Elektronické systémy na báze obvodov FPGA

2004/2005

Pavol Galajda, KEMT, FEI, TUKE
Pavol.Galajda@tuke.sk

1 Úvod do ASIC- teoretický základ

- 1.1 Základné pojmy
- 1.2 Historický vývoj a rozdelenie IO
- 1.3 Typy PLD obvodov
  - SPLD
  - CPLD
  - FPGA
- 1.4 Ekonomické aspekty
  - Porovnanie ASIC technológií
- 1.5 ASIC verzus FPGA – migrácia, prechod ku ASIC

2 Metodika návrhu PLD

- 2.1 Činnosť pred započatím návrhu
- 2.2 Rozdelenie CAD nástrojov
- 2.3 Modely pre metódy návrhu systémov
  - metóda „vodopád“
  - metóda „špirála“
- 2.4 Etapy návrhu číslicových systémov s obvodmi FPD

3 Klasifikácia PLD z hľadiska technológie výroby

- 3.1 FUSE
- 3.2 EPROM a EEPROM
- 3.3 SRAM
- 3.4 ANTIFUSE
- 3.5 FLASH

4 Architektúry a typy číslicových obvodov SPLD

- 4.1 Úvod do PLD (Programmable Logic Device)
- 4.2 Obvody PLA (Programmable Logic Array)
- 4.3 Obvody PAL (Programmable Array Logic)
- 4.4 Obvody GAL (Generic Array Logic)

5 Architektúry a typy číslicových obvodov CPLD

- 5.1 Lattice pLSI a ispLSI
- 5.2 MAX 7000 CPLD (Multiple Array matriX, Altera)
- 5.3 Xilinx XC 7000

6 Architektúry a typy číslicových obvodov FPGA

- 6.1 Xilinx XC 4000
- 6.2 Altera FLEX 10K
- 6.3 Altera Cyclone

7 Vývojové prostriedky obvodov FPGA

- 7.1 Altera- Quartus II
- 7.2 Xilinx- ISE WEB Pack
- 7.3 Mentor Graphic- FPGA Advantage

7.1 Altera- Quartus II

Quartus II je vývojový systém pre návrh číslicových systémo
s programovateľnými logickými obvodmi od firmy ALTERA.

Je to systém, ktorý umožňuje vytvoriť návrh rôznymi spôsobmi:

- schematicky,
- vytváraním zdrojového kódu, …

pričom jednotlivé spôsoby môžeme kombinovať v jednom návrhu (č
umožňuje hierarchické usporiadanie návrhu**).**

7.1 Quartus II- Postup pri návrhu

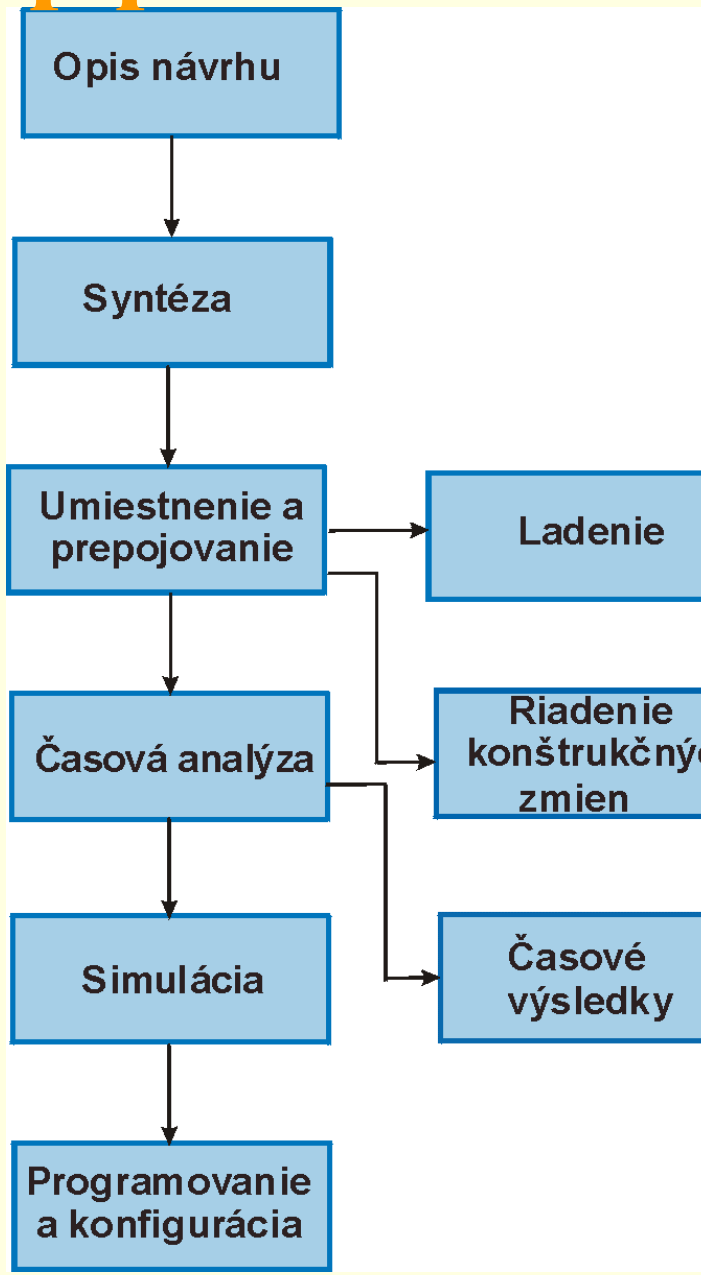
Quartus II:

- Poskytuje kompletné návrhové prostredie, ktoré sa ľahko prispôsobuje potrebám návrhu užívateľa,
- poskytuje riešenia pre všetky fázy návrhu FPGA a CPLD.

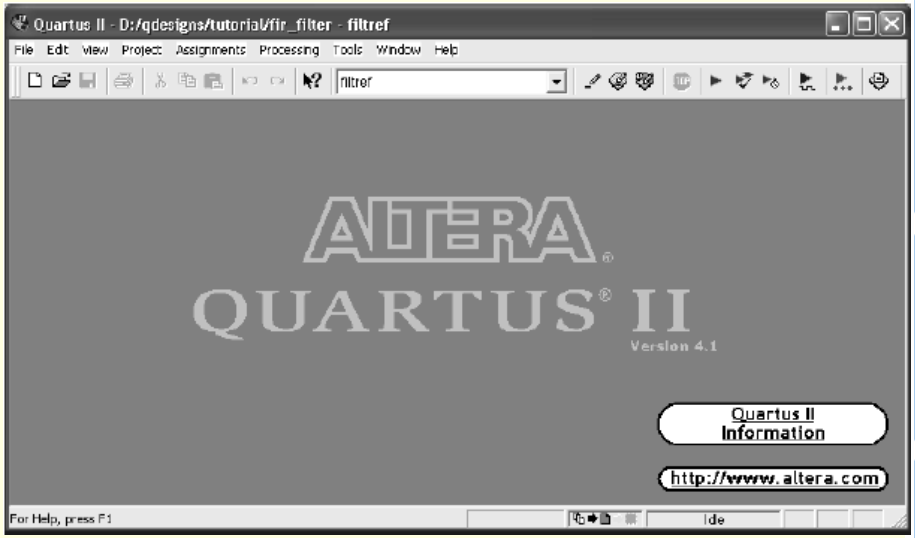
Dovoľuje používať:

- grafické návrhové prostriedky,
- EDA nástroje,
- príkazový riadok,

pre celý návrh, alebo je možné používať rôzne nastavenia pre jednotlivé fázy návrhu.



7.1 Quartus II- Grafické prostriedky



Obr. ukazuje funkcie, ktoré Quartus II grafický návrhový prostriedok poskytuje v jednotlivých fázach návrhu.

Design Entry
- Text Editor
- Block & Symbol Editor
- MegaWizard Plug-In Manager
- Assignment Editor
- Floorplan Editor

Synthesis
- Analysis & Synthesis
- VHDL, Verilog HDL & AHDL
- Design Assistant
- RTL Viewer
- Technology Map Viewer

Place & Route
- Fitter
- Assignment Editor
- Floorplan Editor
- Chip Editor
- Report Window
- Incremental Fitting
- Resource Optimization Advisor

Timing Analysis
- Timing Analyzer
- Report Window
- Technology Map Viewer

Simulation
- Simulator
- Waveform Editor

Programming
- Assembler
- Programmer
- Convert Programming Files

System-Level Design
- SOPC Builder
- DSP Builder

Software Development
- Software Builder

Block-Based Design
- LogicLock Window
- Floorplan Editor
- VQM Writer

EDA Interface
- EDA Netlist Writer

Timing Closure
- Floorplan Editor
- LogicLock Window
- Timing Optimization Advisor

Debugging
- SignalTap II
- SignalProbe
- In-System Memory Content Edito
- RTL Viewer
- Technology Map Viewer
- Chip Editor

Engineering Change Management
- Chip Editor
- Resource Property Editor
- Change Manager

7.1 Quartus II- EDA nástroje

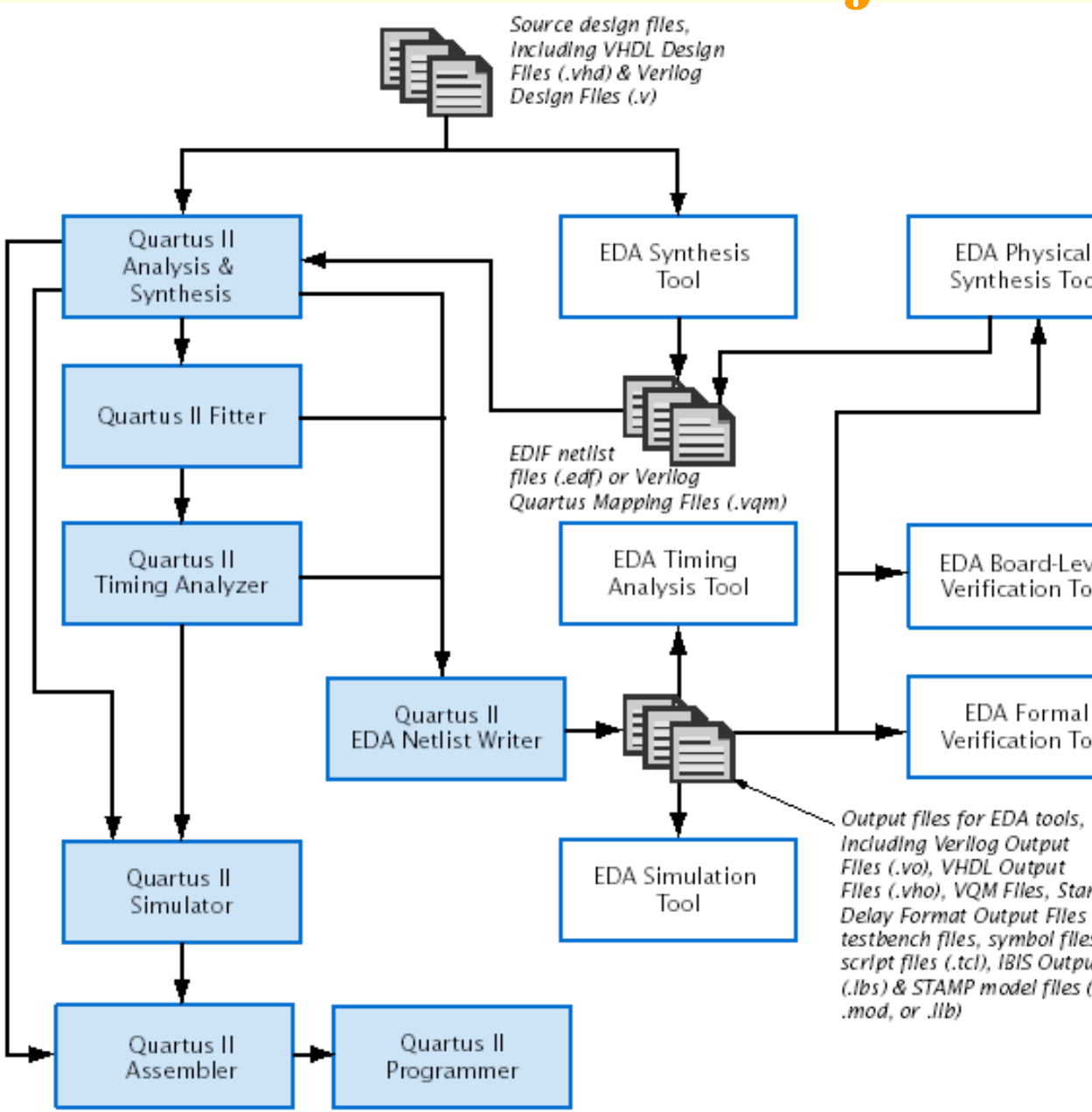
Quartus II dovoľuje použiť EDA prostriedky pre rôzne fázy návrhu.
Syntéza:

- MG LeonardoSpectrum
- MG Precisin

Simulácia:

