

Konverze programovatelných obvodů FPGA na obvody ASIC

Úvod

Článek navazuje na příspěvky týkající se programovatelných logických obvodů zveřejněné od počátku roku ve Sdělovací technice a společně s [3] je logickým ukončením celého seriálu. Rozebírá problematiku převodu kompletně navrženého systému ověřeného v reálném zařízení pomocí programovatelných součástek do vyšších a úspornějších forem zákaznických integrovaných obvodů. V souladu s běžně používanou terminologií [2] je i v tomto článku pod zkratkou ASIC chápán zákaznický integrovaný obvod klasického typu, jehož funkce se „programuje“ pomocí masek při výrobních operacích.

Hlavní výhody a charakteristiky konverze

Při větších požadovaných množstvích digitálních zákaznických integrovaných obvodů není ekonomické realizovat celou výrobní sérii pomocí programovatelných součástek. V řadě případů je mnohem výhodnější převést již hotový návrh do podoby masek, které slouží pro personifikaci zákaznických čipů při závěrečných výrobních operacích. Zákazník dostává do rukou integrovaný obvod, který má funkční vlastnosti stejné jako prototyp odzkoušený pomocí obvodu FPGA nebo CPLD, vyznačuje se však lepšími technickými i ekonomickými parametry. V dalším textu budeme slovem konverze rozumět právě konverzi navrženého obvodu FPGA do obvodu ASIC a dále popsat charakteristiky jsou porovnáním řešení pouze s FPGA nebo pouze s ASIC.

Rychlost a bezpečnost

Pomineme-li ekonomické faktory, konverze přináší jistotu toho, že do výroby se dostávají podklady, které byly nejen ověřeny pomocí simulace, ale také odzkoušeny v reálném prostředí zákazníka. Při současném stavu integrace, kdy se úroveň integrace posouvá k celým systémům na čipu (SoC, [2]), přináší použití obvodů FPGA při návrhu podstatné zvýšení efektivity vývojových prací, protože modifikace obvodu, ať už z důvodu změny specifikace nebo chyby v průběhu návrhu, je relativně rychlá a levná.

Zde se nabízí otázka, proč nenavrhnout ASIC přímo a obejít proces konverze? Důvodů je několik:

- zákazník předem nemusí vědět, jak velkou sérii bude potřebovat,
- v průběhu návrhu ASIC lze obvod pouze simulovat a není testován v reálném prostředí,
- postup prací při návrhu ASIC je složitější,
- vyžaduje se těsná spolupráce s výrobcem obvodů,
- riziko technologické obrátky je vysoké.

Univerzálnost procesu

V současnosti lze konverzi provést ze všech dostupných obvodů FPGA nabízených hlavními výrobci, jako jsou Xilinx, Altera, Atmel, Lattice, Actel a další. Omezení může nastat pouze u výrobce obvodu ASIC, který by nepodporoval některé typy programovatelných součástek nebo v dostupnosti příslušných návrhových prostředků.

Nízké jednorázové náklady

Značná část nákladů na konverzi je jednorázová a pokrývá výrobu masek. Z tohoto dů-

vanou sérii, velikost pouzdra atd. Typická doba celé konverze (asi 10 týdnů) zahrnuje procesy od převodu dat a jejich přípravu pro technologické operace, včetně generace masek, až po výrobu funkčních vzorků obvodu ASIC.

Lepší zabezpečení IP

Dalším, nezanedbatelným důvodem pro konverzi může být podstatné ztižení možnosti kopírování obvodu. Obvody FPGA jsou v podstatě běžně dostupné součástky, kterým „život“ vdechne připojení externí konfigurační paměti, která obsahuje vlastní propojení uvnitř obvodu. I když existují možnosti, které ztíží kopírování aplikace s FPGA, uspokojivé řešení se nedá najít, pokud v průběhu konfigurace FPGA musí být přiveden obsah konfigurační paměti na vývod obvodu. Právě tato kritická fáze u obvodů ASIC odpadá a možnost kopírování je v podstatě minimální, ne-li nulová.

