

a organizační záležitosti, výši podílu na nákladech a ziscích a vlastnické vztahy k výsledkům projektu pro jednotlivé zúčastněné subjekty. „Pokud se nám podaří předem vypořádat veškeré vztahy ke spokojenosti všech stran, mohou být nakonec všichni vítězi. Společné know-how lze rozvíjet v dalších grantech či licencovat, prototyp dotáhnout do formy produktu, ten prodat a získané prostředky rozdělit podle smlouvy mezi partnery. Možností je víc,“ uvádí Radek Fiala.

Největší překážkou bezproblémového vzniku spin-off firem je v této chvíli absence zažitých pravidel, která by přehledně a rychle uspořádala vzájemné vztahy. CleverTechnologies jako spin-off firma mohla být vytvořena jen díky osobní podpoře konkrétních osob na vedoucích místech obou univerzit, v jejichž společném náručí celý projekt vznikl. Za FBMI patří největší dík paní děkance prof. Vrbové a panu prof. Knappovi, z 1. lékařské fakulty nejvíce pomohl vizionářský přístup pana prof. Zimy (děkan 1. LF UK). V současné době se již na obzoru rýsují zajímavé zakázky, jejichž výsledky ale ve firmě CleverTechnologies zatím nechtějí předvídat. Vždy zde však budou další studenti FBMI ČVUT a 1. LF UK, pro něž je možnost uplatnění vlastních nápadů v praxi tím pravým lákadlem, které je přivádí ke dveřím Společného pracoviště FBMI ČVUT a UK. Pokud tedy hledáte úspěšný model spin-off firmy v českém prostředí, neuděláte chybu, když se také vydáte na Albertov. Je to po všech stránkách příjemná procházka.

Ing. Michal Brnušák

## Programovatelná hradlová pole

*Článek rozebírá základní vlastnosti a užítí programovatelných hradlových polí a ve stručnosti popisuje jejich možné aplikace. Obvodů FPGA je dnes na trhu dostupné velké množství v široké škále variant od mnoha výrobců. Cílem článku je seznámit čtenáře s obecnými principy těchto prvků, na trhu dostupnými moderními variantami a poskytnout základní vodítko pro výběr správného obvodu.*

### 1 Úvod

Programovatelná hradlová pole (FPGA – *Field Programmable Gate Array*) jsou speciální integrované obvody obsahující různé složité programovatelné bloky propojené konfigurovatelnou maticí spojů (*obr. 1*). Termín *Field Programmable* je to, čím se obvody FPGA (dále jen FPGA) odlišují od zákaznických integrovaných obvodů. Obvod je „perzonifikován“ (nakonfigurován) u zákazníka v laboratoři či ve finální aplikaci (*field* – „v poli“, *in-circuit programming*), nikoliv při výrobě.

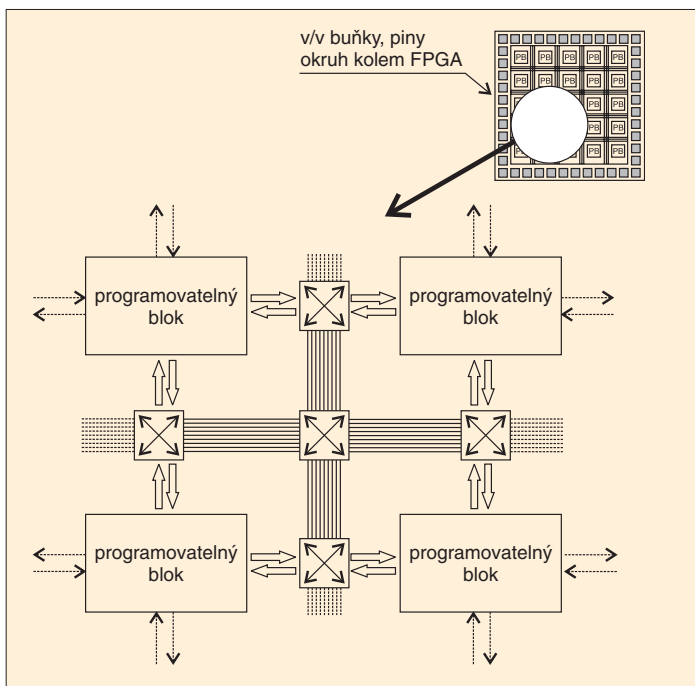
Obvody FPGA dnes nacházejí uplatnění v široké škále aplikací díky své programovatelnosti, snadnému návrhu, flexibilitě, neustále klesajícím cenám a zvolna se snižující spotřebě energie vlastním čipem. Typické

