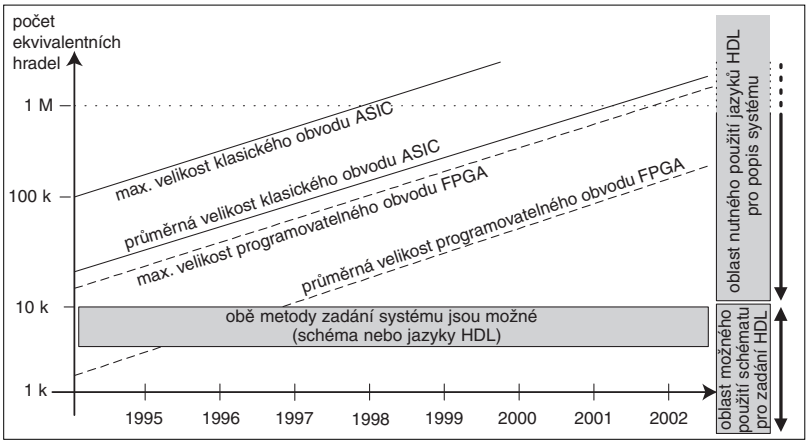
Moderní postupy při návrhu programovatelných logických obvodů

Úvod

Tento příspěvek navazuje na sérii článků z minulých čísel, které se zabývaly problematikou programovatelných logických obvodů. Na rozdíl od přehledu a porovnání návrhových metod [3] se tentokrát zaměříme na konkrétní etapy v návrhovém procesu a na způsoby, kterými je možné tyto etapy v dnešní době efektivně řešit.

Nové technologie výroby programovatelných obvodů umožňují integrovat stále složitější systémy do jediné součástky. Tradiční návrhové metody již současným technologickým možnostem a požadavkům trhu nedostačují. Není proto divu, že se hledají nové cesty, jak s tímto vývojem udržet krok.

Do popředí zájmu vývojářů se tak dostávají nové metody v návrhu součástek FPGA a s tím související i požadavky na nástroje používané při návrhu. Příkladem vhodného řešení návrhového prostředí může být programový soubor *FPGA Advantage* (dříve *Packaged Power*) firmy Mentor Graphics. S jeho pomocí zvládnou návrháři programovatelných logických systémů bez problémů přechod od klasických metod návrhu schématu k využívání popisů prostřednictvím jazyků HDL.

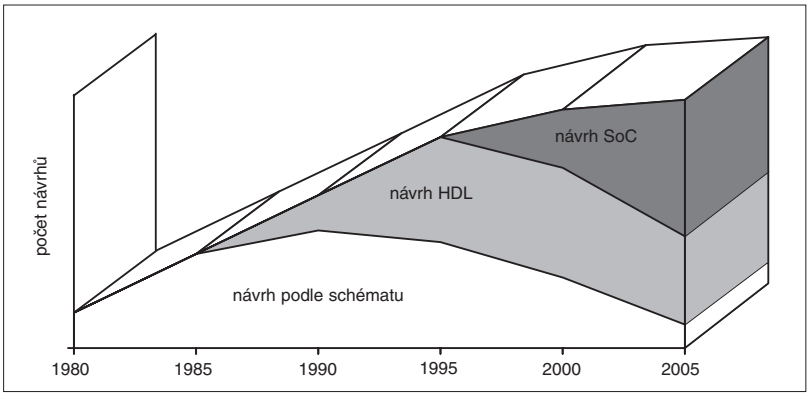


Obr. 1 a) Rostoucí složitost FPGA vyžaduje nové metody návrhu

Motivace

Jak již bylo blíže popsáno v [1], dosáhli výrobci součástek FPGA a CPLD v posledních několika letech díky novým technologiím velkého pokroku. Zvláště v oblasti submikronových struktur a v souvislosti s dalšími průkopnickými vylepšeními se jim daří nabízet stále výkonnější součástky při zachování konkurenceschopných cen. Dnes již existují součástky FPGA obsahující až několik miliónů hradel.