Zlepšení technických parametrů

Obvody ASIC mohou pracovat s vyšší frekvencí než FPGA z toho důvodu, že propojení logických buněk je realizováno přímými metalickými spoji na rozdíl od konfigurovatelných spojů u FPGA. Obecně lze také říci, že obvody ASIC přinášejí snížení spotřeby obvodů, protože ASIC obsahuje pouze logiku pro vykonávání vlastní funkce. I když nebývá, např. ani u klasických hradlových polí, plocha využitá stoprocentně, nezaplněné buňky (na rozdíl od obvodů FPGA) nezvyšují spotřebu. Spotřeba obvodu je dána také velikostí napájecího napětí. Současná technologie podporují napájecí napětí v rozsahu 2,5–5 V.

Častým důvodem, proč se rozhodnout pro konverzi, bývá také integrace speciálních buněk, které obecně nemusí obsahovat obvod FPGA. Typickým příkladem může být krystalový oscilátor s režimem snížené spotřeby, kdy pro jeho realizaci se z vnějšího obvodu připojí pouze krystal, nebo rezonátor a pomocné kapacity.

Při současné velikosti obvodů FPGA jsou běžné aplikace, kdy se v FPGA realizuje procesor provádějící algoritmus, který většinou vyžaduje určitou kapacitu operační paměti. Příkladem takových aplikací může být např. FFT (Fast Fourier Transformation) nebo kterýkoliv z kompresních algoritmů. Pokud je tato operační paměť řádu desítek nebo stovek bajtů, lze ji v celku bez problémů realizovat uvnitř FPGA. Větší kapacity lze realizovat pouze v nejmodernějších rodinách FPGA

Tabulka 1 Praktické příklady konverze

aplikace DSP pro telekomunikace – velikost asi 22 000 logických hradel – frekvence 18 MHz – paměti ROM a RAM na čipu	
prototypová verze programovatelné hradlové pole Xilinx XC4028XL – technologie 0,35 μm – napájení 3,3 V – pouzdro HQFP160 konfigurační paměť 1× XC1701	verze pro výrobu maskami programované hradlové pole ORBIT 0,5 μm GA Base 100 – technologie 0,5 μm – napájení 3–5,5 V – pouzdro PLCC44 – paměť ROM a RAM na čipu
aplikace pro telekomunikace – obvod pro přizpůsobení dat. terminálu na síť ISDN – velikost asi 16 000 logických hradel – maximální frekvence 10 MHz – paměť RAM na čipu	
prototypová verze programovatelné hradlové pole Xilinx XCS40 – technologie 0,5 μm/0,35 μm – napájení 5 V – pouzdro PQFP208 konfigurační paměť 2× XC17C256	verze pro výrobu maskami programované hradlové pole ORBIT 0,5 μm GA Base 100 – technologie 0,5 μm – napájení 3 až 5,5 V – pouzdro PLCC44, MQFP44 – paměť RAM na čipu

vodu se výrobci obvodů snaží použít co nejmeně masek. Konverze se tedy (v závislosti na velikosti série a technických požadavcích zákazníka) většinou neprovádí do plně zákaznického obvodu nebo standardních buněk, ale na hradlové pole, u kterého je pro závěrečné výrobní operace nutné vyrobit jenom minimální množství masek.

Ekonomická výhodnost

Obecně lze říci, že konverze může být ekonomicky výhodná již od série několika tisíc obvodů, v některých případech se může vyplatit i u kusových sérií objemu několika set obvodů. Základní ekonomické úvahy je nutné provést pro konkrétní typ součástky, požado-

(Xilinx – Virtex, Altera – Apex). Tady konverze nabízí řešení, kdy výsledný čip integruje FPGA i paměť.

Některé aplikace mohou vyžadovat obvod v pouzdru, které výrobce FPGA nenabízí. Typicky, složité obvody FPGA obsahují velké množství vývodů, které nejsou vždy použity. Zákaznický čip naopak může být umístěn v libovolném pouzdru, pokud není omezen jeho velikostí. Navíc rozložení vývodů může být libovolné a to včetně napájecích vývodů. Zde se rovněž nabízí možnost, že se výsledný otestovaný čip nepouzdří, ale použije se na desce s plošnými spoji přímo (tzv. technologie Chip On Board nebo Flip Chip).