použití je v oblasti menších sérií navrhovaných zařízení (řekněme řádově pod 10 000 kusů), kdy se nevyplatí návrh zákaznického integrovaného obvodu a současně konvenční řešení systému s procesorem už není vhodné. Další aplikace můžeme nalézt například v oblasti prototypování složitějších návrhů zákaznických integrovaných obvodů, kdy nám FPGA umožňuje testovat navrhovaný systém už během návrhu, ještě před finančně náročnou výrobou integrovaného obvodu. Programovatelnost je sama o sobě ohromnou výhodou – umožňuje totiž (pokud dobře navrheme zařízení a vybereme FPGA) měnit chování systému u zákazníka, a tak například opravovat chyby návrhu stažením aktualizované verze (*update*) z webu či přidat nové funkce (ať už zdarma či po dokončení licence uživatelem).

Kromě obvodů FPGA se lze běžně setkat i s obvody CPLD. Přestože se dnes rozdíl mezi oběma skupinami postupně stírají, je mezi obvody FPGA a CPLD stále několik významných rozdílů: zatímco obvody FPGA jsou vhodné pro velké systémy s komplexními algoritmy a často obsahují složité funkční bloky (násobičky, jednotky MAC atd.), CPLD použijeme spíše na návrh jednoduché propojovací logiky při integraci složitějších prvků (*glue logic*). Jsme omezeni malým počtem interních registrů, absencí paměti a dalších integrovaných bloků. V dalším textu se budeme zabývat především obvody FPGA.

### 2 Historie

Historické kořeny moderních programovatelných polí je třeba hledat v prvních programovatelných pamětech typu PROM (firma Radiation, 1970) a jejich zákaznických verzích EPROM (Intel, 1971) a EEPROM (Intel, 1978). Permanentní paměti



Obr. 1 Schematické znázornění základní architektury obvodu FPGA

### LITERATURA

- [1] ŠMEJKAL, L., *Co mi na ČVUT neřekli*. Automatizace 50 (2007), č. 5, str. 378.

jako takové ale neumožňovaly nikdy příliš úspornou realizaci logické funkce, protože realizované logické funkce se jen zřídka skládají z většího množství termů, zatímco permanentní paměť obsahuje kompletní adresový dekodér. Logickým vývojovým krokem proto byl vznik obvodů FPLA (Signetics/Philips, 1970) obsahujících dvě programovatelné matice pro realizaci součinných a součtových termů v kaskádě. Obvody FPLA byly velmi drahé a současně pomalé díky dvěma programovatelným maticím za sebou. Aby bylo možné dosáhnout vyšší rychlosti logiky, byla součtová matice realizována jako fixní. Vznikla koncepce obvodů PAL (Monolithic Memories, 1978) s programovatelnou maticí AND a pevnou maticí OR. Obvody PAL byly ve své době velmi úspěšné, umožňovaly dosáhnout zhruba 50× vyšší hustoty logických funkcí než u tehdy používaných logických obvodů.

S postupně se zlepšujícími technologiemi výroby integrovaných obvodů bylo možné vyrábět programovatelné obvody s vyšší kapacitou a realizující složitější logické funkce. Zde výrobci PLD naráželi na další obtíž – s rostoucím množstvím nezávislých proměnných v realizovatelných logických funkcích prudce rostla velikost programovatelného pole. Začalo se proto místo rozšiřování logických funkcí užívat spíše skládání více matic obvodů PLD do jednoho pouzdra.

Vznikly tak obvody, které dnes nazýváme CPLD (Altera, 1988).