Takto rozsáhlé struktury FPGA se zatím používají jen v relativně malém počtu aplikací. Je však důležité, že odrážejí celkový trend. I návrhy průměrné velikosti jsou ovšem v poslední době složitější a používá se při nich větších součástek (*obr. 1 a*). Přitom 20–50 tisíc ekvivalentních hradel v jedné součástce je v současnosti naprosto běžná hustota a nepředstavuje žádnou výjimku.



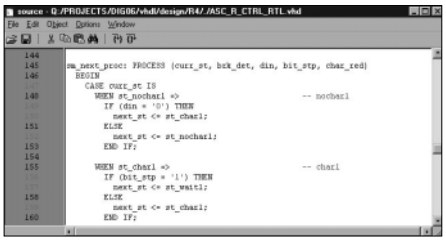
Obr. 1 b) Návrh SoC – třetí generace EDA

Návrháři se tak ocitají před novou výzvou. Tradiční návrhy s využitím schématu se stávají až nezvládnutelně složitými a časově náročnými. Zároveň se zvyšuje tlak na maximální zkracování vývoje funkčního vzorku. Není proto divu, že poptávka po nových metodách je oprávněná a je zřejmé, že využití jazyků HDL se stává nezbytností nejen v oblasti návrhu klasických zákaznických obvodů ASIC, ale i součástek FPGA.

V návrhu obvodů ASIC proběhla podobná revoluce začátkem devadesátých let. Ukázalo se, že při složitosti více než 10 tisíc ekvivalentních hradel je pro popis číslicových systémů rychlejší použití návrhových metod podporujících jazyky HDL. Tyto metody zároveň zjednodušují opakované používání bloků, integraci stávajících návrhů do nových projektů a v poslední době i integraci zakoupených makrobloků (tzv. makra IP – Intellectual Property). Podrobnější informace a příklady maker IP byly popsány v [3].

Návrh s využitím jazyků HDL

Na rozdíl od vytváření klasického logického schématu návrhář nyní popisuje funkci obvodu pomocí vhodného jazyka HDL (Hardware Description Language). Převod z tohoto jazyka do konkrétního zapojení používaných prvků (klopné obvody, hradla atd.) zajišťuje nástroj pro syntézu. K ověření správnosti funkce slouží nástroj pro simulaci, který se používá před syntézou i po ní.



Obr. 2 Příklad popisu části obvodu jazykem VHDL

V posledních letech se na trhu prosadily dva jazyky HDL – v Evropě dominuje VHDL, zatímco v USA Verilog. Většina nástrojů předních světových firem z oblasti automatizace návrhových prací v elektronice (EDA) podporuje oba jazyky HDL a umožňuje v tomto směru smíšený návrh, což návrhářům značně usnadňuje práci. Základní zásady tvorby popisů HDL z hlediska bezpečného a snadno syntezovatelného logického návrhu byly popsány v [3].

Složitost navrhovaných systémů dosahuje dnes i pro součástky FPGA úrovně SoC (System on a Chip) viz *obr 1 b)*. To vyžaduje týmový způsob práce, někdy i na úrovni mezinárodních návrhářských týmů.

Tato nová metodika s sebou přináší i nové problémy, se kterými se návrháři musí vyrovnat. Příkladem může být použití rozdílných HW platforem, různých nástrojů EDA a zejména obou jazyků HDL, což znesnadňuje integraci jednotlivých bloků.

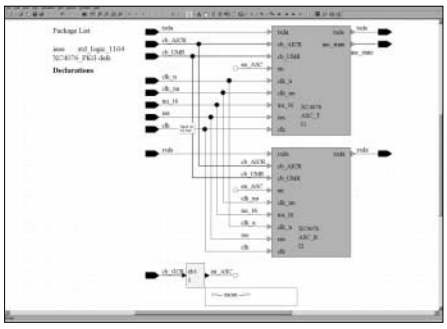
Doporučený postup při návrhu

Aby byl návrhář schopen překonat všechny výše uvedené problémy, musí respektovat určité požadavky na návrhové prostředí a postup při návrhu obvodů FPGA.