- MG ModelSim

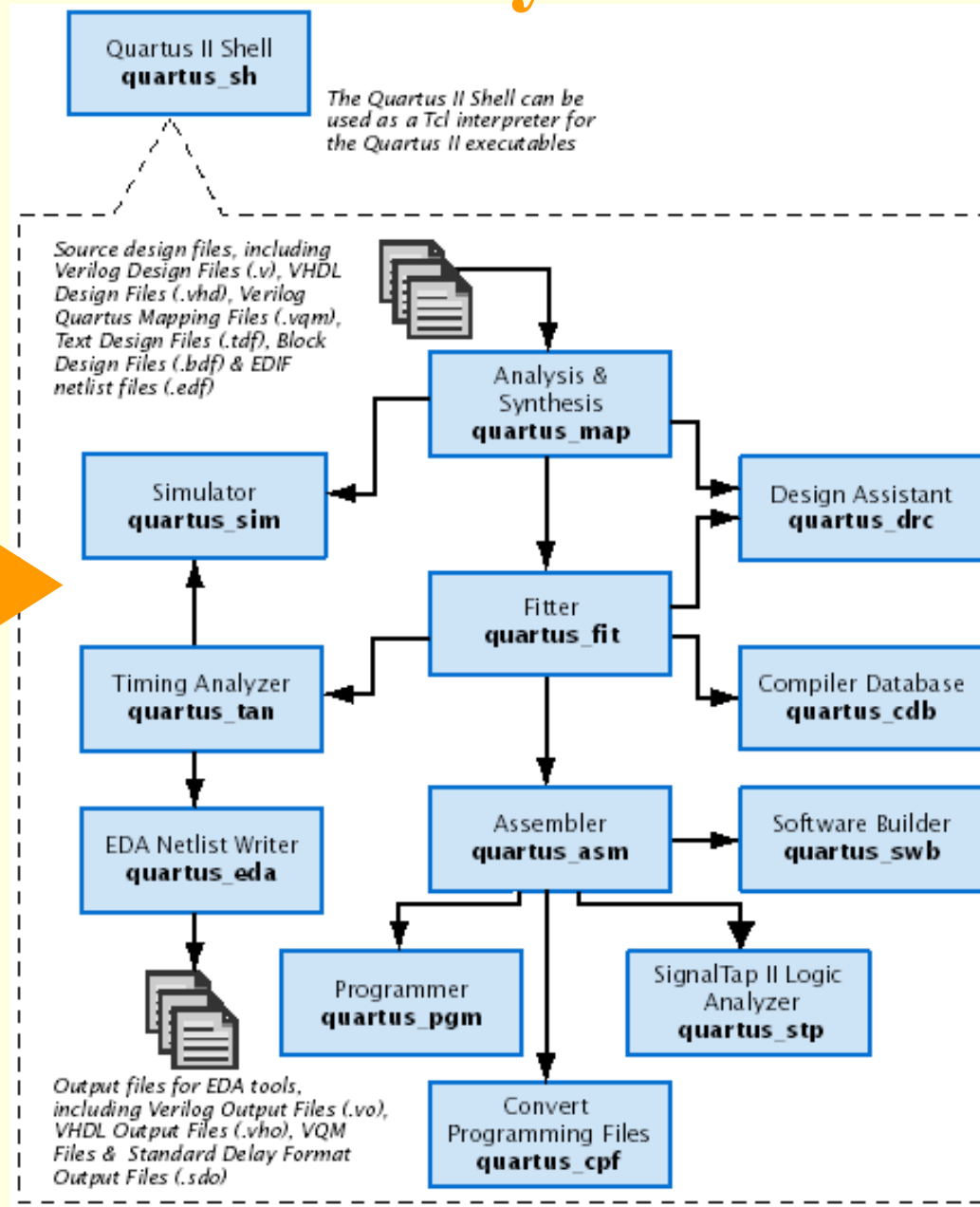
Obr. ukazuje postup pri návrhu použitím EDA.



7.1 Quartus II- Príkazový riadok

Quartus II
dovoľuje použiť v rôznych fázach návrhu aj príkazový riadok.

Obr. ukazuje postup pri návrhu použitím príkazov v príkazovom riadku.

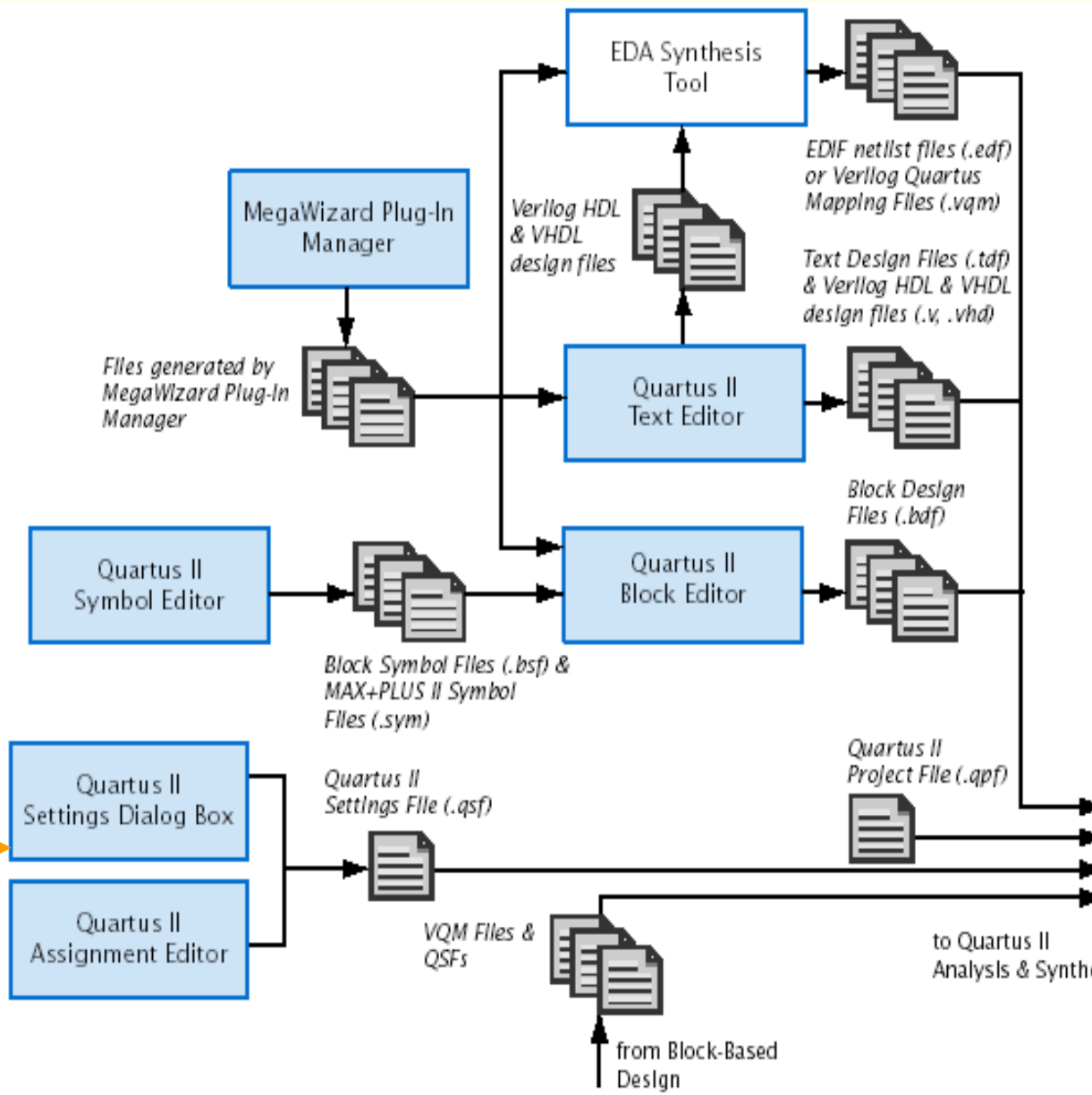


7.1 Quartus II- Opis návrhu

Quartus II projekt obsahuje rôzne návrhové súbory, zdrojové súbory a iné sprievodné súbory potrebné pre správnu činnosť návrhu.

+ megafunkcie
+ LPM funkcie
+ IP funkcie

Obr. ukazuje opis návrhu



7.1 Quartus II- Opis návrhu

Vytvorenie projektu

Nový projekt je možné vytvoriť použitím príkazu **New Projekt Wizard** z **File menu** alebo pomocou príkazu **quartus_map.**

Príkazom **New Projekt Wizard** sa špecifikuje pracovný adresár projektu, meno projektu a určí sa meno najvyššej úrovne navrhovaného objektu. Okrem toho môžeme špecifikovať, ktoré návrhové súbory, zdrojové súbory, užívateľské knižnice a EDA prostriedky chceme používať v projekte ako aj špecifikovať použitý obvod.



7.1 Quartus II- Opis návrhu

Vytvorenie návrhu

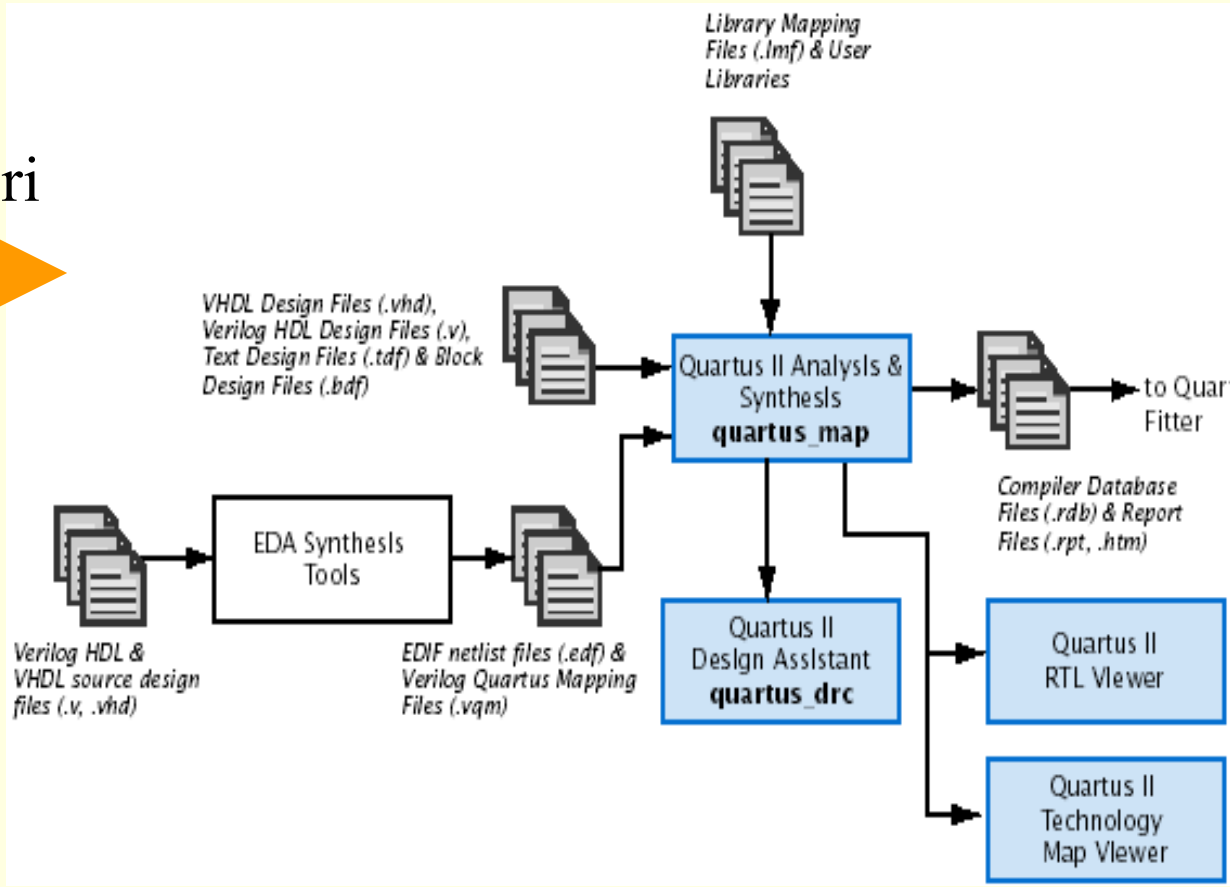
Tabuľka ukazuje rôzne typy súborov, ktoré môžu byť použité pri vytváraní návrhov v Quartus II alebo v EDA prostriedkoch.

Typ súboru	Popis	Prípona
Block Design File	Schematický súbor vytvorený s *Quartus II Block Editor*	.bdf
EDIF Input File	EDIF netlist- ovský súbor,	.edf .edif
Graphic Design File	Schematický súbor, vytvorený s *MAX+PLUS II Graphic Editor*	.gdf
Text Design File	AHDL (Altera HDL) súbor	.tdf
Verilog Design File	Súbor, ktorý obsahuje navrhovanú logiku definovanú s *Verilong HDL*	.v .vlg .verilong
VHDL Design File	Súbor, ktorý obsahuje navrhovanú logiku definovanú s *VHDL*	.vh .vhd .vhdl

7.1 Quartus II- Analýza a syntéza

Analýzu a Syntézu *(Analysis & Synthesis)* môžeme využívať na analýzu návrhu a na vytváranie databázy projektu. Syntéza používa Quartus II Integrated Synthesis na syntézu návrhu vo VHDL (**.vhd)** alebo Verilog (**.v**.) Ak na syntézu návrhu vo VHDL alebo Verilong použijeme iné EDA prostriedky, sa generuje EDIF Netlist File (**.edf)** alebo Verilog Quartus Maping File (**.vqm**), ktoré môžu byť potom použité v Quartus II.

Obr. ukazuje postup pri syntéze.