Specifika návrhu pro konverzi

S procesem konverze je vhodné počítat již před fází návrhu vlastního FPGA. Nejdůležitějším důvodem je skutečnost, že finální obvod ASIC je nutné testovat, což většinou při vývoji obvodu FPGA odpadá. V případě složitějších obvodů s více časovými doménami nemusí být generování testovacích vektorů snadné a vyžaduje promyšlenou strategii. Protože testovací vektory se aplikují v konstantních časových intervalech, je nutné jednotlivé hodinové vstupy do obvodu svázat do pevného vztahu, nejrychlejší z těchto hodinových signálů pak určuje testovací frekvenci obvodu. Rovněž je nutné zabezpečit, aby data vstupující do obvodu měla správný předstih a přesah vzhledem k příslušnému hodinovému signálu. Obecně lze doporučit postup návrhu popsaný v [1] nebo [2].

Postup při procesu konverze

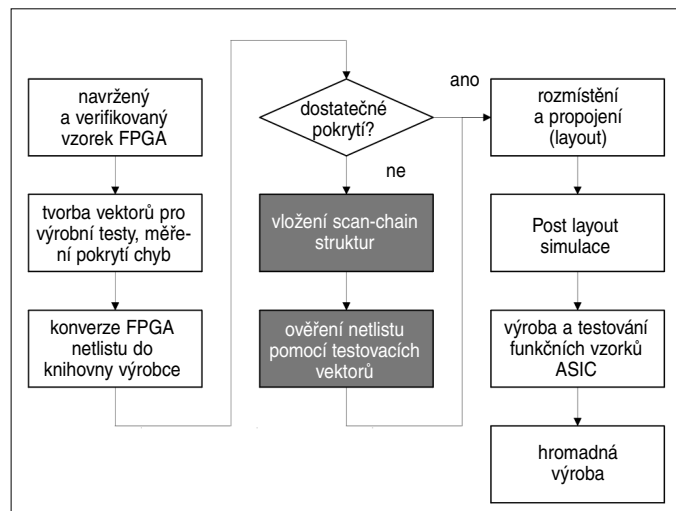
Postup konverze je uveden na obr. 1. Konverze začíná ve chvíli, kdy byl návrh ověřen pomocí FPGA.

Styčným bodem při ověřování funkce mezi zákazníkem, návrhářským střediskem a výrobcem obvodů jsou testy. Slouží k ověření funkce obvodu podle specifikace zákazníka (tzv. funkční testy) a zároveň k zachycení technologických defektů při výrobě obvodu. Posouzení, nakolik je možné pomocí testů tyto defekty odhalit, udává pokrytí. Aby měl zákazník dostatečně velkou jistotu, že drtivá většina chyb bude zachycena testovacími vektory, je nutno dosáhnout pokrytí minimálně 85 % jejich celkového počtu při daném modelu defektu.

Výrobní testy musí být přizpůsobeny testovacímu zařízení, které bude v průběhu výroby použito. Test se skládá z testovacích vektorů, které se postupně přikládají na vstupy obvodu, přičemž se kontroluje správnost odezvy. Generování testovacích vektorů se provádí v simulátoru HDL pomocí bloku, který v průběhu simulace v konstantních časových intervalech sleduje všechny vývody testovaného obvodu a ukládá je do souboru.

Testovací vektory lze generovat, kromě funkčních testů, i pomocí automatických nástrojů ATPG případně dalších praktik uvedených v [3].

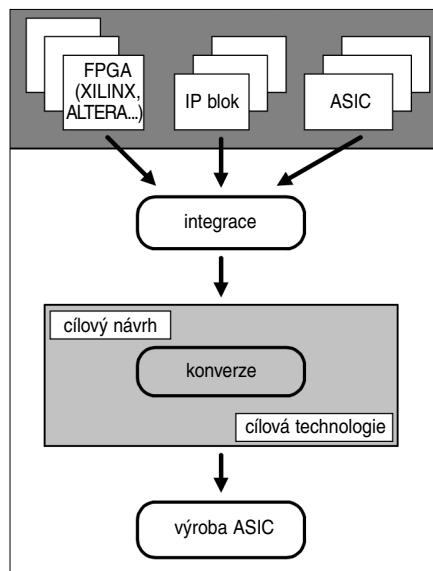
Vlastní zapojení obvodu FPGA na úrovni hradel představuje netlist, který je pomocí konverzního programu převeden do knihovny výrobce obvodu ASIC. Po konverzi netlistu se měří pokrytí defektů testy. Pokud je



Obr. 1 Postup konverze FPGA na ASIC

dostatečné, lze provést další operace, zahrnující standardní postup při návrhu ASIC.