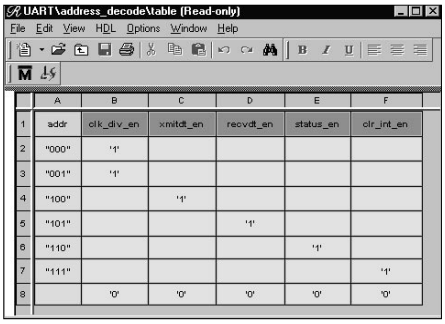
Všeobecná podpora standardů a formátů dat popisných jazyků

Všechny nástroje musí tyto standardy podporovat počínaje popisem návrhu, přes verifikaci a syntézu, až po převod dat do následných nástrojů užívaných různými výrobci FPGA. Pokud se mají jednotlivé části obvodu, vzniklé z různých zdrojů, integrovat do jednoho návrhu, je nezbytné, aby v procesu návrhu byly podporovány oba jazyky HDL.

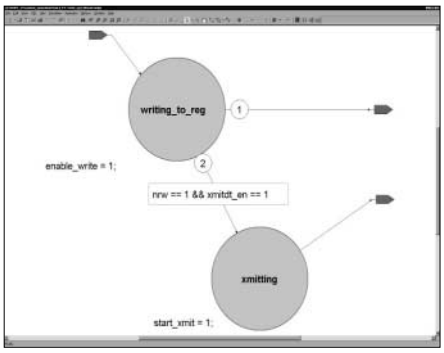
To platí jak pro popis návrhu, tak i pro simulaci a syntézu. Pro zajištění hladkého přenosu syntezovatelných návrhů do následných nástrojů (pokud možno všech výrobců) musí být nástroj pro syntézu schopen generovat výsledné logické schéma ve formě netlistu, nejlépe ve standardizovaném formátu EDIF. Současně je nutno automaticky rozpoznat signály a hierarchické struktury specifické pro jednotlivé výrobce a přihlédnout k jejich označení. Aby bylo možno ověřit i časové chování obvodu, musí být umožněn zpětný přenos informací o zpoždění z následných nástrojů zpět do netlistu. Standardní formát pro soubory obsahující zpoždění jednotlivých hradel a propojovacích vodičů je SDF.



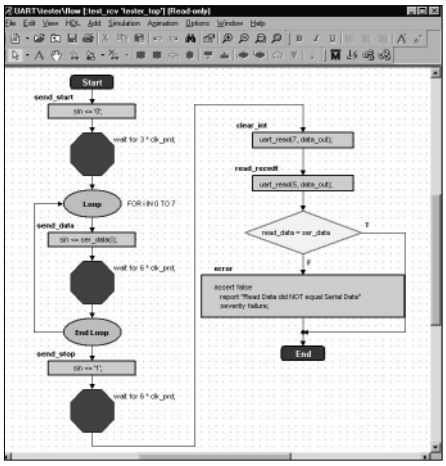
Obr. 3 Blokový diagram



Obr. 5 Pravdivostní tabulka



Obr. 4 Diagram přechodů



Obr. 6 Vývojový diagram

Grafický popis obvodu

Popisy návrhových bloků založené čistě na popisném jazyku (VHDL/Verilog) jsou zpravidla srozumitelné pouze pro jejich autora. Popisný kód může být často bez jakékoliv dokumentace, přičemž použitelná pravidla pro srozumitelné komentáře se těžko prosazují. V takových situacích se jako řešení nabízí využití graficky orientovaných nástrojů. Je možné je použít nejen k popisu návrhu, ale i ke zviditelnění opakovaně používaných nebo externě vyvinutých návrhových bloků. Dále umožňují návrháři soustředit se pouze na řešení vlastního úkolu – není třeba se zabývat rutinní prací (např. propojováním jednotlivých bloků). Grafické možnosti popisu funkce navrhovaného obvodu jsou patrné z *obr. 3* až *obr. 6*.