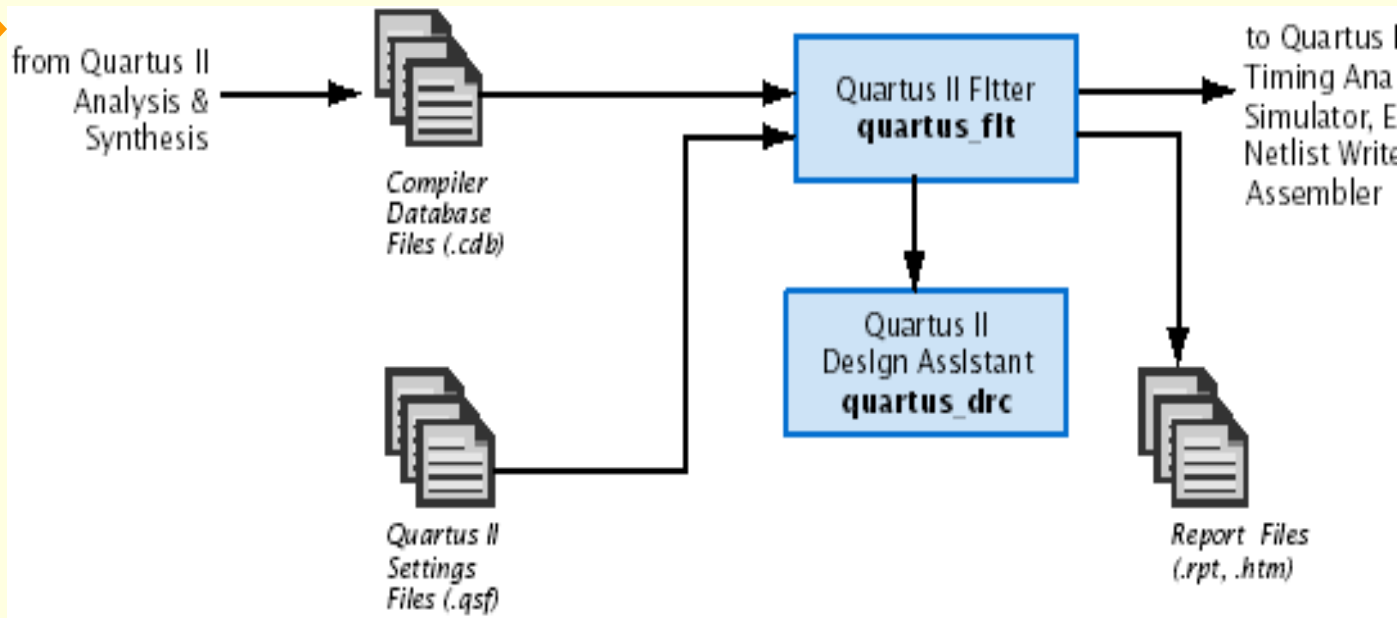
7.1 Quartus II- Analýza a syntéza

Pri použití iných EDA nástrojov na syntézu môže užívatel špecifikovať aj súbor mapovania knižnice *(Library Mapping File* – **.lmf)**, ktorý sa bude využívať na mapovanie funkcií, ktoré nie sú funkciami Quartus II. Tieto ale aj iné nastavenia je môžeme definovat vo **Verilong Input** a **VHDL Input** v dialógovom okne **Settings.**

Analýza a syntéza používa niekoľko algoritmov na minimalizáciu počtu hradiel, odstránenie nadbytočnej logiky a efektívne využíva architektúru obvodu.

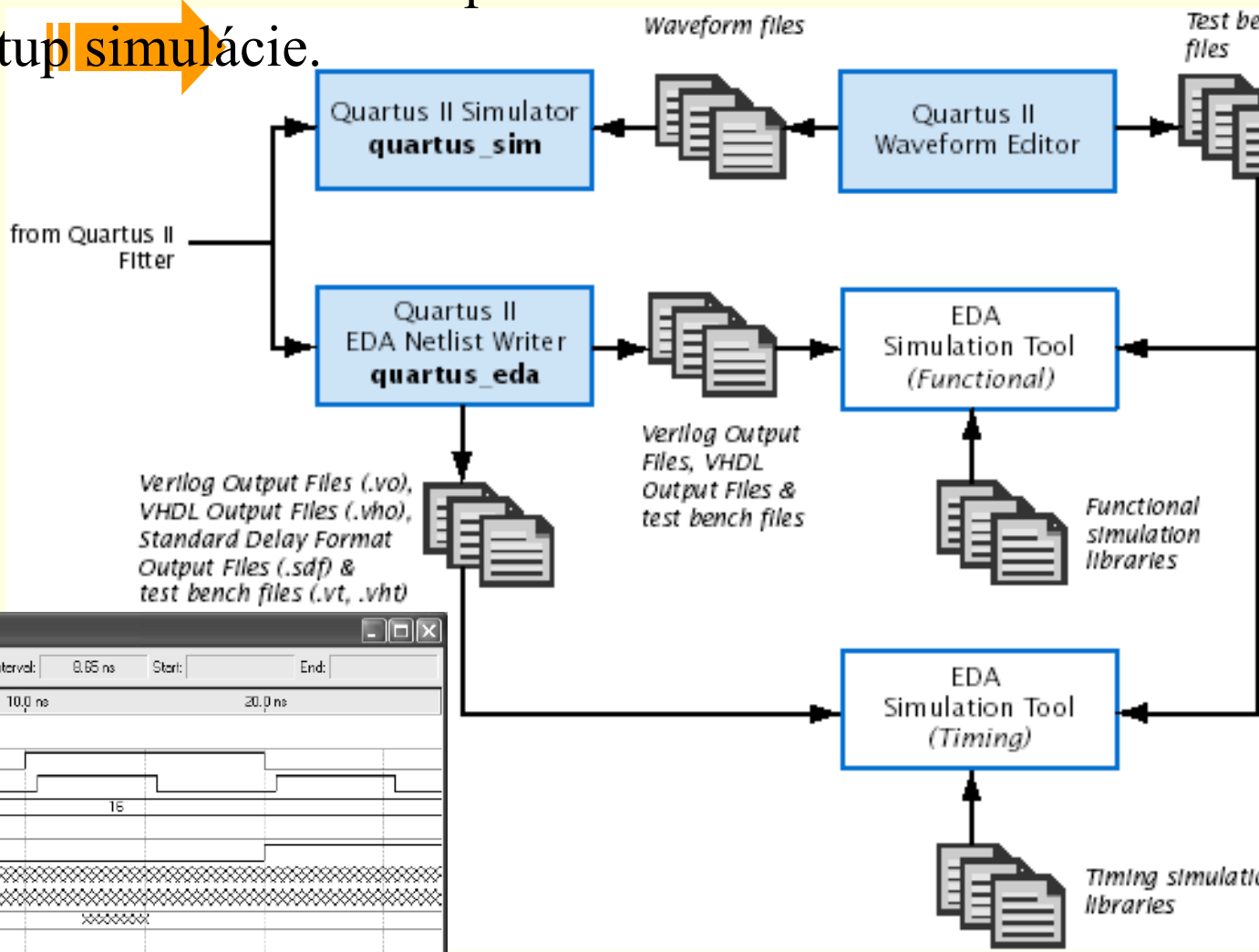
7.1 Quartus II- Umiestnenie a prepojovanie

Quartus II *Fitter*, realizuje umiestnenie a prepojovanie- navrhovanej logickej funkcie. Využíva sa databáza, ktorá bola generovaná pri analýze a syntéze. Fitter porovnáva logické a časové požiadavky projektu s dostupnými zdrojmi obvodu. Každú logickú funkciu priradí k logickej bunke, ktorá má najlepšie umiestnenie z pohľadu prepojovania a oneskorenia a vyberie vhodnú prepojovaciu cestu a pin obvodu. Obr. ukazuje postup umiestnenia a prepojovania v návrhu.

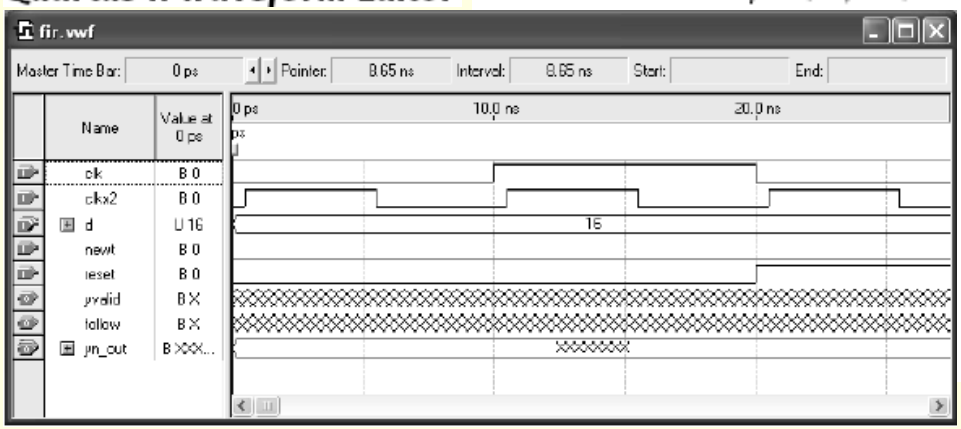


7.1 Quartus II- Simulácia

Funkčnú a časovú simuláciu návrhu môžeme vykonať použitím Quartus II simulátora alebo použitím EDA simulátora. Obr ukazuje postup simulácie.



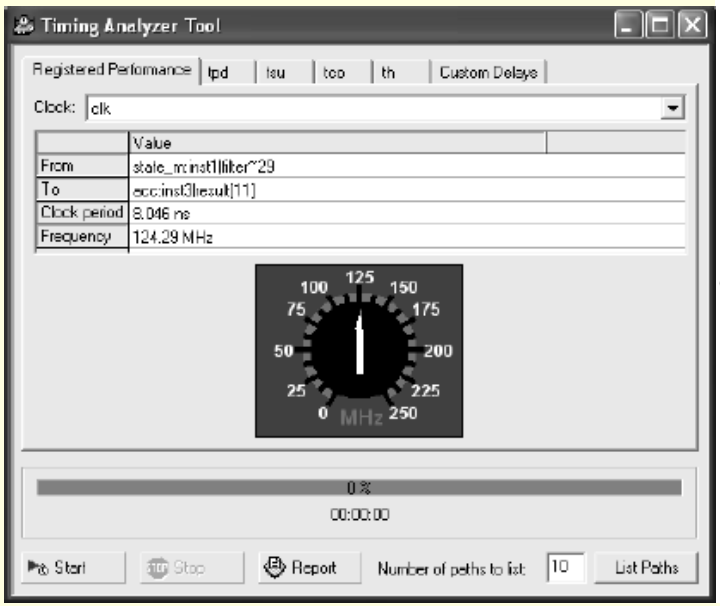
Quartus II Waveform Editor

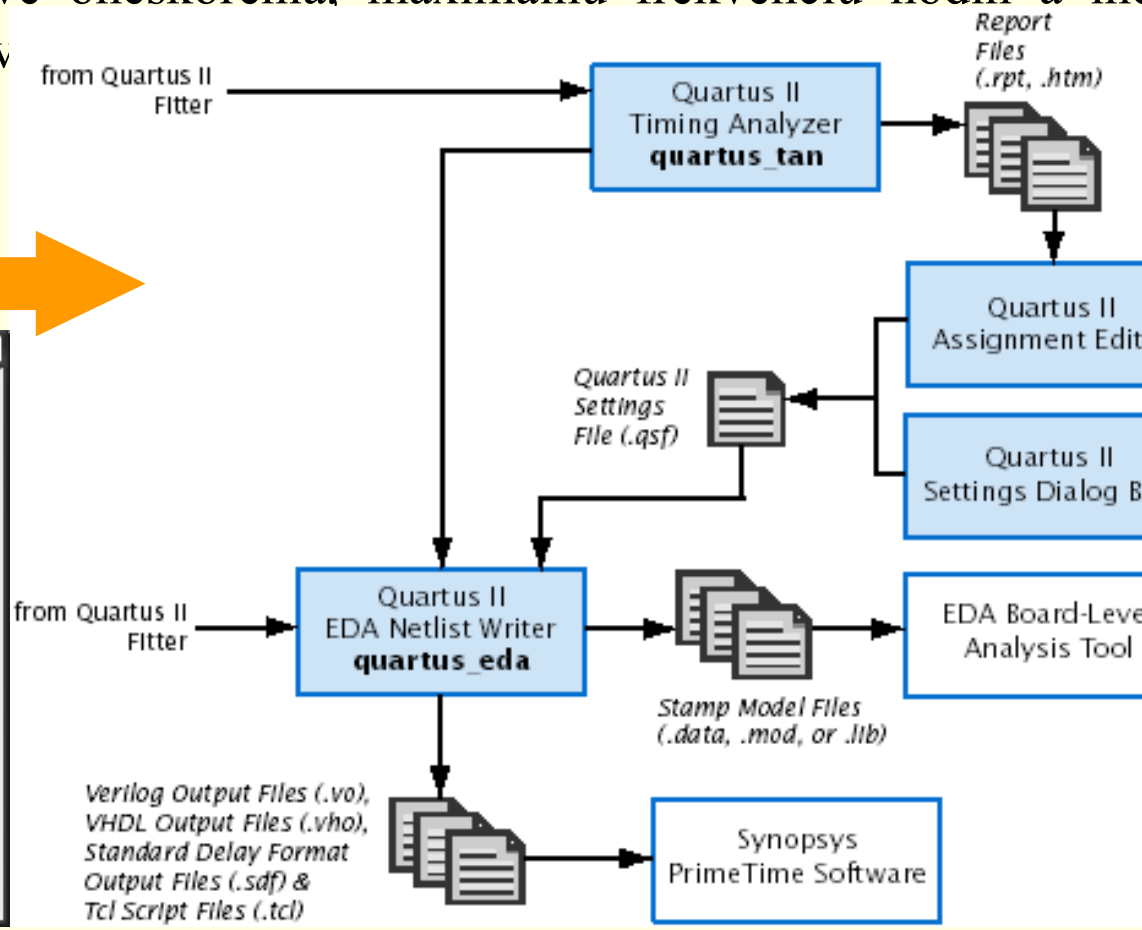


7.1 Quartus II- Časová analýza

Quartus II Timing Analyzer umožňuje analyzovať charakteristiky realizovanej logiky a napomáha pri fyzickom usporiadaní návrhu (*fitting*) tak, aby návrh vyhovoval časovým požiadavkám. Štandardne sa Timing Analyzer spúšťa automaticky– je súčasťou kompilácie pričom analyzuje a zaznamenáva rôzne časové informácie, časové oneskorenia, maximálnu frekvenciu hodín a iné časové charakteristiky náv

