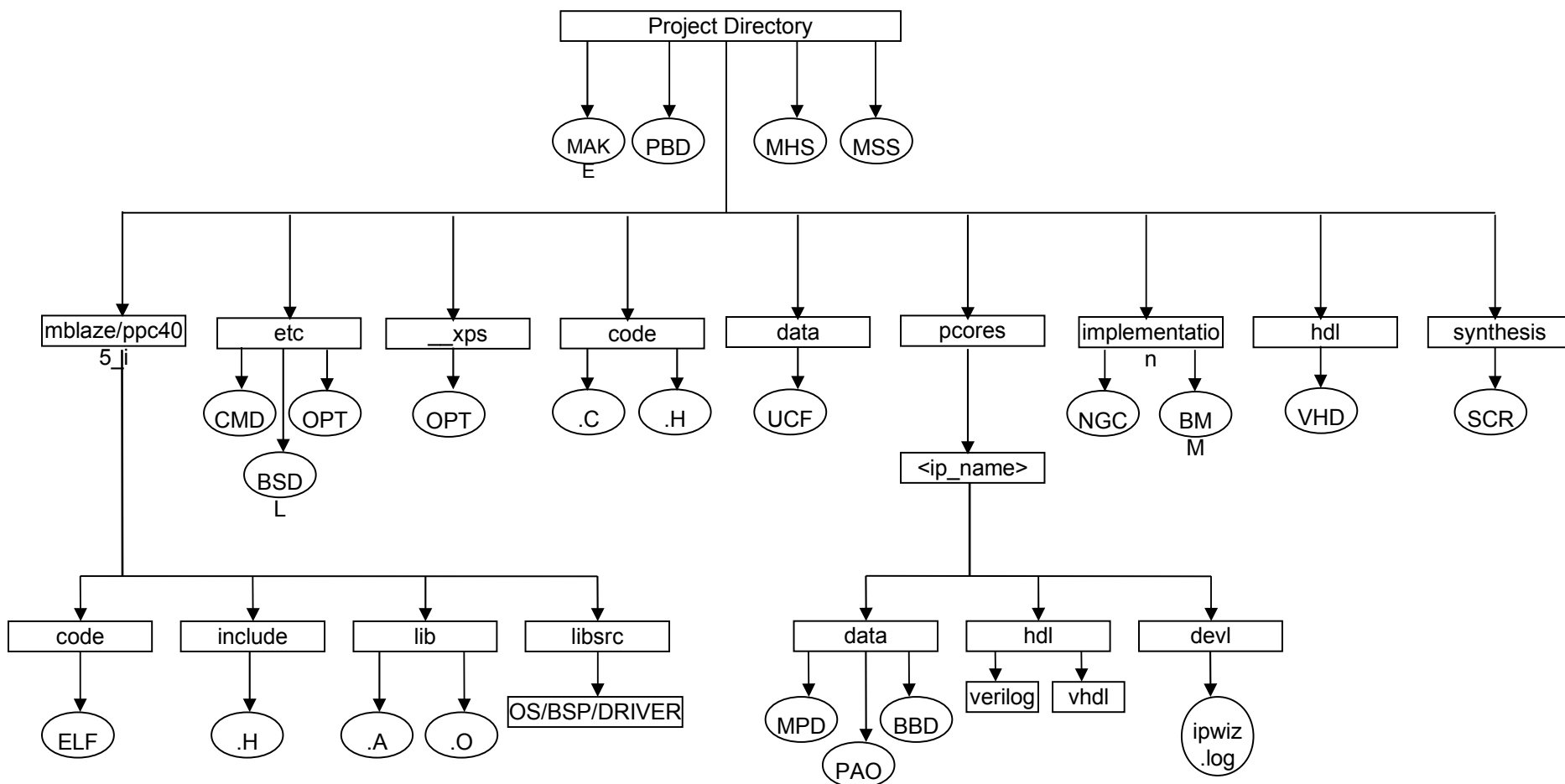


Struktura katalogów projektu

- `__xps`
 - *makefile*
- `synthesis`
 - *system.scr file*
- `HDL`
 - *system.[vhd|v], system_stub.[vhd|v]* file*
 - *Peripheral_wrapper.[vhd|v] files*
- `implementation`
 - *peripheral_wrapper.ngc files*
 - *system.ngc, system_stub.ngc* file*
 - *system.bmm file*

* files created if system is a submodule

Struktura katalogów projektu





KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



„Układy reprogramowalne i SoC”
„Projektowanie systemów wbudowanych z użyciem EDK”

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