Snadno ovladatelná logická syntéza

Aby bylo možné získat uspokojivé výsledky logické syntézy pro technologii zvoleného výrobce FPGA, je důležitá jednoduchá a kompletní formulace všech možných omezení (přípustná plocha čipu, provozní frekvence, časování na rozhraních, apod.). Samozřejmostí by měla být i podpora nástrojů pro rozmístění a propojení ze strany všech důležitých výrobců. Nástroj pro syntézu také musí umožňovat kontrolu dodržování návrhových omezení. Při nesrovnalostech musí být syntezátor schopen rychlé optimalizace s přihlédnutím na požadavky na plochu a provozní frekvenci.

Efektivní verifikace

Funkci navrženého obvodu je nutno ověřit neboli verifikovat pomocí simulace. Nástroje používané pro simulaci by měly podporovat pokud možno oba používané jazyky HDL. Je velkou výhodou, když simulátory mají snadné ovládání a zároveň umožňují verifikovat obvod v nejkratším možném čase. V současné době výrazně vzrostly požadavky na kvalitní a robustní verifikační prostředí, sestávající z tzv. bloku „test bench“ (blíže [3], s. 8), který lze navrhnout rovněž pomocí HDL a sady testů.

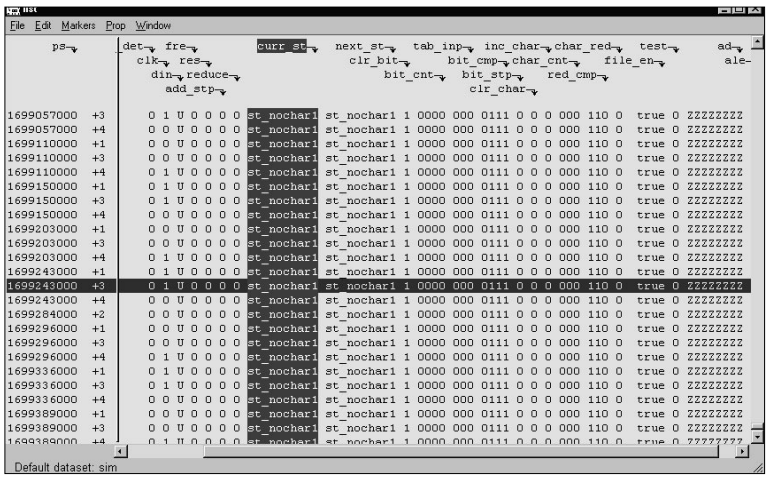
Bezproblémová integrace nástrojů

V optimálním případě návrhový systém nevyžaduje opakované zadávání stejných údajů a umožňuje hladký přechod z jedné fáze návrhu do jiné. Úzce spolupracovat by měly zejména moduly pro grafické zadání a pro simulace, v ideálním případě je možné simulátor spustit přímo z grafického modulu a naopak. Grafický popis, simulace i syntéza by také měly být natolik těsně propojeny, aby umožnily jednoduché porovnání na úrovni RTL, funkční simulace a zároveň simulace na úrovni hradel po syntéze. Pokud je však potřeba provést modifikace, měly by na ně být vynaloženy co nejmenší náklady.