Od CPLD byl už pak jen malý krok k prvním obvodům FPGA (Xilinx, 1984). Dnes dostupné FPGA se ovšem od architektur z poloviny 80. let výrazně odlišují. Trendem je pozvolný příklon k hrubozrným architekturám; obvodům, které kromě elementárních programovatelných logických bloků obsahují i další komplexní podpůrné bloky. Zvolna se také objevují nové architektury postavené na principu kombinace procesoru a programovatelného hradlového pole (například MathStar FPOA) – jeden čip s polem procesorů a polem programovatelných objektů [1].

### 3 Různé úrovně programovatelnosti

Protože budeme hovořit o programovatelných systémech, je vhodné rozebrat s úrovně programovatelnosti, s nimiž pracujeme.

Standardem pro dnešní výpočetní techniku je *systém programovatelný na úrovni strojového kódu*. Vyšší programovací jazyky (C, C++, Pascal apod.) se do strojového kódu překládají. Strojový kód je optimalizován pro konkrétní architekturu procesoru, jeho instrukce reprezentují jednotlivé operace. Vykonání jedné operace může trvat jednotky i více strojových cyklů a jedinou instrukcí lze realizovat i složité operace. Operace mohou být v omezené míře prováděny paralelně.

Na obr. 2 jsou znázorněny základní možnosti realizace stavového automatu pomocí logických prvků – *systém programovatelný na úrovni mikrokódu*. Obsah paměti – pravdivostní tabulka přechodů a výstupů automatu – určuje chování automatu a můžeme na něj pohlížet jako na program stavového stroje, přesněji *mikroprogram*. Funkcí takového mikroprogramu může být například interpretace strojových instrukcí procesoru (mikroprogram v řadiči CPU). Procesor, kterým lze změnit mikroprogram řadiče, tak umožňuje například změnit chování svých instrukcí, případně přidat nové operace apod. Mikroinstrukce (instrukce mikroprogramu) se vykonávají celkem rychle (v naznačeném případě jedna instrukce za jeden hodinový cyklus), ale umožňují kódovat jen jednoduché operace. Operace jsou prováděny sériově. Jednoduchý příklad mikroprogramovatelného řadiče je uveden například v [2].

Přímé řízení propojení jednotlivých funkčních bloků, konfiguraci logických prvků umožňuje *systém programovatelný na úrovni hardwaru*. Těžko lze hovořit o tom, jak rychle jsou prováděny instrukce či jak je jejich vykonávání paralelizováno. Spíše je nutné zhodnotit parametry výsledné naprogramované struktury. Je možné získat vysoce paralelní systém s hodinovým cyklem řádu několika set megahertz [MHz]. Obvody FPGA, které jsou tématem tohoto článku, jsou typickými představiteli tohoto systému.

*Neprogramovatelný systém* je zákaznický obvod realizující danou a předem stanovenou množinu funkcí. Touto kategorií se dále nebudeme zabývat.

### 4 Dostupné technologie

V tabulce lze nalézt základní srovnání jednotlivých technologií návrhu. Technologické dělení je hrubé. Nerozlišujeme mezi plně zákaznickým systémem ASIC, použitím standardních buněk, nebo hradlového pole a zcela opomíjíme oblast analogových integrovaných obvodů, ale pro naše účely je naprosto postačující. Do souhrnu nezahrnujeme obvody typu PAL/GAL – jsou příliš malé a pro velké digitální obvody nepoužitelné. Structured ASIC jsou moderní platformy, hybrid mezi zákaznickým integrovaným obvodem navrženým pomocí standardních buněk a FPGA. Technologie v mnohém připomíná starší hradlová pole typu „*sea of gates*“: výrobce vyrobí křemíkové plátky s připravenými základními bloky – hradly, registry i složitějšími komponentami. Připravené plátky jsou potom perzonalizovány jedním či více metalami – vrstvami se spojí propojujícími bloky. Perzonalizační metalové masky jsou dodány zákazníkem. Jsou výsledkem jeho konkrétního návrhového procesu.