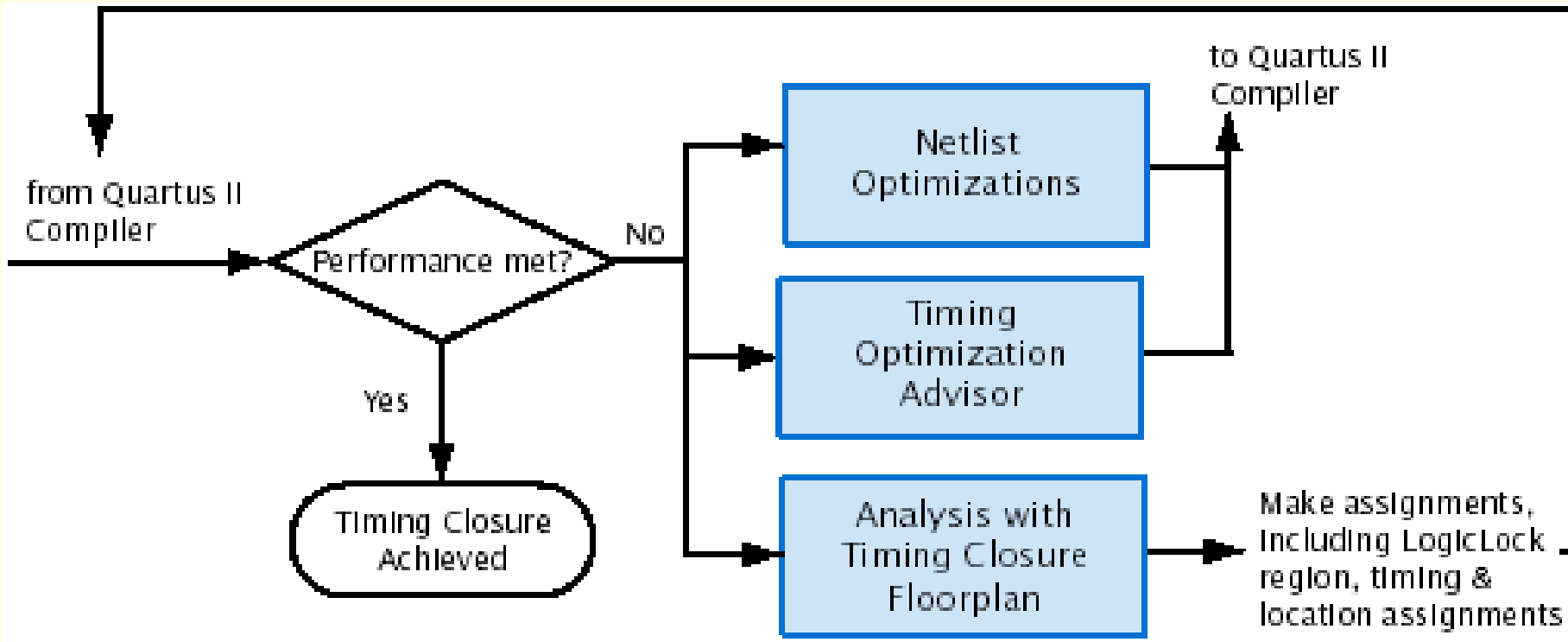
Obr. ukazuje postup pri časovej analýze





7.1 Quartus II- Časové spínanie

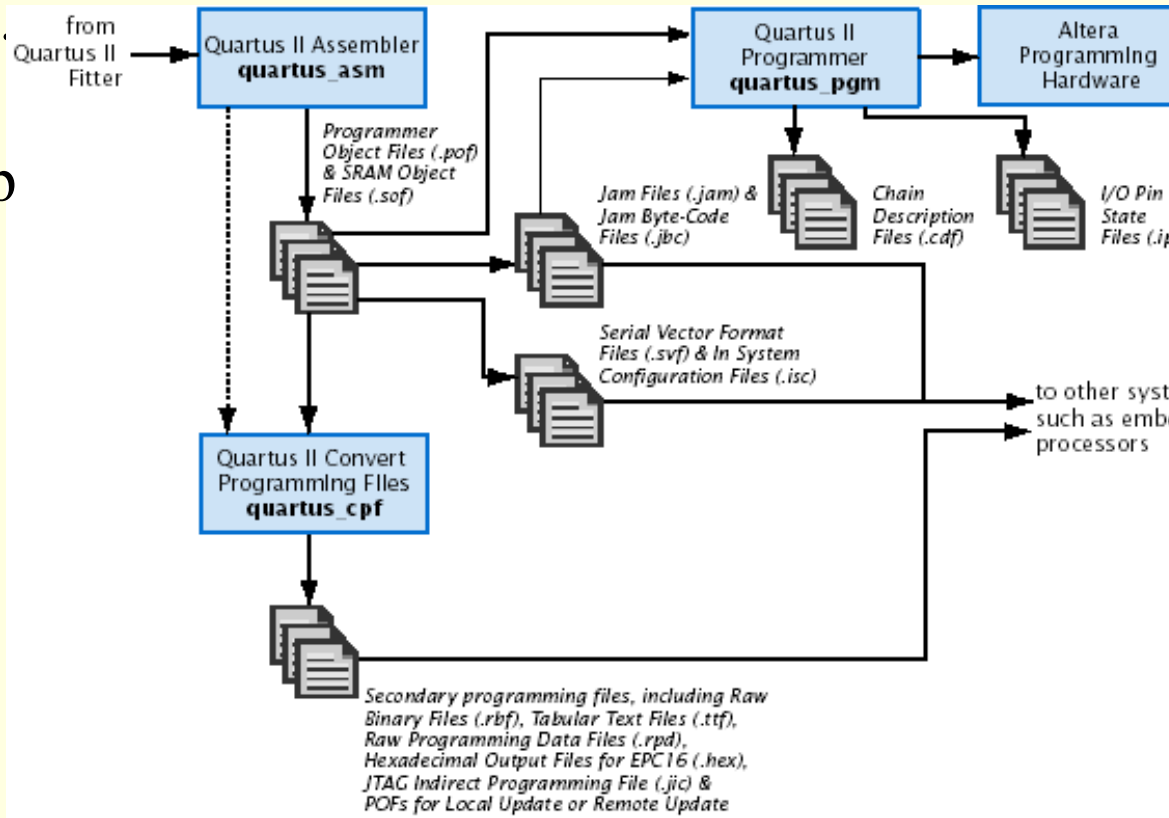
Quartus II Timing Closure umožňuje uskutočniť prvotnú kompiláciu, zobrazit výsledky návrhu a následne uskutočniť ďalší optimalizovaný návrh (riadením syntézy, umiestnenia a prepojovania návrhu), ktorý vyhovuje vyšpecifikovaným časovým požiadavkám. Výsledkom tohto procesu sú rýchlejšie realizácie zložitých návrhov a zníženie počtu optimalizačných iterácii.



7.1 Quartus II- Programovanie a konfigurácia

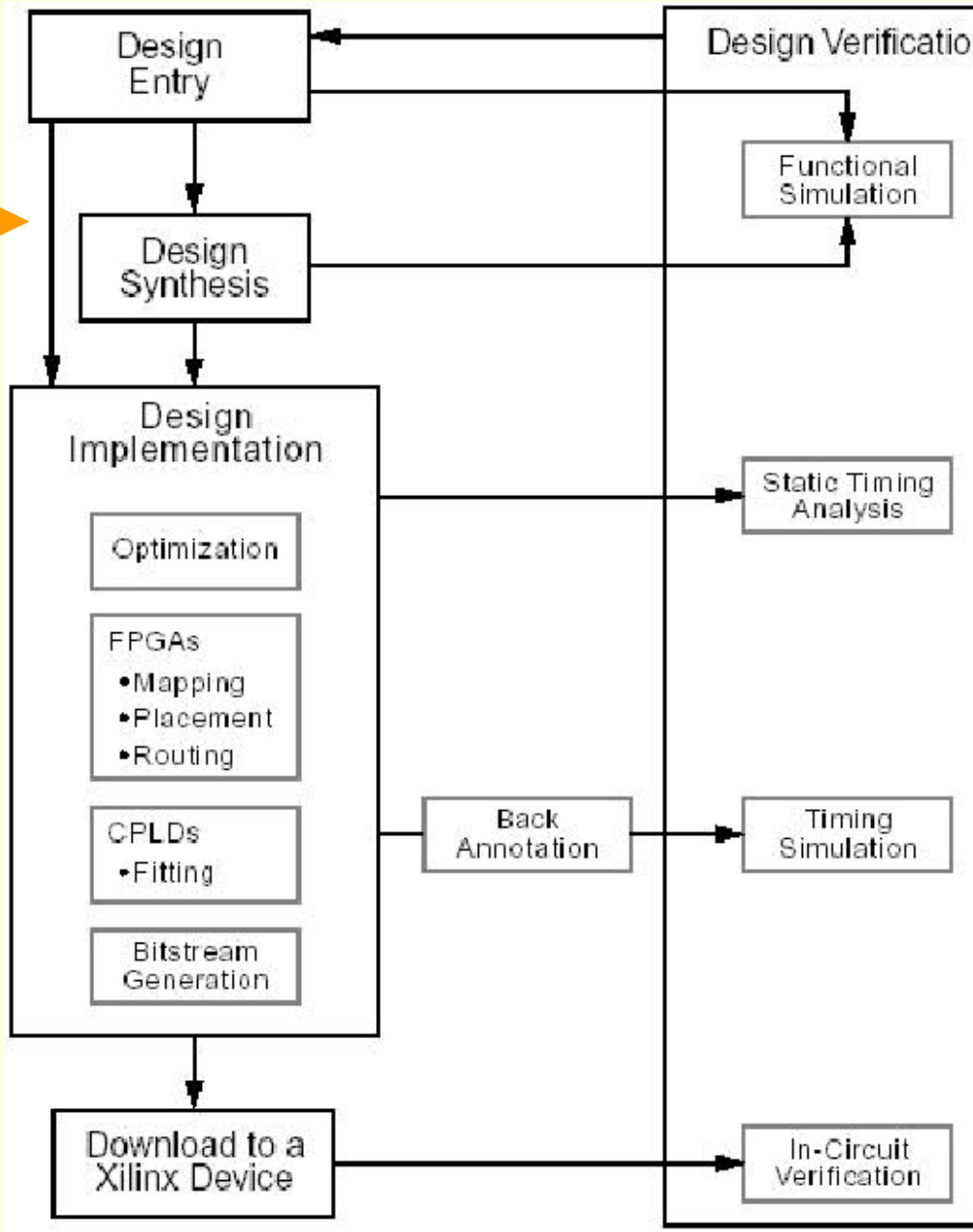
Po úspešnej kompilácii projektu je možné obvod Altera® programovať, alebo konfigurovať. Modul *Assembler* Quartus II kompilátora generuje súbory, potrebné na programovanie a Quartus II *Programmer* vie tieto súbory s programovateľným hardvérom od firmy Altera použiť na programovanie, alebo konfiguráciu obvodu.

Obr. ukazuje postup programovania súčiastky.



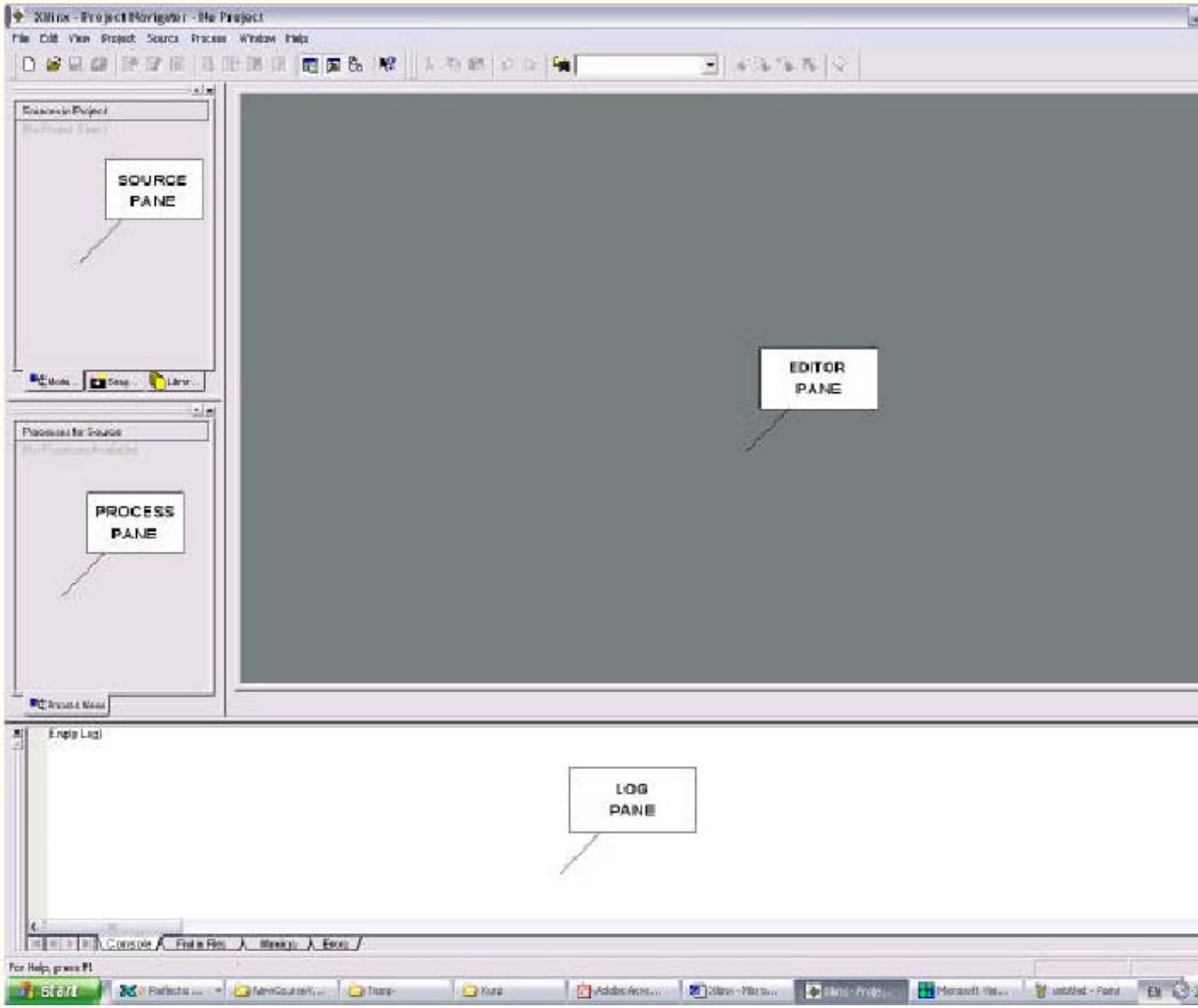
7.2 Xilinx- ISE WEB Pack

Postup pri návrhu s obvodmi Xilinx vo vývojovom prostredí ISE.