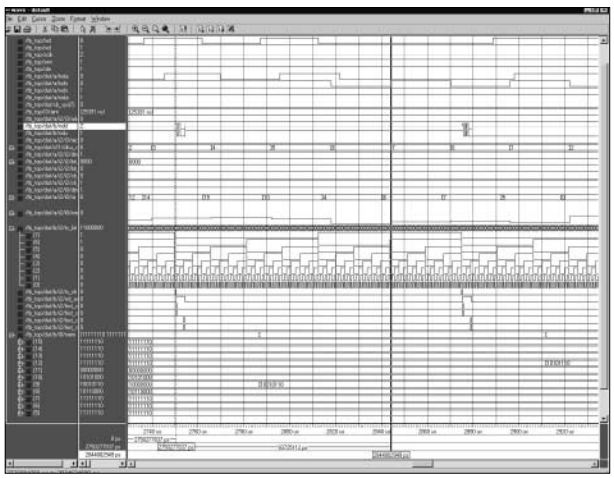
Takováto bezproblémová integrace však v žádném případě nesmí v žádné fázi návrhu nutit návrháře k nějakým kompromisům nebo k výrazným změnám.

Řešení na míru

S různou velikostí prvků a složitostí návrhu souvisí i různé požadavky na návrhové prostředí, zejména s ohledem na kvalitu a rozsah funkce a s tím samozřejmě souvisí i cena výsledného řešení. Ideálním výsledkem je schopnost přizpůsobit se potřebám uživatele s možností využít kvalitu a rozsah funkce při současném zhodnocení investic vložených do vývoje SW a HW. S různými metodami návrhu systémů a jednotlivými vývojovými etapami jste se již seznámili v [3].



Obr. 7 Výsledek simulace v textové podobě



Obr. 8 Výsledek simulace v grafické podobě

FPGA Advantage – příklad vhodného řešení

Výše uvedené požadavky zohlednila firma Mentor Graphics při vývoji palety svých nástrojů *HDL Designer Series*, *ModelSim* a *Leonardo Spectrum*. Z hlediska práce návrháře číslicových systémů pokrývají uvedené nástroje celý proces návrhu součástek FPGA, tedy vývoj, správu projektu, verifikaci i přenos dat do cílové technologie.

Všechny tyto nástroje představují to nejlepší, co je v jednotlivých segmentech na světě k dispozici. Jejich integrací vznikl programový soubor s názvem *FPGA Advantage*, představující svým způsobem nový revoluční postup v návrhu součástek FPGA.

Mezi významné přednosti *FPGA Advantage* patří:

- neomezené používání obou jazyků u všech nástrojů (VHDL i Verilog lze při návrhu použít v libovolné kombinaci),
- podpora více hardwarových platforem při současném zachování všech funkcí (nástroje pracují v prostředí Windows 95, 98 i NT a též v prostředí systému UNIX – Sun i HP při současném zachování stoprocentní kompatibility mezi jednotlivými platformami),
- důsledná podpora všech standardních formátů dat (návrhář tím není vázán na databázi jednoho výrobce).

Všechny nástroje, které budou dále popsány, se dají kombinovat s nástroji jiných výrobců, pokud tito podporují zmíněné standardy.

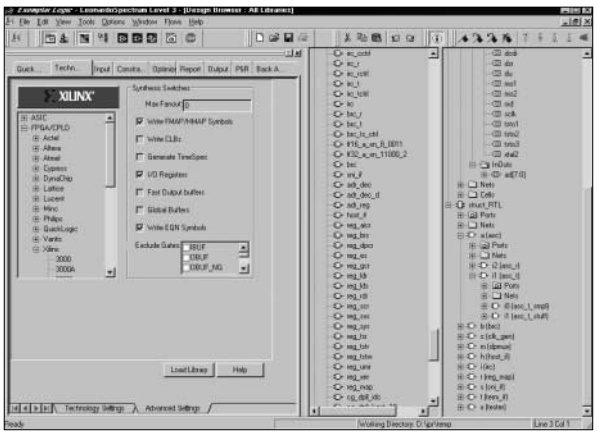
Popis obvodu a simulace RTL

Programový prostředek *HDL Designer*, určený pro popis obvodu, se skládá z několika relativně samostatných modulů (editorů pro grafické popisy obvodu, prostředí pro správu dat, pro spuštění symulace i syntézy atd.).