### 5 Proč FPGA

#### 5.1 Implementace aplikace

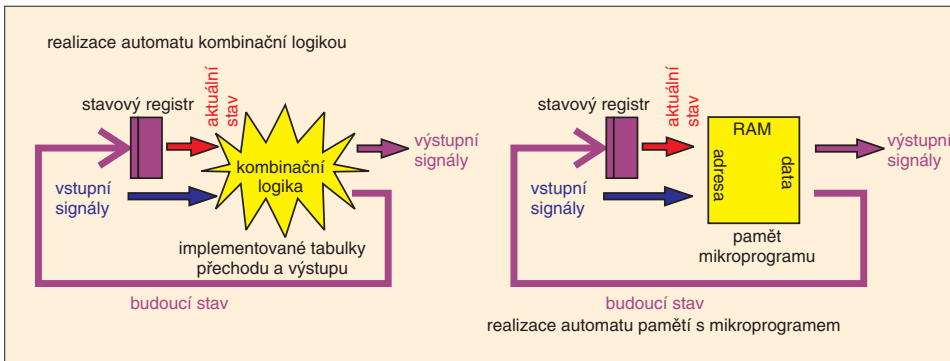
Systémy s FPGA zažívají v posledních letech nebyvalý boom, pronikají do stále rostoucí množiny aplikací. To je dáno mnoha jejich příjavnými vlastnostmi. Srovnáme použití klasického procesoru a obvodu FPGA z pohledu návrhu finální aplikace.

CPU je víceméně klasické programování (typicky stále v assembleru). Výkon systému hodnotíme v MIPS (miliony instrukcí za sekundu).

Návrh FPGA je návrhem hardwaru i přes metodiky a nástroje umožňující návrh FPGA pomocí vyšších programovacích jazyků (C, System C atd.). FPGA umožňuje silnou paralelizaci, a tak zde nemá smysl měřit výkon v MIPS. Výrobci udávají teoretický maximální výkon v MMACs. Toto číslo je často nadsazené a nesmyslné (často získáno jako počet násobiček na FPGA × maximální hodinová frekvence násobičky), přesto ale FPGA systémy dosáhnou minimálně o řád vyššího výkonu než nejrychlejší DSP CPU. V případě, že je FPGA dražší než procesor, je třeba hodnotit i poměr cena/výkon – některé aplikace (např. zpracování digitálního rádia) nejsou na DSP CPU vůbec implementovatelné, ale obvody FPGA je dnes běžně realizují. Vlastní princip obvodů FPGA umožňuje spojit výhody realizace systému zákaznickým integrovaným obvodem s možností rychlého testování výsledného produktu, nízkými počátečními náklady a jednoduchou změnou vlastností.

#### POUŽITÉ ZKRATKY

ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal-Oxide Semiconductor
CPLD	Complex Programmable Logic Devices
CPU	Central Processing Unit
DSP	Digital Signal Processing
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EPROM	Electrically-Programmable Read-Only Memory
FPGA	Field Programmable Gate Array
FPLA	Field Programmable Logic Array
FPOA	Field Programmable Object Array
GAL	Generic Array Logic
HDL	Hardware Description Language
HLS	High-Level Synthesis
v/v	vstupně/výstupní
IP	Intellectual Property
LUT	Look-Up Table
MIPS	Million Instructions Per Second
MMAC	Million Multiply and ACcumulate operations per second
PAL	Programmable Array Logic
RTL	Register Transfer Level
SRAM	Static Random Access Memory
TMR	Triple Module Redundant



Obr. 2 Možnosti realizace sekvenčního logického obvodu (kombinační logika nebo paměť mikroprogramu)

## 5.2 Systémový návrh aplikace

U CPU jsou počet funkčních jednotek a maximální dosažitelná hodinová frekvence systému určeny výrobem procesoru. Výkon procesoru přitom nelze přidáváním dalších výpočetních jednotek (např. další jednotkou ALU, MAC apod.) libovolně zvyšovat. S každou novou jednotkou totiž vzrostou kapacitní zátěže sběrnic v procesoru, a tak se sníží maximální dosažitelná hodinová frekvence. Navíc není nijak zajištěno, že kompilátor skutečně dokáže všechny jednotky optimálně vytížit – plánování operací samo je algoritmicky obtížně řešitelná úloha. Naopak, procesor může mít jednotky, které konkrétní aplikace nikdy nevyužije. Pokud cílová aplikace používá

např. netradiční aritmetické operace může návrhář aplikace narazit na problémy s jejich nedostatečnou podporou zvoleným procesorem.