7.2 ISE WEB Pack- Project Navigator

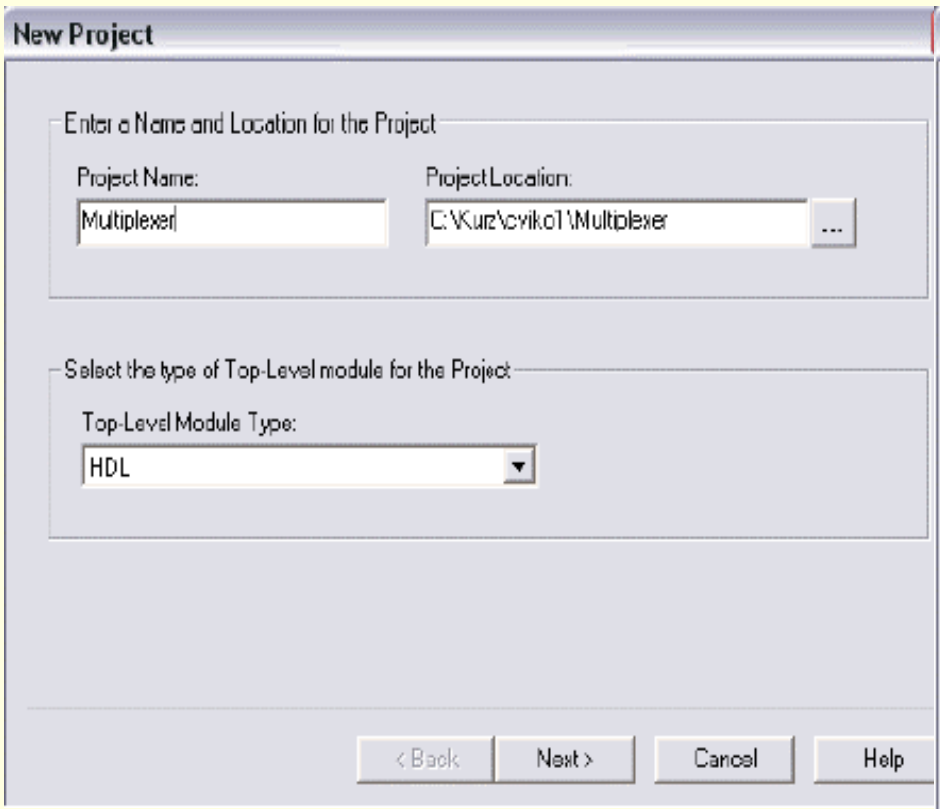
- Oblasť zdrojových súborov
- Oblasť procesov
- Oblasť záznamov
- Editovacia oblasť

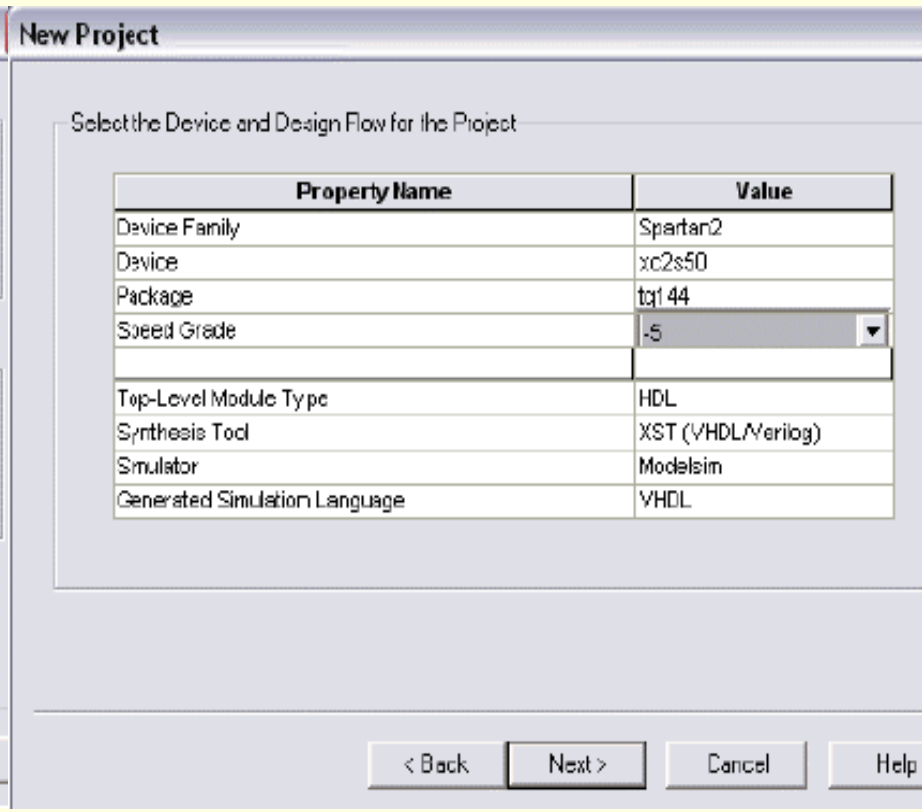


7.2 ISE WEB Pack- Vytvorenie projektu

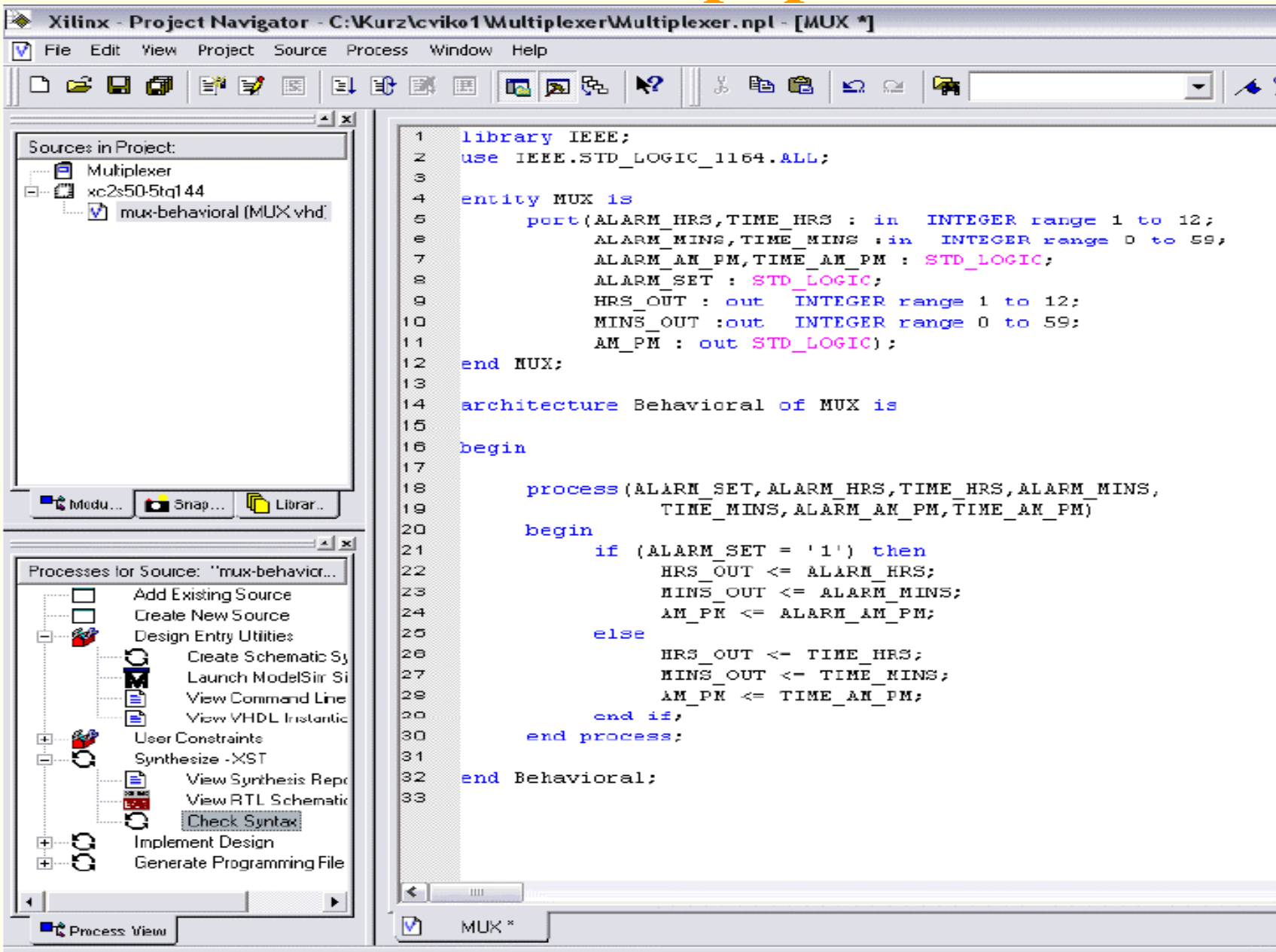
V okne New Project definujeme:

- umiestnenie súborov nášho projektu,
- meno projektu,
- použitý obvod,
- nástroje použité pri syntéze logiky zo zdrojového súboru.



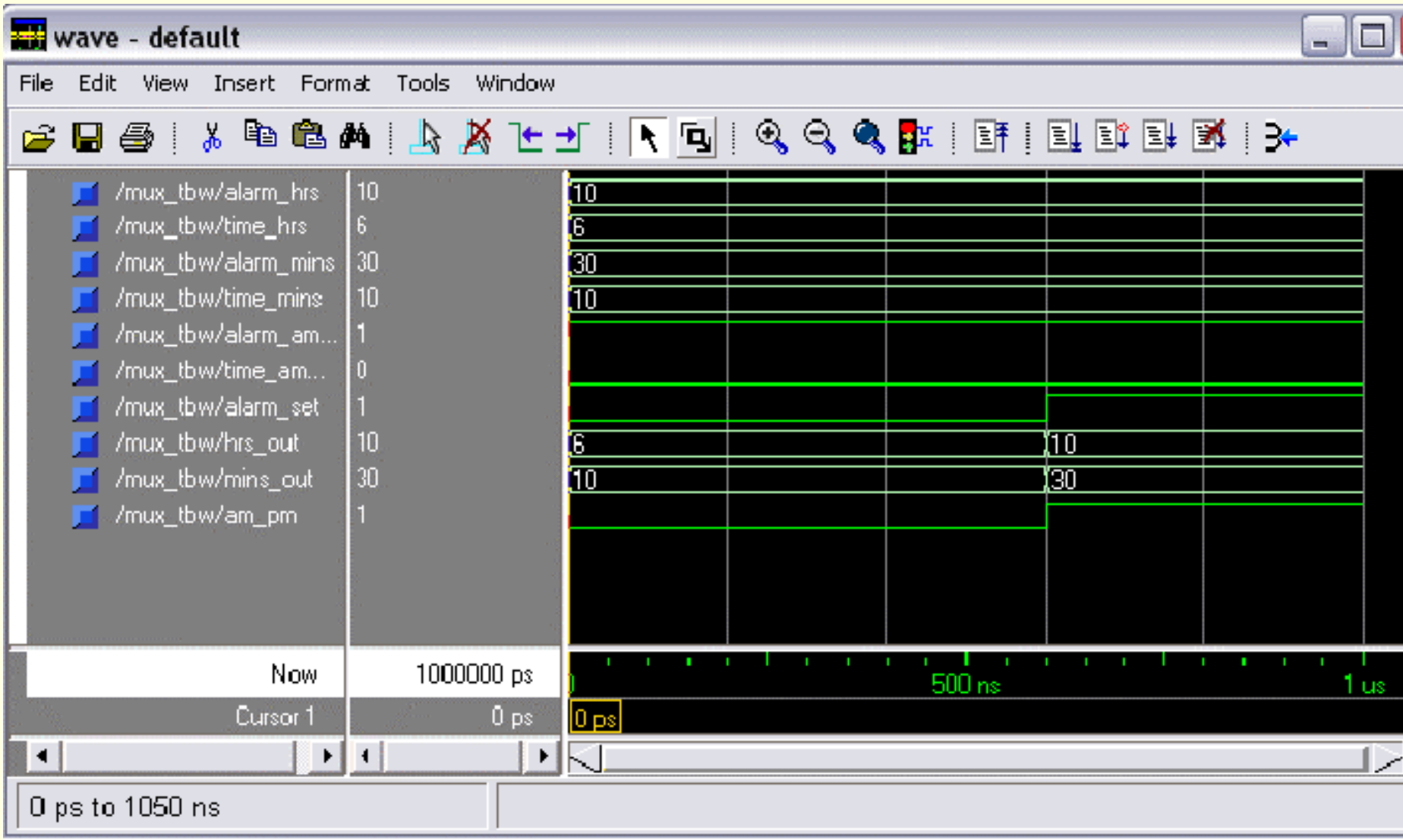


7.2 ISE WEB Pack- popis vo VHDL



7.2 ISE WEB Pack- Funkčná simulácia

Funkčná simulácia- sa vykonáva pred syntézou, za účelom overenia logického návrhu.