Pro popis obvodu nabízí *HDL Designer* řadu grafických funkcí, které návrháři ulehčují vytváření kódu HDL. Editor blokových diagramů umožňuje zadávání hierarchických struktur. Editorem stavových diagramů lze graficky zapsat i nejsložitější stavové automaty. Editor vývojových diagramů umožňuje zachytit prostředí pro test a algoritmy. Pro popsání logických vztahů je k dispozici editor pravdivostních tabulek. Je samozřejmě možné využít i čistě textového zápisu VHDL nebo Verilog a grafiku včlenit až později. Tyto možnosti jsou demonstrovány na *obr. 3* až *obr. 6*.

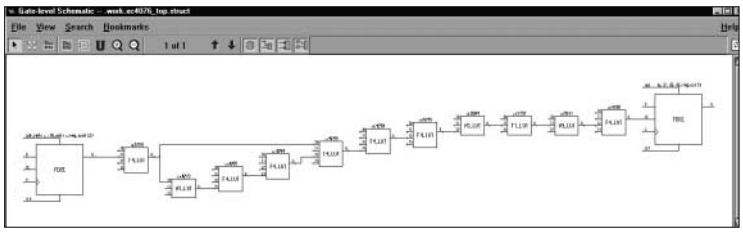
HDL Designer poté převede připravený grafický popis do jazyka HDL. Kódy VHDL nebo Verilog, nezbytné pro simulaci a syntézu, se generují pouhým stisknutím tlačítka.

Již existující moduly návrhu popsané pomocí jazyka HDL je možné importovat a připojit k právě vznikající databázi. Funkce *HDL2Graphics* umožňuje tak přeměnu vzniklých bloků zpět do grafické formy. Tím je zajištěno rozpoznání hierarchických vztahů, stavových automatů i vývojových diagramů.

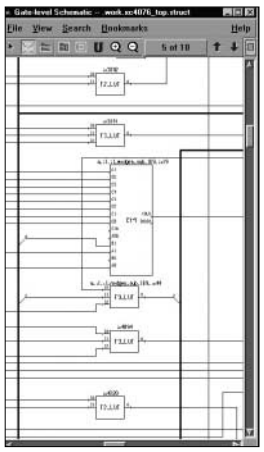


Obr. 9 Ovládání logické syntézy

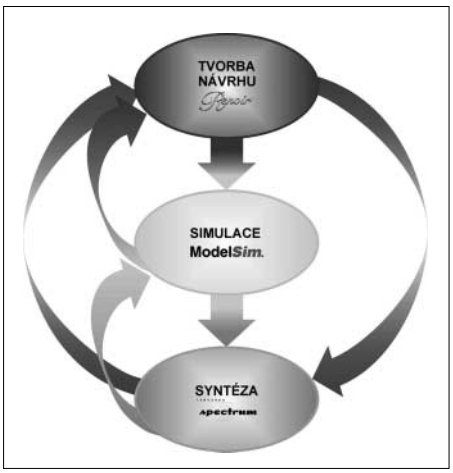
K ověření funkce návrhu slouží návrháři nástroj pro simulaci. V programovém souboru *FPGA Advantage* plní tuto úlohu programový prostředek *ModelSim*, v současnosti v celosvětovém měřítku nejpoužívanější simulátor pro verifikaci návrhu v jazycích HDL. *FPGA Advantage* obsahuje funkci *Design Browser Cockpit*, která automaticky používá kód HDL z programu *HDL Designer* a spouští simulaci. Rozšířené prostředí pro odstranění chyb umožňuje zviditelnění verifikace využitím animované simulace s různobarevnými kódy, křížovými odkazy a křížovým zvýrazněním vztahů mezi prostředím pro simulaci, zdrojovým kódem HDL a grafickým objektem. Verifikaci zjednodušuje i možnost simulace kódu HDL a grafické reprezentace po jednotlivých krocích a zobrazení hodnot signálu přímo v grafickém prostředí. Takto lze velice rychle nalézt chyby vzniklé v předchozí etapě. Návrhář má možnost iteračně provádět potřebné úpravy v návrhu a spouštět další simulační běhy.