FPGA umožňuje volit implementaci systému, což poskytuje výrazně vyšší flexibilitu. Struktura hardwaru a stupeň paralelizace mohou být zvoleny s ohledem na charakter řešeného problému. Jediným omezením je pak míra paralelizace, jakou implementovaný algoritmus umožňuje. Hardware je ušit na míru problému. V případě realizace systému pomocí klasického procesoru volíme pouze rychlost procesoru podle složitosti (výpočetní náročnosti) implementovaného algoritmu. Pokud navrhujeme na obvodu FPGA, můžeme pracovat i s velikostí obvodu (stupněm

paralelizovatelnosti výpočtu; větší plocha = více paralelních jednotek). Návrh na FPGA umožňuje i další jemné optimalizace – pokud např. navrhovaný systém DSP potřebuje alespoň sedmnáctibitové slovo, aby si zachoval minimální odstup signál-kvantovací šum, může být bez problémů navržen hardware pro práci se sedmnáctibitovými vzorky. Je-li třeba pracovat s nezvyklou aritmetikou (např. Montgomeryho doména v kryptografii, česká soustava, apod.), lze navrhnout speciální dedikovaný hardware s vysokým výpočetním výkonem.

## 5.3 Integrace funkcí

Na dostatečně velkém FPGA lze soustředit více funkcí, které by jinak zabraly několik pouzder a nezanedbatelné místo na desce s tištěnými spoji. Systém se tak přes užití dražšího FPGA zlevní. Do jednoho čipu můžeme například soustředit několik jader procesorů.

## 5.4 Shrnutí

Shrme tedy, za jakých podmínek pro nás může být použití FPGA výhodné:

- Pro aplikace, které vyžadují zpracování velkých datových toků, pro výpočetně náročné algoritmy (multimédia – audio/video, 3G síť, obecně číslicové zpracování signálu, kryptografie), kde nestačí klasické procesorové řešení