7.2 ISE WEB Pack- Ďalšie fázy návrhu

Syntéza VHDL kódu

Implementácia logických obvodov do FPGA

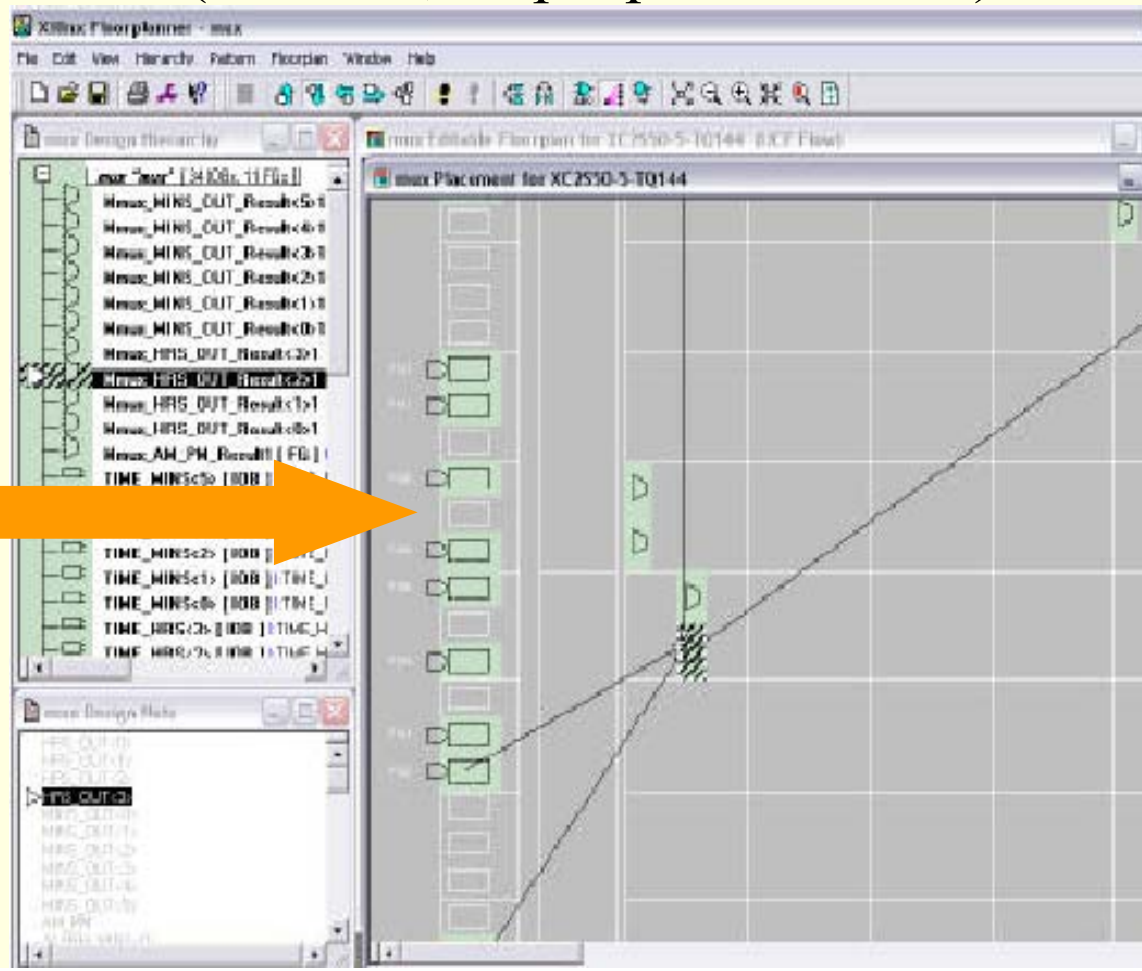
- umiestnenie a prepojovanie (translate, map a place & route)

Kontrola implementácie

Priradenie pinov

Zobrazenie čipu

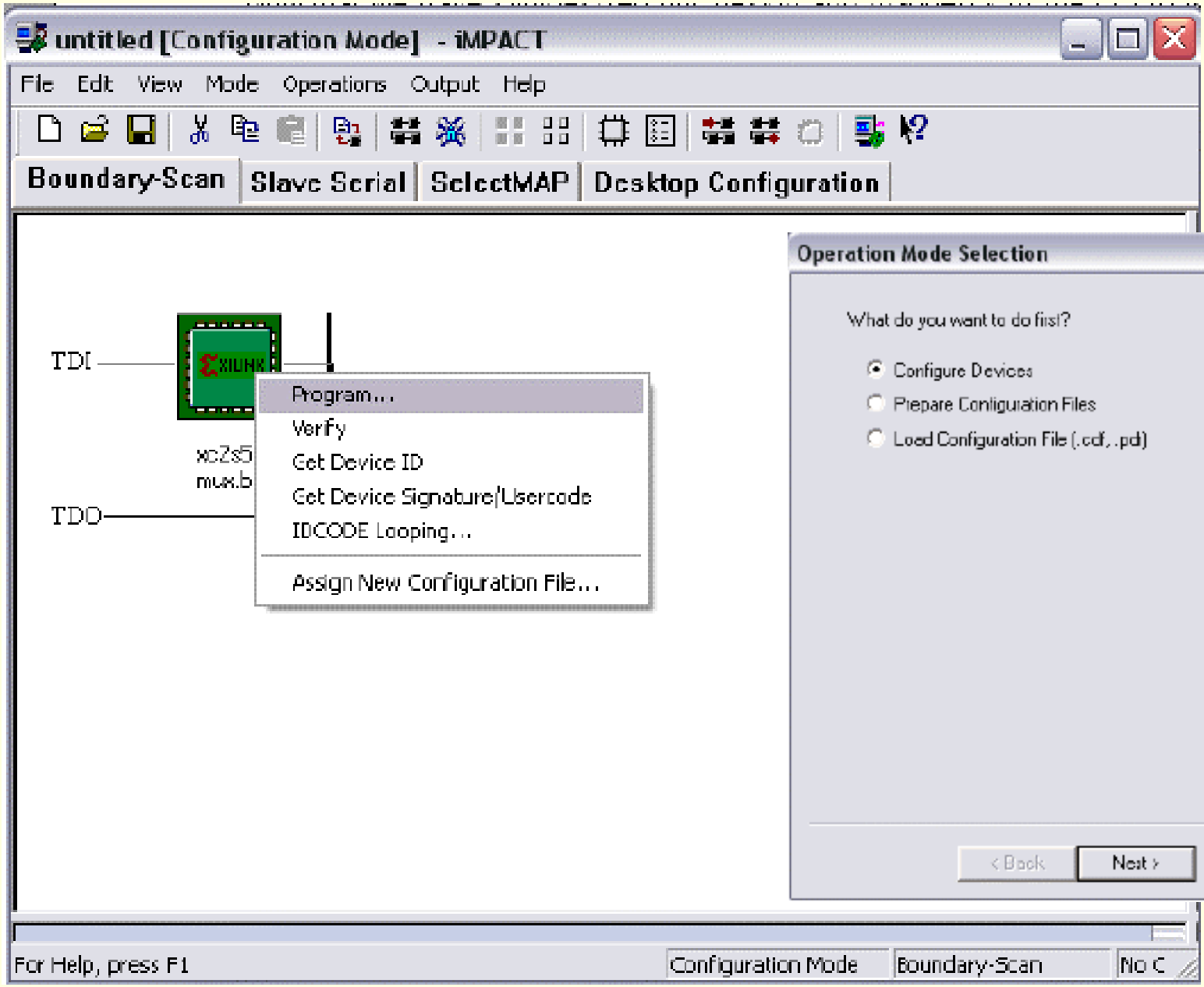
- Floor Planner



Časová simulácia

- VHDL Test Bench

7.2 ISE WEB Pack- Konfigurácia



Moderné postupy pri návrhu PLD

Zameriame sa na konkrétne etapy v procese návrhu a na spôsoby, ktorými je možné tieto etapy v dnešnej dobe efektívne riešiť.

Príkladom vhodného riešenia návrhového prostredia môže byť programový súbor ***FPGA Advantage*** firmy Mentor Graphics.

Moderné postupy pri návrhu PLD

Motivácia

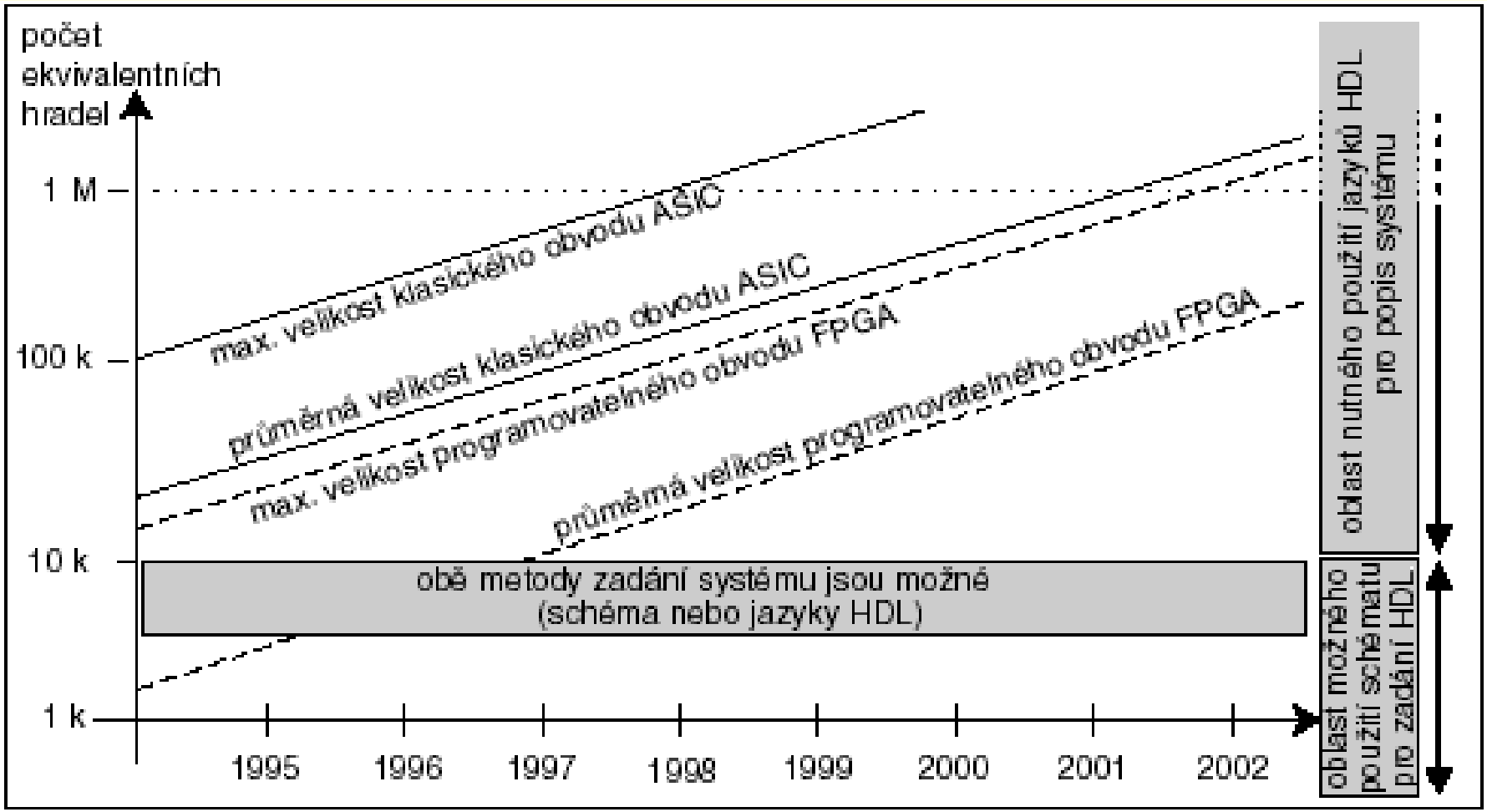
Dnes už existujú súčiastky FPGA obsahujúce až niekoľko miliónov hradiel.

Tradičné návrhy s využitím schém sa stávajú zložité (nezvládnuteľné) a časovo náročné. Zároveň sa zvyšuje tlak na maximálne skracovanie vývoja funkčného vzorku.

Dopyt po efektívnych metódach je oprávnený a je zrejme, že využitie jazyku HDL sa stáva nevyhnutným nielen v oblasti návrhu klasických zákazníckych obvodov ASIC, ale aj obvodov FPGA.

Moderné postupy pri návrhu PLD

Pri zložitosti viac než 10 tisíc ekvivalentných hradiel je pre popis číslicovýc systémov výhodnejšie použiť návrhové metódy, ktoré podporujú jazyky HDL. Tiet metódy zároveň zjednodušujú opakované používanie blokov, integráciu existujúcic návrhov do nových projektov a integráciu zakúpených makrofunkcií (tzv. IP funkcií)



Moderné postupy pri návrhu PLD

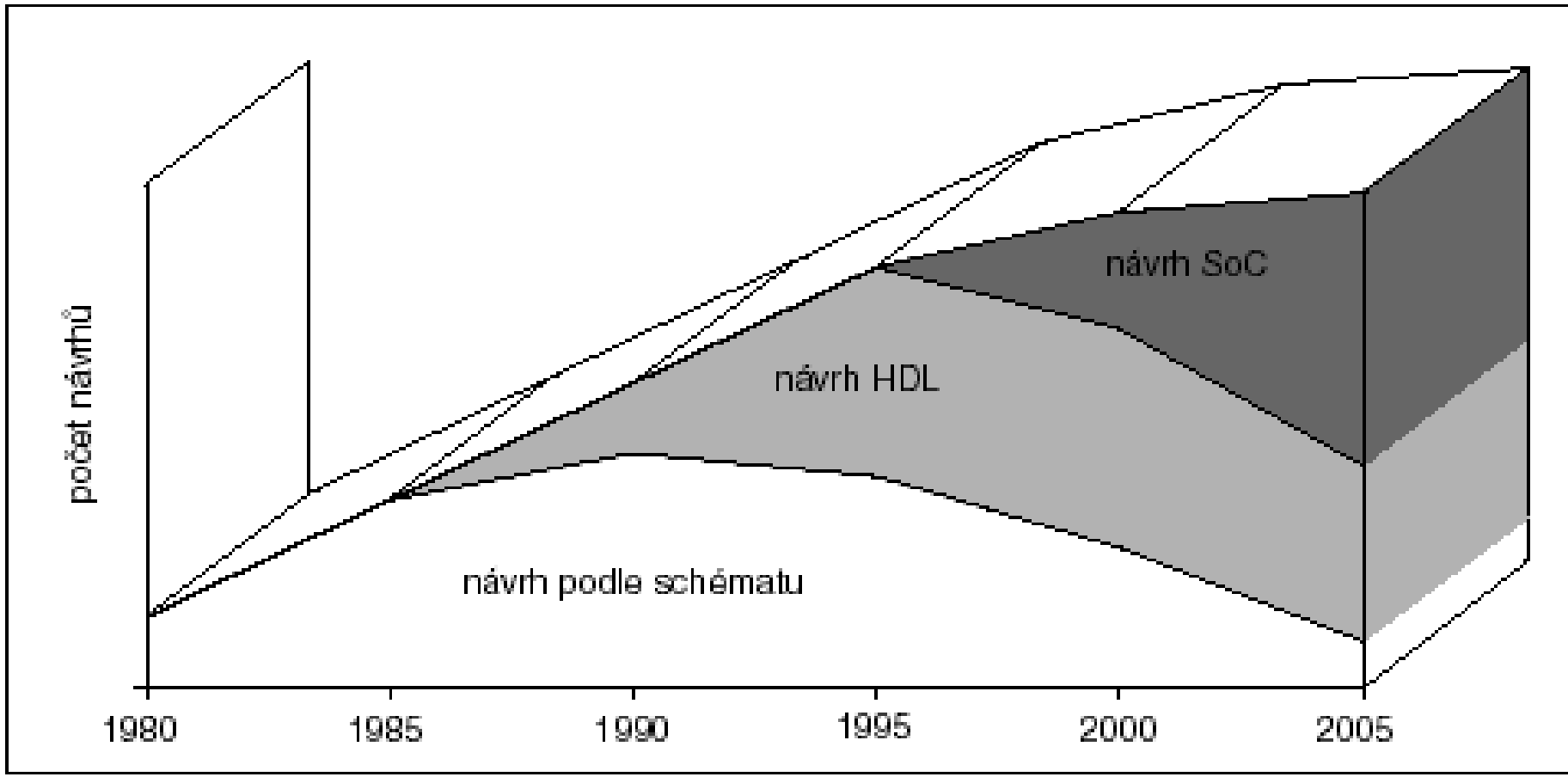
Transformáciu z tohoto jazyka do konkrétneho zapojenia používaných prvkov (klopné obvody, hradla atd.) zabezpečuje nástroj pre syntézu. Na overenie správnosti funkcie slúži nástroj na simuláciu, ktorý sa používa pred syntézou aj po nej.