Obr. 11 Kritická cesta



Obr. 10 Propojení bloků uvnitř FPGA



Obr. 12 Programový soubor FPGA Advantage

Syntéza

Jako nástroj pro syntézu je v *FPGA Advantage* integrován program *Leonardo Spectrum*. Prostředí pro syntézu i vlastní návrh se automaticky spouští z prohlížeče návrhu. Návrhář nejdříve specifikuje omezení (časování, nároky na plochu atd.), *Leonardo Spectrum* pak při dodržení těchto omezení převede návrh prostřednictvím syntézy a optimalizace do cílové technologie. *Leonardo Spectrum* předává návrh ve formátu EDIF k dalšímu zpracování nástroji pro rozmístění a propojení daného výrobce FPGA.

Průběh rozmístění, propojení a simulace na úrovni hradel

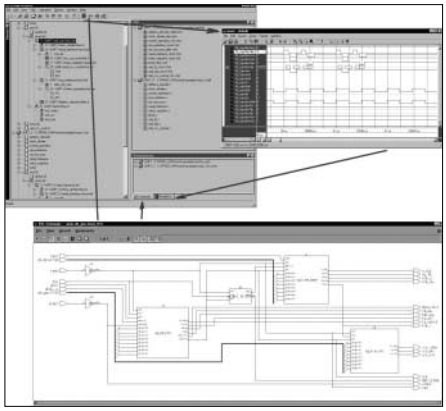
Z exportovaného netlistu vytvoří nástroj pro rozmístění a propojení takovou konfiguraci dat, která je vhodná k naprogramování cílové součástky FPGA. Kromě toho také předává data ve formátu SDF s informacemi o časování návrhů FPGA, které odpovídají reálnému rozmístění komponent a délkám vodičů.

Údaje specifické obvodu dané technologie simulátor získá z příslušných knihoven VHDL nebo Verilog (nebo obou dohromady). Po kompilaci a mapování těchto knihoven pro simulaci může program *ModelSim* načíst odpovídající data SDF a provést časovou simulaci na úrovni hradel. Tato simulace vyhodnocuje, zda a jak jsou dodržena časová omezení.

Pokud se při návrhu vyskytnou problémy, musí návrhář v nástroji pro syntézu změnit užitá omezení a znovu provést syntézu, rozmístění a propojení. Výsledek je pak nutné znovu ověřit další simulací na úrovni hradel. K tomu účelu nabízí *Leonardo Spectrum* řadu funkcí pro ladění. Například možnost rychle extrahovat a zviditelnit kritické cesty, čímž si návrhář vytvoří přesnější obraz skutečného stavu. Jednotlivé prvky kritické cesty lze identifikovat pomocí jména. Je rovněž možné měnit hierarchickou strukturu navrhovaného obvodu. Takto může návrhář cíleně, postupně a s přihlédnutím na časování optimalizovat jednotlivé cesty.

Spolupráce výrobců součástek FPGA s firmami z oblasti EDA

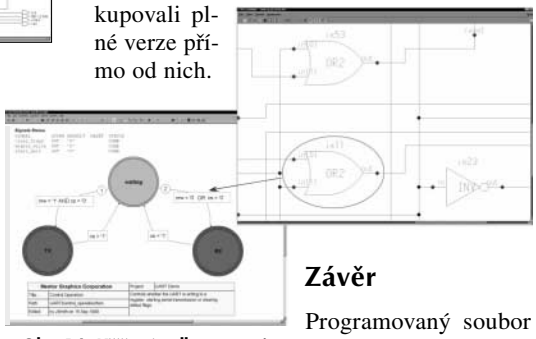
Univerzálně použitelné programové prostředky jako například *HDL Designer, ModelSim, Leonardo Spectrum* nebo obdobné nástroje