## Technologie návrhu obvodů

Vlastnosti	Technologie ASIC	Structured ASIC	FPGA	CPLD	DSP CPU	Obecné CPU
Charakter návrhu	Všechny druhy návrhu – asynchronní, synchronní, low power, analogové obvody atd.	Všechny druhy digitálního návrhu. Časové vlastnosti a spotřeba návrhu jsou o něco horší než u obvodů ASIC.	Plně synchronní návrh, nedoporučuje se používat asynchronních technik návrhu a je třeba pečlivě řešit případné přepínání či hradlování hodin v obvodu.	Jednoduché logické funkce, logika umožňující integraci komponent (tzv. glue logic).	Lze realizovat vše, co lze naprogramovat. Vlastnosti systému jsou určeny především architekturou použité CPU. Procesory DSP jsou určeny pro signálové aplikace, mají speciální funkční jednotky pro zjednodušení a zrychlení příslušných typů výpočtů (adresní generátory, jednotky MAC apod.).	Lze realizovat vše, co lze naprogramovat. Vlastnosti systému jsou určeny především architekturou použité CPU.
Cena návrhu	Drahý návrh, vysoké fixní náklady	Levnější než ASIC a dražší než FPGA.	Nízké fixní náklady	Nízké, ale také malé obvody.	Nejlevnější návrh.	Nejlevnější návrh.
Vhodná velikost série	Levná výroba ve velkém (stovky tisíc kusů a víc), pro malé série drahé.	Mezi ASIC a FPGA.	Drahá součástka pro velké série, pro malé levnější než řešení obvodem ASIC.	Drahá součástka pro velké série, pro malé levnější, než řešení obvodem ASIC.	Spiše pro menší série nebo tam, kde je jednoduchá programovatelnost přínosem pro cenu návrhu.	Spiše pro menší série nebo tam, kde je jednoduchá programovatelnost přínosem pro cenu návrhu.
Možnost upgradu finálního systému	Nemá-li produkt speciální podporu pro změnu funkce (vestavný procesor apod.), je změna funkce nemožná.	Nemá-li produkt speciální podporu pro změnu funkce (vestavný procesor apod.), je změna funkce nemožná.	Jednoduché, pokud je systém navržen s možností změny konfigurace	Nemá téměř smysl.	Jednoduché, pokud systém umožňuje aktualizovat program.	Jednoduché, pokud systém umožňuje aktualizovat program.
Náročnost návrhu	Profesionální pracoviště.	Profesionální pracoviště.	I méně odborně fundovaná pracoviště.	Lze bez problémů doma „na koleni“.	Zvládne desetileté dítě.	Zvládne desetileté dítě.
Dosažitelný výpočetní výkon	Mohutná paralelizace, multiplexování bloků, vysoký výpočetní výkon, maximální využití křemíku, možnost zajímavých kompromisů výpočetní výkon–plocha.	Mohutná paralelizace, multiplexování bloků, vysoký výpočetní výkon, maximální využití křemíku, možnost zajímavých kompromisů výpočetní výkon–plocha.	Mohutná paralelizace, multiplexování bloků, vysoký výpočetní výkon, maximální využití křemíku, možnost zajímavých kompromisů výpočetní výkon–plocha.	Nemá smysl.	Předem dané množství výkonných jednotek, často nemožnost všechny jednotky vytižit tokem instrukcí.	Předem dané množství výkonných jednotek, často nemožnost všechny jednotky vytižit tokem i instrukcí.
Zabezpečení duševního vlastnictví	Okopírovat návrh je velice složité, doména specializovaných pracovišť.	Okopírovat návrh je velice složité, doména specializovaných pracovišť.	Záleží na typu FPGA – technologii (SRAM nejméně, antifuse nejvíce bezpečně).	Téměř není co chránit.	Nízká bezpečnost, stačí okopírovat program.	Nízká bezpečnost, stačí okopírovat program.

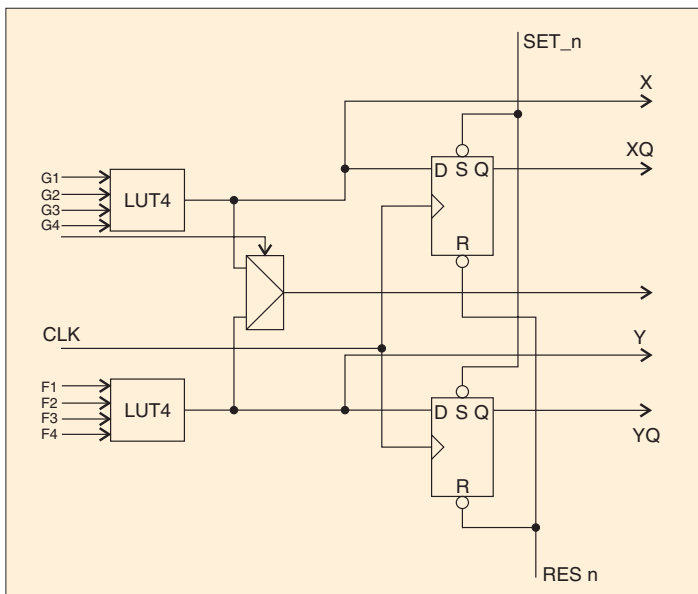


svým výkonem a plánujeme příliš malou sérii, aby se vyplatilo navrhnout rovnou ASIC.