V posledných rokoch sa na trhu presadili dva jazyky HDL:

- v Európe dominuje ***VHDL***,
- zatiaľ čo v USA ***Verilog***.

Moderné postupy pri návrhu PLD

Väčšina nástrojov významných svetových firiem z oblast
automatizácie návrhových prác v elektronike (EDA) podporuje oba
jazyky HDL a umožňuje v tomto smere zmiešaný návrh, čo návrhárom
značne uľahčuje prácu.



Moderné postupy pri návrhu PLD

Za účelom zabezpečenia hladkého prenosu syntezovateľných návrhov do vývojových nástrojov (pokiaľ možno všetkých výrobcov) musí byť nástroj na syntézu schopný generovať výslednú logickú schému vo forme netlistu, najlepšie v štandardizovanom formáte ***EDIF***.

Za účelom overenia časového správania sa obvodu, musí byť k dispozícii spätný prenos informácií o oneskoreniach z vývojových nástrojov späť do netlistu. Štandardný formát pre súbory obsahujúce oneskorenia jednotlivých hradiel a prepojovacích vodičov je ***SDF***.

Moderné postupy pri návrhu PLD

Grafický popis obvodu

Popisný kód môže byť často bez akejkoľvek dokumentácie, pričon
použiteľné pravidlá pre zrozumiteľné komentáre sa ťažk
presadzujú.

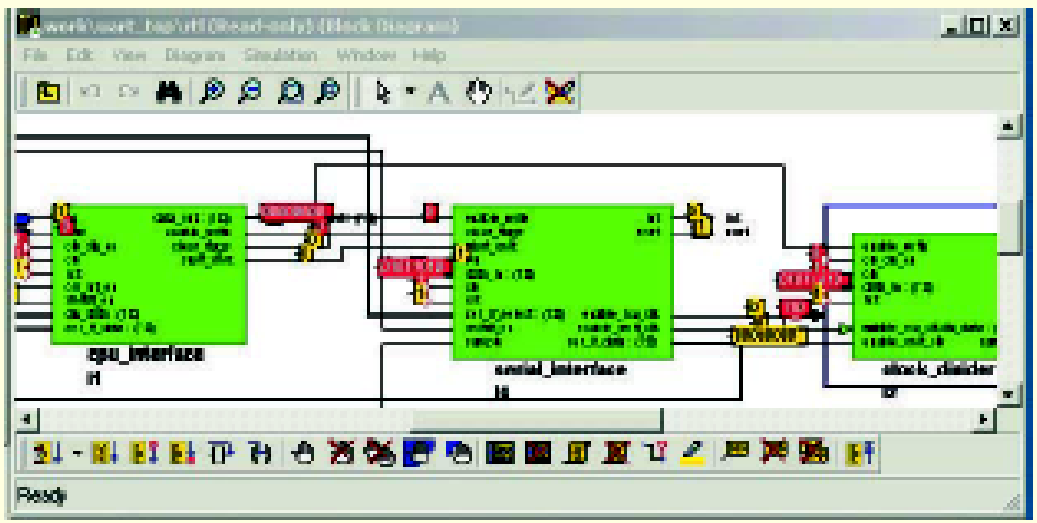
V takýchto situáciách sa ako riešenie ponúka využitie grafick
orientovaných nástrojov. Je ich možno použiť nielen k popis
návrhu, ale aj na zviditeľnenie opakovane používaných aleb
externe vyvinutých návrhových blokov.

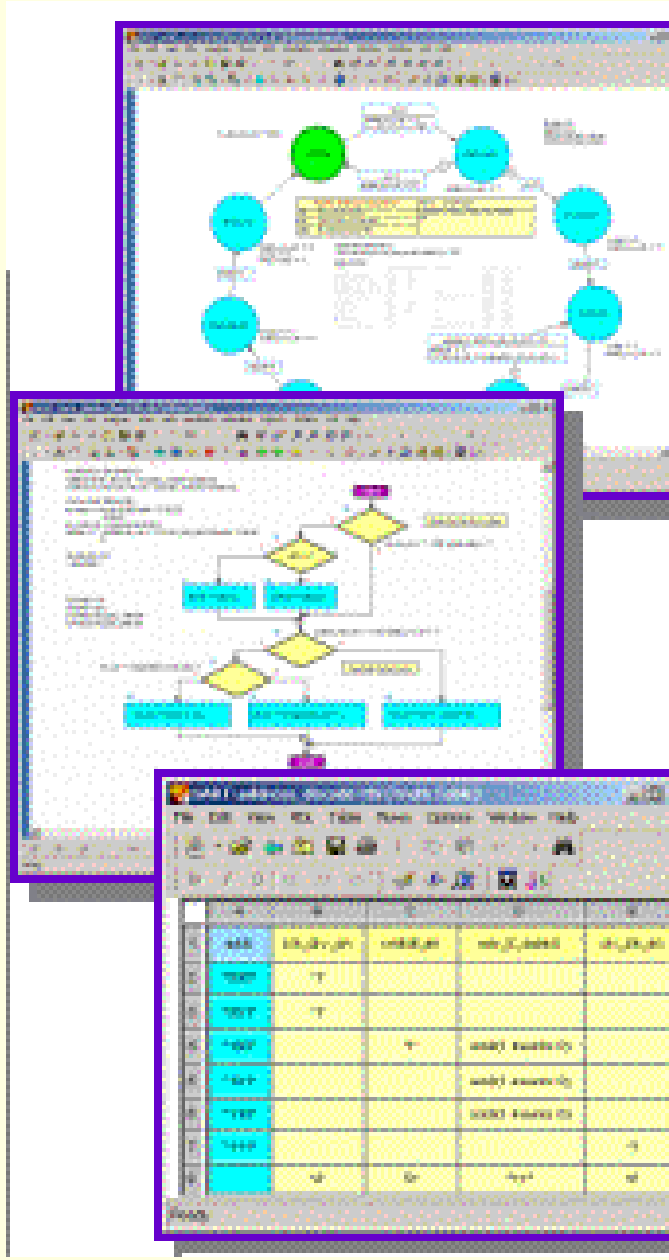
Moderné postupy pri návrhu PLD

Grafický popis obvodu

Umožňujú návrhárom sústrediť sa iba na riešenie vlastnej úlohy– nieje treba sa zaoberať rutinnými prácami (napr. prepojovaním jednotlivých blokov).

Grafické možnosti popisu funkcie navrhovaného obvodu sú patrné z Obr.





Moderné postupy pri návrhu PLD

Efektívna verifikácia

Je veľkou výhodou, keď majú simulátory ľahké ovládanie a zároveň umožňujú verifikovať obvod v najkratšom možnom čase. V súčasnosti výrazne vzrástli požiadavky na kvalitné a robustné verifikačné prostredia pozostávajúce z tzv. blokov „test bench“, ktoré môžeme navrhnúť pomocou HDL a sady testov.

Ľahko ovládateľná logická syntéza

Aby bolo možné získať uspokojivé výsledky logickej syntézy pre technológiu zvoleného výrobcu FPGA, je dôležitá jednoduchá a úplná formulácia všetkých možných obmedzení (prípustná plocha čipu, pracovná frekvencia, apod.).

Nástroj na syntézu musí tiež umožňovať kontrolu dodržiavania návrhových obmedzení. Pri nezrovnalostiach musí byť syntezátor schopný rýchlej optimalizácie s prihliadnutím na požiadavky na plochu a pracovnú frekvenciu.

Moderné postupy pri návrhu PLD

Bezproblémová integrácia nástrojov

V optimálnom prípade návrhový systém nevyžaduje opakované zadávanie rovnakých údajov a umožňuje hladký prechod z jednej fáze návrhu do inej. Takáto bezproblémová integrácia však v žiadnom prípade nesmie v žiadnej fáze návrhu nútiť návrhára k nejakým kompromisom, alebo k výrazným zmenám.

Riešenie „na mieru“

S rôznou veľkosťou prvkov a rôznou zložitosťou návrhu súvisia aj rôzne požiadavky na návrhové prostredia, hlavne s ohľadom na kvalitu a rozsah funkcií a s tým samozrejme zodpovedajúca cena výsledného riešenia. Ideálnym výsledkom je schopnosť prispôsobiť sa potrebám užívateľa s možnosťou využiť kvalitu a rozsah funkcií pri súčasnom zhodnotení investícií vložených do vývoja SW a HW.

7.3 Mentor Graphic- FPGA Advantage

Vyššie uvedené požiadavky zohľadnila firma Mentor Graphics pri vývoji súboru svojich nástrojov:

- ***HDL Designer Series,***
- ***ModelSim a***
- ***Leonardo Spectrum.***

Z pohľadu práce návrhára číslicových systémov pokrývajú uvedené nástroje celý proces návrhu obvodov FPGA, teda vývoj, správu projektu, verifikáciu aj prenos dát do cieľovej technológie.

Všetky tieto nástroje predstavujú to najlepšie, čo je v jednotlivých segmentoch na svete k dispozícii. Ich integráciou vznikol programový súbor s názvom FPGA Advantage- postup pri návrhu obvodov FPGA.

7.3 Mentor Graphic- FPGA Advantage

Medzi významné výhody FPGA Advantage patrí:

neobmedzené používanie oboch jazykov vo všetkých nástrojoch (pri návrhu je možné použiť ľubovoľnú kombináciu VHDL i Verilog),

podpora viacerých hardwarových platforiem pri zachovaní všetkých funkcií (nástroje pracujú v prostredí Windows XX aj v prostredí systému UNIX pri zachovaní stopercentnej kompatibilite medzi jednotlivými platformami),

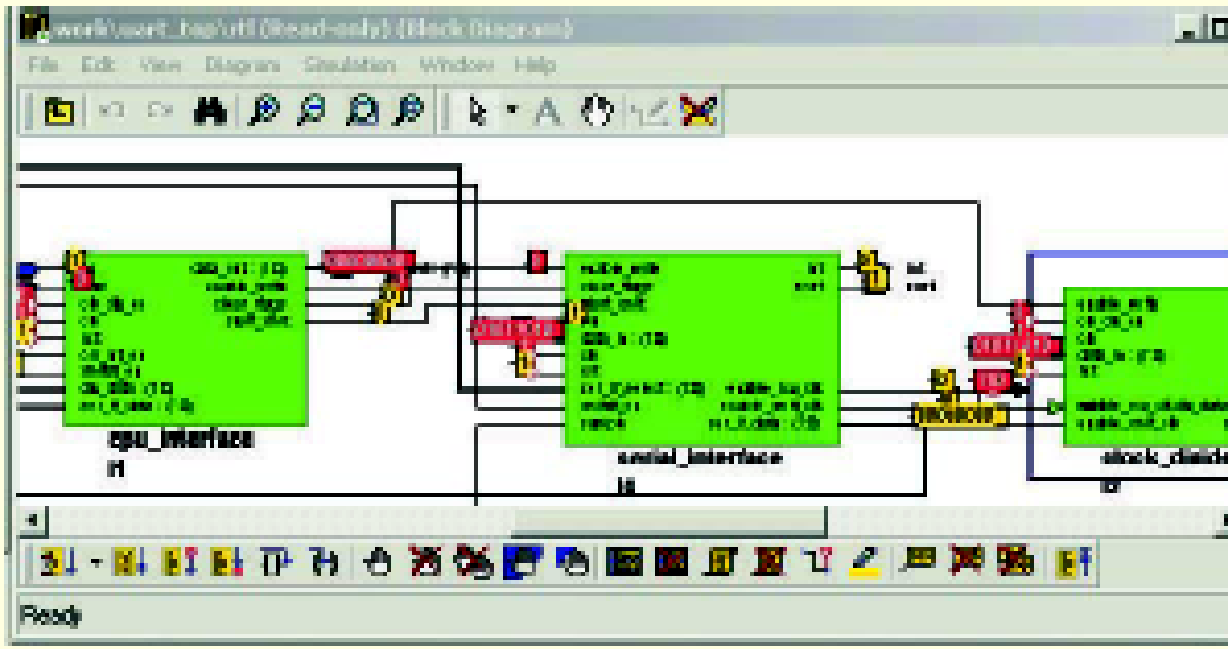
podpora všetkých štandardných formátov dát (návrhár tým nieje viazaný na databázu jedného výrobcu).