Obr. 13 Funkce Design Browser Cockpit

jiných firem z oblasti EDA nacházejí optimální uplatnění zejména u těch návrhářů obvodů FPGA, kteří ve svých výrobcích používají součástky FPGA od více různých výrobců současně. Odpadá tak nutnost učit se pracovat se specifickými nástroji toho kterého výrobce FPGA, nehledě k tomu, že obvod navržený programovými prostředky jednoho výrobce není použitelný pro implementaci do obvodu jiného výrobce.

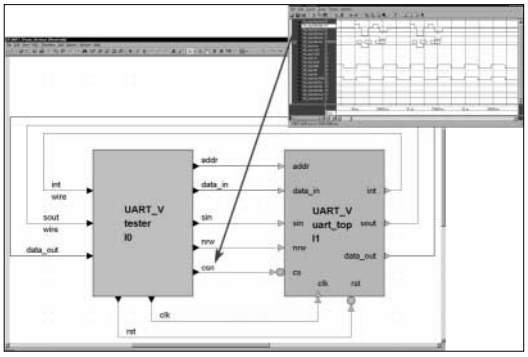
Opodstatněnost využití všech výše popsaných postupů a programových nástrojů při návrhu součástek FPGA potvrzuje i nejnovější vývoj v této oblasti. Výrobci součástek FPGA totiž zjistili, že není praktické, efektivní a vlastně ani dlouhodobě možné vyvíjet vlastní návrhové prostředky, zvláště když se na trhu již jasně profilují programové prostředky firem z oblasti EDA speciálně se zabývajících právě jejich vývojem. Proto firmy jako např. Xilinx, Altera a další začaly uzavírat partnerství s firmami z oblasti EDA, která jim zajišťují přístup k jejich programovým nástrojům, s možností přímého začlenění do balíku vlastních distribuovaných nástrojů. Je však nutno zmínit, že firmy z oblasti EDA tyto předávané verze úmyslně zjednodušují a zpomalují, a nepřímo tak návrháře vyvíjející složitější obvody FPGA nutí, aby dokupovali plné verze přímo od nich.



Obr. 14 Křížové odkazy mezi jednotlivými úrovněmi návrhu

Závěr

Programovaný soubor *FPGA Advantage* umožňuje pohodlný, rychlý a spolehlivý návrh součástek FPGA. Nástroje pro popis obvodu, simulaci i syntézu podporují oba jazyky – VHDL i Verilog, pracují na všech v praxi běžně používaných hardwarových platformách a využívají všechny příslušné standardní formáty dat. Sadu nástrojů lze nakonfigurovat téměř pro každé prostředí. Přednosti tohoto programového systému vyniknou zejména při složitějších návrzích a při opakovaném použití makrobloků IP [6].



Obr. 15 Těsná vazba mezi simulací a popisem obvodu

Ing. Petr Matějka, Ing. Luboš Hradecký,
Ing. Antonín Pleštil, CSc.

LITERATURA

[1] Hlavička, J., Schmidt, J.: Programovatelné logické obvody, Sdělovací technika 48 (2000), č. 1, str. 3

[2] Schmidt, J.: Architektura programovatelných logických obvodů, Sdělovací technika 48 (2000), č. 2, str. 16

[3] Hradecký, L., Bečvář, M., Slavík, P.: Návrh zákaznických systémů na bázi FPGA, Sdělovací technika 48 (2000), č. 3, str. 7

[4] www.mentor.com/fpga-advantage/

[5] www.asicentrum.cz/ASICs/Texty/digides.htm

[6] www.design-reuse.com

[7] Steinemann, H.-P., Mentor Graphics (Schweiz) AG: „Neue Ansätze im Design programmierbarer Logik“