V případě, že není pokrytí dostatečné, je nutné přistoupit k modifikaci netlistu vložením speciálních struktur usnadňujících zvýšení pokrytí. Značného rozšíření zaznamenaly techniky BIST pro testování zabudovaných RAM, dále JTAG, neboli boundary scan, a především pak vložení struktur scan chain.



Obr. 2 Konverze z několika zdrojů

Tato metoda představuje účinnou techniku pro zvýšení pokrytí blízké hodnotě 100 %.

Obecně lze metodu scan-chain popsat následovně. Klopné obvody D se nahradí typem, který obsahuje dva datové vstupy. První slouží pro běžnou funkci obvodu, pomocí druhého se klopné obvody propojují do řetězce, který se chová jako posuvný registr. Obvod pak může pracovat v normálním režimu,

nebo při testování v režimu scan. Testovací vektory v režimu scan-chain jsou generovány automatickým nástrojem. Test se skládá z fáze naplňování posuvného registru, pak se na dobu jedné periody hodinového signálu přepne do normálního režimu, kdy se obsahy klopných obvodů naplní hodnotou danou kombinační logikou, pak se obvod opět přepne do režimu scan a posuvný registr se načte. Test pomocí scan-chain netestuje správnost funkce obvodu, pouze ověřuje zda je daný obvod shodný s netlistem. Vložení struktur scan-chain se provádí automatickým nástrojem a znamená značný zásah do netlistu. Z tohoto důvodu se netlist po vložení scan-chain opět verifikuje pomocí funkčních testů. Dalším krokem je rozmístění a propojení netlistu s následnou kontrolou dodržení pravidel návrhu (DRC). Nyní se provede tzv. simulace post-layout, která již zahrnuje reálná zpoždění v obvodu. Následuje výroba masek a zkušební série obvodů. Po ověření těchto funkčních vzorků u zákazníka následuje hromadná výroba uzavírající celý proces.

Pokročilé techniky konverze

Konverzi do jednoho finálního ASIC je možné provést obecně z více stávajících návrhů – obr. 2. Příkladem je navrhování obvodů s použitím maker IP [2] nebo integrace dříve navržených obvodů v FPGA nebo ASIC. Důležitým krokem je vytvoření správných rozhraní mezi bloky, jejich integrace do jediného celku a hlavně důkladná verifikace celého obvodu. Praktické příklady konverze jsou uvedeny v tabulce 1.

Závěr

Jednotlivé etapy v procesu konverze FPGA na ASIC jsou v mnohém podobné etapám při návrhu zákaznických integrovaných obvodů. Byly již dříve podrobně popsány např. ve [2] nebo [4]. Tamtéž lze nalézt i příklady vhodných programových prostředků. Při provádění konverze však existují operace, které jsou pro tuto činnost specifické a taktéž vyžadují speciální vývojové prostředky. Nezbytná je maximální podpora ze strany výrobce ASIC, dostupnost kvalitních návrhových prostředků od renomovaných firem [4] nebo od výrobců obvodů FPGA, ale především zkušenosti návrhářů.

Ing. Luboš Hradecký

Ing. Juraj Priskin

LITERATURA

- [1] Keating M., Bricaud P.: *Reuse Methodology Manual*, Kluwer Academic Publishers, 1998
- [2] Bečvář J., Slavík P.: *Metody návrhu systémů na bázi FPGA*, ST 2000, č. 3, str. 7–11
- [3] Hlavička, J.: *Testování programovatelných hradlových polí*, ST 2000, č. 6, str. 3–6
- [4] Matějka, P., Hradecký, L., Pleštil, A.: *Moderní postupy při návrhu programovatelných logických obvodů*, ST 2000, č. 4, str. 27–30