7.3 Mentor Graphic- FPGA Advantage

Popis obvodu a simulácia RTL

Programový prostriedok HDL Designer, určený pre popis obvodov, sa skladá z niekoľko relatívne samostatných modulov (editora pre grafické popisy obvodov, prostredia pre správu dát, pre spúšťanie simulácie aj syntézy atd.).

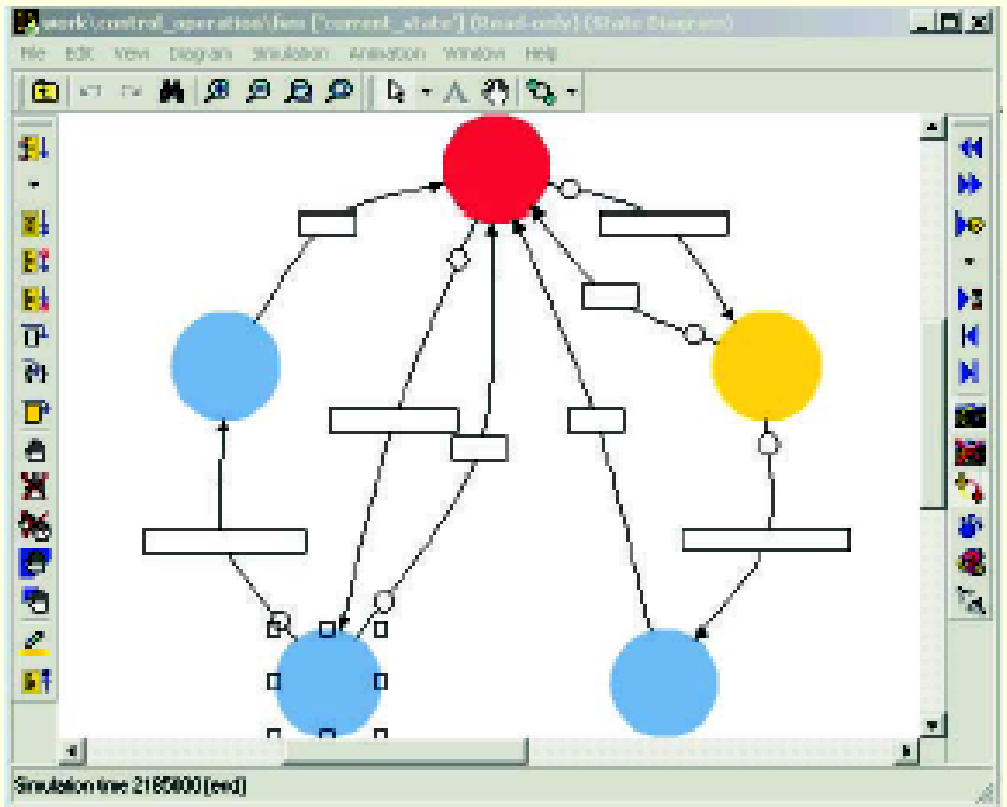
Pre popis obvodov ponúka HDL Designer rad grafických funkcií, ktoré návrhárom uľahčujú vytváranie kódu HDL. Editor ***blokových diagramov*** umožňuje zadávanie hierarchických štruktúr.

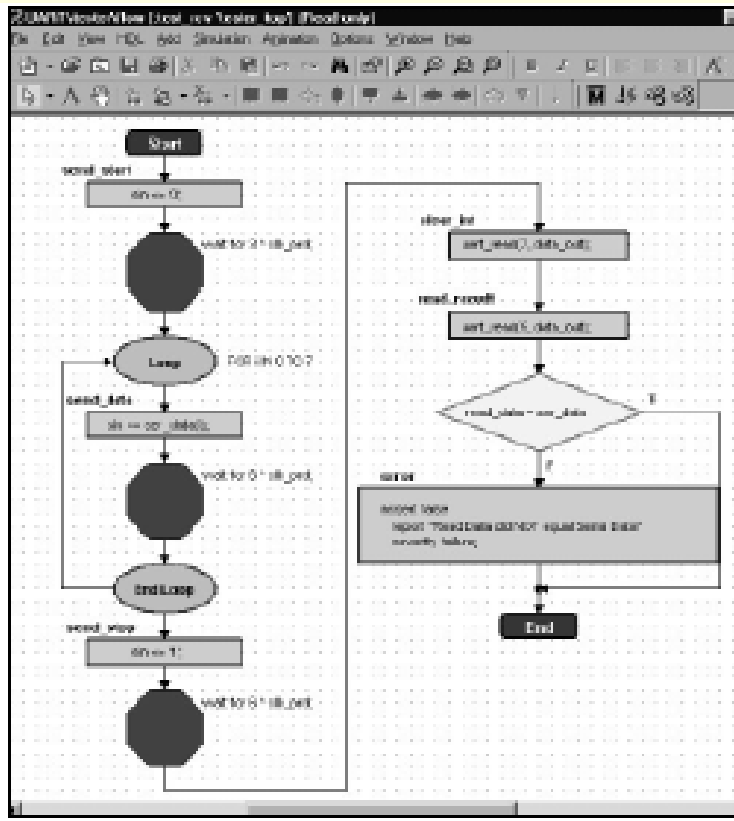


7.3 Mentor Graphic- FPGA Advantage

Popis obvodu a simulácia RTL

Editorom stavových diagramov je možné graficky zapísať aj najzložitejšie stavové automaty. Editor ***vývojových diagramov*** umožňuje zachytiť prostredie pre test a algoritmy. Pre popis logických vzťahov je k dispozícii editor pravdivostných tabuliek. Je samozrejme možné využiť len textový zápis VHDL alebo Verilog a „grafiku“ včleniť až neskôr. Tieto možnosti sú demonštrované na Obr.





7.3 Mentor Graphic- FPGA Advantage

HDL Designer prevedie pripravený grafický popis do jazyka HDL.

Už existujúce moduly návrhu popísané pomocou jazyka HDL je možné importovať a pripojiť k práve vznikajúcej databáze. Funkcia HDL2Graphics umožňuje transformáciu vzniknutých blokov naspät do grafickej formy. Tým je zabezpečené rozpoznanie hierarchických vzťahov, stavových automatov aj vývojových diagramov.

Na overenia funkcie návrhu slúži návrhárom nástroj na simuláciu. V programovom súbore FPGA Advantage plní túto úlohu programový prostriedok ModelSim, v súčasnosti jeden s najpoužívanejších simulátorov na verifikáciu návrhu v jazykoch HDL.

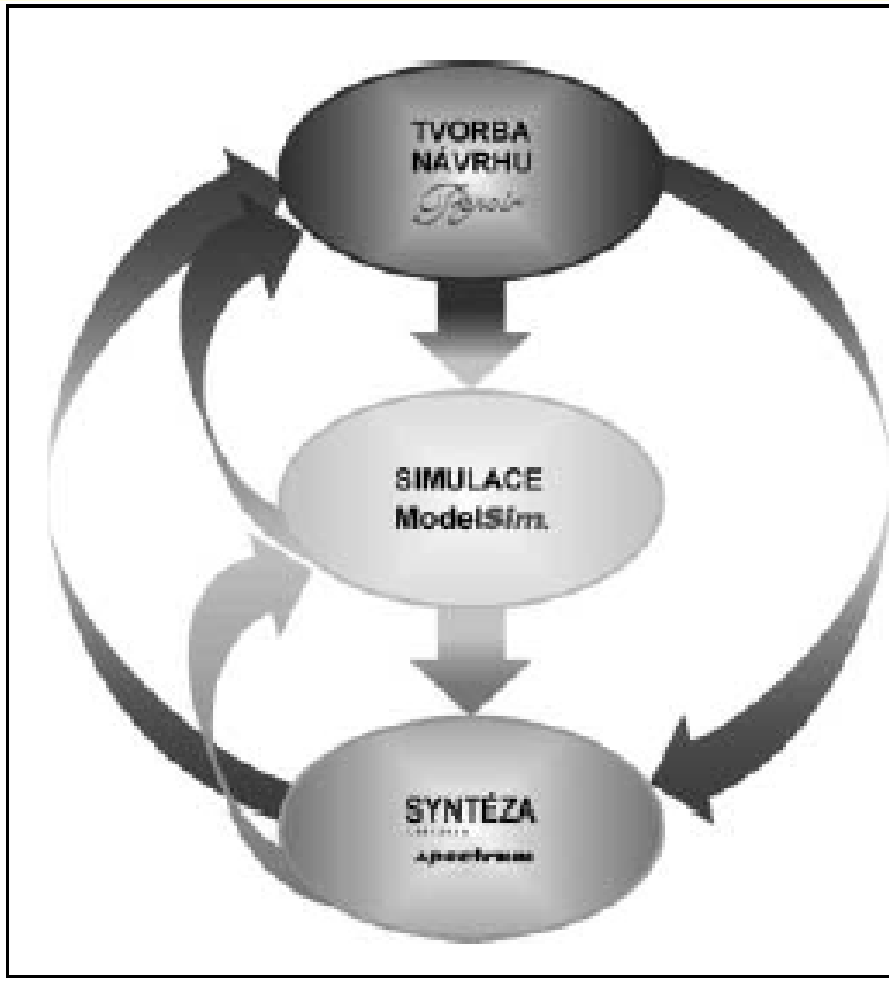
7.3 Mentor Graphic- FPGA Advantage

Syntéza

Nástroj na syntézu je v FPGA Advantage integrovaný program *Leonardo Spectrum*.

Návrhár najprv špecifikuje obmedzenia (časovanie, nároky na plochu atd.), Leonardo Spectrum potom pri dodržaní týchto obmedzení uskutoční návrh prostredníctvom syntézy a optimalizácie do cieľovej technológie.

Leonardo Spectrum postupuje návrh vo formáte **EDIF** na ďalšie spracovanie prostriedku pre rozmiestnenie a prepojenie príslušného výrobcu FPGA.



7.3 Mentor Graphic- FPGA Advantage

Rozmiestnenie, prepojenie a simulácia na úrovni hradiel

Z exportovaného netlistu vytvorí nástroj pre rozmiestnenie a prepojenie tak konfiguráciu dát, ktorá je vhodná k naprogramovaniu cieľového obvodu FPGA. Okrem toh tiež poskytuje dáta vo formáte SDF s časovými informáciami príslušného návrhu FPGA, ktor zodpovedajú reálnemu rozmiestneniu prvkov a dĺžkam vodičov.

Údaje, ktoré sú špecifické obvodu danej technológie získa simulátor z príslušnýc knižníc VHDL alebo Verilog (alebo oboch dohromady). Po kompilácií a mapovaní týcht knižníc pre simuláciu môže program ModelSim načítať zodpovedajúce dáta SDF a realizova časovú simuláciu na úrovni hradiel. Tato simulácia vyhodnocuje, či a ako sú dodržané časov obmedzenia.

Pokiaľ sa pri návrhu vyskytnú problémy, musí návrhár v nástroji na syntézu zmeni použité obmedzenia a znovu uskutočniť syntézu, rozmiestnenie a prepojenie. Výsledok j potom nutné znovu overiť opätovnou simuláciou na úrovni hradiel. Na tento účel m Leonardo Spectrum celý rad funkcií na ladenie. Napríklad možnosť extrahovať a zviditeľni kritickú cestu, čím si návrhár vytvorí presnejší obraz skutočného stavu návrhu.

7.3 Mentor Graphic- FPGA Advantage

Spolupráca výrobcov obvodov FPGA s firmami z oblasti EDA

Univerzálne použiteľné programové prostriedky ako napríklad:

- HDL Designer,
- ModelSim,
- Leonardo Spectrum

alebo podobné nástroje iných firiem z oblasti EDA nachádzaj optimálne uplatnenie hlavne u návrhárov obvodov FPGA, ktorí v svojich návrhoch používajú obvody FPGA od rôznych výrobco súčasne.

7.3 Mentor Graphic- FPGA Advantage

Záver

Programovaný súbor FPGA Advantage umožňuje pohodlný, rýchly a spoľahlivý návrh obvodov FPGA.

Nástroje pre popis obvodov, simuláciu aj syntézu podporujú oba jazyky – VHDL i Verilog, pracujú na všetkých v praxi bežne používaných hardwarových platformách a využívajú všetky príslušné štandardne formáty dát.

Výhody tohoto programového systému vyniknú hlavne pri zložitejších návrhoch.

8 HDL

- 8.1 VHDL
- 8.2 Verilog HDL

8.1 VHDL- Základná štruktúra

Model (konštrukcia) má v jazyku VHDL dve základné časti:

deklarácia entity *(entity declaration)*- popisuje vstupy a výstupy konštrukcie

telo architektúry *(architecture body)*- definuje funkcie konštrukcie

Príklad VHDL kódu

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;

ENTITY orhradlo IS
      PORT(
            PB1, PB2    : IN  STD_LOGIC;   -- vstupy
            MSD_Dp      : OUT STD_LOGIC ); -- výstu
END othradlo;

ARCHITECTURE Dp OF orhradlo IS
BEGIN
      MSD_Dp <= NOT( NOT PB1 OR NOT PB2 );
END Dp;
```

8.1 VHDL- Deklárácia entity

Brány sú v deklarácii entity deklarované v zátvorke za kľúčovým slovom PORT. Patria k dátovým objektom. Deklarácia brány sa skladá z mena brány, t.j. identifikátorov príslušného signálu a ďalej z určeného smeru jeho prenosu (mód - *mode*) a typu dát, ktoré signál predstavuje.

Módy môžu byť:

IN	–vstup
OUT	–výstup (výstupný signál nemôže byť použití vo vnútri entity)
BUFFER	–výstup so spätnou väzbou, signál z takéhoto výstupu môže byť použitý vo vnútri entity
INOUT	–obojsmerný vývod

8.1 VHDL-Telo architektúry

Architektúra sa môže opisovať rôznymi štýlmi jazyka VHDL.

Pojem štýlu nie je v tomto jazyku priamo definovaný, ale niektoré jazykové konštrukcie vo VHDL môžu byť zaradené do skupín, ktoré tieto štýly predstavujú.

Najčastejšie sa hovorí:

- **o behaviorálnom štýle,**
- **o štýle opisujúci tok dát a**
- **o štrukturálnom štýle.**