

## Vývoj zařízení pro FPGA na konceptuální úrovni

R. Schönecker, L. Lorenc, Z. Křivka

Ústav informačních systémů, FIT, VUT v Brně

Při výstavbě hardware-software systémů je kladen obecně maximální důraz na vytvoření efektivních a výkonných zařízení, nicméně v reálném konkurenčním prostředí se ukazuje, že prioritou je především samotná rychlost vývoje a flexibilita návrhu. Konkurenční boj nutí výrobce a tým i vývojáře zařízení k neustálému zkracování doby vývoje a přípravy výroby nového zařízení. Pro úspěšnost na trhu (především v takzvané low-end kategorii) přestává být důležitá kvalita samotného výrobku, úspěšnější bývá ten, kdo nové zařízení dodá jako první.

Vysoké tempo vývoje však často bývá doprovázeno vznikem nejružnějších chyb a nedostatků, které není možné odhalit v krátkém čase vyhrazeném pro testování. Řada chyb či nedostatků tak bývá odhalena až v reálném provozu a pokud nechce společnost přijít o své zákazníky, musí existovat jednoduchá cesta, jak uživatelům již prodaných zařízení umožnit rychlou automatickou opravu software či rekonfiguraci hardware.

Tyto dva protichůdné požadavky na rychlost vývoje a dostatečnou obecnost návrhu sehrály již dříve podstatnou roli při vzniku komponentního modelu vývoje a tzv. modelem řízených architektur v oblasti software (viz [1]).

Stále více se však ukazuje, že podobný návrh je výhodný i v oblasti vývoje hardware či hardware v kombinaci s řídicím software.

Opakované využívání předem připravených a otestovaných komponent umožňuje soustředit většinu úsilí při vývoji nového zařízení pouze na propojení těchto komponent do funkčního celku. Tím dojde ke značnému zkrácení doby vývoje a splnění prvního požadavku.

Důležitý je však i druhý požadavek na možnost pozdějších automatických úprav zařízení. Takovéto úpravy je možné realizovat dostatečně obecným návrhem zařízení a jeho doplněním o vhodné rozhraní a řízení pro automatické nahrání požadovaných informací.

Jedním ze způsobů, jak lze efektivně upravovat jak řídicí program, tak i vnitřní logiku (hardwarové obvody) takového zařízení je využití kombinace mikrokontroléru a FPGA. V tomto případě je jedinou pevnou softwarovou součástí takového zařízení pouze takzvaný bootloader, což je poměrně krátký kód ve vnitřní paměti mikrokontroléru. Tento bootloader provádí dvě různé činnosti. Za normální činnosti po spuštění zařízení nahraje do mikrokontroléru pracovní program a předá mu řízení. V okamžiku, kdy je potřeba provést

aktualizaci řídicího programu, provede bootloader natažení jiného, aktualizacího programu z připojeného vnějšího zařízení. Tímto zařízením může být USB flash, USB master v počítači a podobně. Jelikož je aktualizacího program zaváděn z externího zařízení, lze pomocí něj opravit téměř jakoukoliv chybu s výjimkou chyby v bootloadeu zabraňující spuštění aktualizace. Po spuštění může aktualizacího program nejen modifikovat kód nahrávaný do mikrokontroléru pro běžnou činnost, ale i paměť obsahující bitstream nahrávaný do FPGA a tím účinně modifikovat i vlastní hardwarové uspořádání tohoto zařízení.

Možnost aktualizace řídicího programu v mikrokontroléru je poměrně stará a rozsáhle využívaná záležitost. Ve spojení s FPGA se však dostáváme do úplně jiné oblasti, kdy má výrobce možnost modifikovat nejen „program“ v zařízení, ale i jeho vlastní obvodové zapojení a tím získává nesrovnatelně vyšší flexibilitu a možnosti rekonfigurace.

Zde nastává zlomový okamžik, kdy všechny požadavky na hardware, které byly nezbytné z hlediska zavedení nové technologie vývoje, jsou splněny.

Prvním krokem změny je přenesení těžiště popisu zařízení směrem k modelu vzájemně propojených komponent, kdy vývojář již operuje v pojmech vlastností komponent, metod a událostí tak, jak je to již delší dobu běžné při vývoji software v RAD studiích, zcela se tak mění koncept přístupu k vývoji a rychlost, s níž je možné vytvořit návrh funkčního zařízení.

Již dnes existují aplikace, které zapouzdřují současně softwarové a pomyslné hardwarové komponenty do reálných, s nimiž lze na konceptuální úrovni pohodlně pracovat (např. nástroj Processor Expert fy Unis, viz [2]). V aplikaci tak lze použít unifikovaný přístup ke komponentám představujícím softwarové algoritmy stejně jako ke komponentám reprezentujícím hardwarové prvky – např. mikroprocesor, paměti, periférie.

Dosud však byla sféra použití těchto aplikací omezena mimo oblast FPGA.

Nový koncept vývoje, který v tomto článku chceme představit, je založen na propojení nástrojů pracujících s komponentami na konceptuální úrovni, a nástrojů, které zastřešují vývoj pro FPGA zařízení (zde konkrétně se jedná o nástroje Xilinx EDK, viz [3]), a umožnění vývoje pro FPGA na konceptuální úrovni s využitím komponentního návrhu a událostmi řízeného programování.

### Reference:

- [1] Sommerville, I.: Software engineering, 2004
- [2] <http://www.processorexpert.com/>
- [3] <http://www.xilinx.com/>