- Pro implementaci algoritmů či protokolů, které nejsou ještě normalizovány. Programovatelnost v poli i v cílové aplikaci umožní později přeprogramovat jednou vyrobené systémy, a tak přizpůsobit už prodaná zařízení novým požadavkům. Použití FPGA se zde vyplatí i pro velké série, protože se finální aplikace dostane na trh dříve. To přináší významnou konkurenční výhodu. Později lze z FPGA udělat ASIC a vyrábět dále zařízení s nižší cenou, avšak fixní funkcí (např. po standardizaci protokolu).
- Všude, kde je třeba speciální hardware, ale kvůli velikosti série se nevyplatí ASIC. Častou aplikací FPGA je například výroba emulátorů. Typickým příkladem je trh s procesory pro vestavné (*embedded*) aplikace (*smartcards*). Takové čipy mají jen minimální nezbytné množství v/v pinů, uvnitř pouzdra je paměť PROM s uživatelskou aplikací a procesor. Návrh softwaru pro takový obvod není jednoduchý především díky omezené viditelnosti do běžícího procesoru, a tedy omezeným možnostem ladění. Obvykle je pak vytvořena speciální verze procesoru s integrovanou podporou pro ladění aplikace (instrumentovaný procesor) a s rozhraním k vývojovému systému. Takový návrh je potom implementován na FPGA, neboť emulátorů potřebujeme standardně jen několik desítek kusů na miliony vyrobených procesorů.

- FPGA se může hodit i během návrhu zákaznického integrovaného obvodu, kdy může sloužit jako platforma pro rychlé prototypování. Realizujeme-li například systém, který obsahuje několik procesorů, jež je nutno naprogramovat (spotřební elektronika – multimédia, herní konzoly), může nám FPGA prototyp významně zkrátit dobu návrhu. Software totiž může být vyladěn už během implementace vlastního návrhu ASIC – tzv. *software-hardware codesign*.

Tolik konvenční aplikace. Ve všech uvedených případech užíváme FPGA jako zákaznický integrovaný obvod – jednou ho nakonfiguruje, a pak používáme. Samotná podstata obvodů FPGA umožňuje daleko zajímavější (byť dnes stále poněkud netradiční) použití. Systémy FPGA jistě v budoucnosti najdou uplatnění v oblasti tzv. rekonfigurovatelného hardwaru. Zjednodušíme-li celou problematiku, můžeme říci, že užijeme možnost přeprogramovat FPGA jako součást funkce systému. Situaci můžeme přirovnat například k víceúlohovému operačnímu systému. Místo přepínání aplikací se rekonfiguruje hardware a zpracovává jednotlivé výpočty. Výsledkem může být úspora v situ-



Obr. 3 Zjednodušené schéma jedné buňky obvodu FPGA

acích, kdy vyžadujeme více implementovaných algoritmů a nepotřebujeme, aby jednotlivé procesy běžely současně. Získáme systém, který vyžaduje menší FPGA. Ušetříme tak za obvod. Konkrétní užití prezentuje například práce [3].

Programovatelná hradlová pole najdou brzy aplikaci i v osobních počítačích ve formě rekonfigurovatelného akceleratoru pro obecné výpočty. Rozdíly mezi návrhem hardwaru a programováním softwaru se tak postupně setrou; kompilátor vyššího programovacího jazyka sám rozhodne jakou část aplikace implementovat ve formě koprocesoru FPGA a jakou zkompileovat do strojového kódu.

Dynamická (za běhu, *on-the-fly*) rekonfigurace může být užitečná i pro zcela jiné účely. FPGA například bude hlídat své probíhající operace a detekuje-li nesprávné výsledky (hardwarový problém), překonfiguruje se tak, aby se postižené místo FPGA matice znovu nepoužilo. Velmi zjednodušená verze takového principu je používána už dnes – SRAM FPGA se v kritických aplikacích pravidelně konfigurují. Zde se tím ovšem zabraňuje tzv. měkkým chybám – změně konfigurace FPGA např. v důsledku interakce s ionizujícím zářením.

Vidíme, že aplikační možnosti FPGA jsou skutečně široké a má smysl se takovými systémy a jejich návrhem hlouběji zabývat.

## 6 Anatomie programovatelného obvodu

Struktura typického obvodu je znázorněna na obr. 1. Obvod je složen ze základních logických buněk (*cells* či *slices* – podle hierarchie) propojených konfigurovatelnými přepínači, dále obsahuje „infrastrukturu“ – vlastní spoje. Programovatelná matice (*fabric*) je obklopena konfigurovatelnými v/v buňkami zajišťujícími komunikaci s okolním světem. Zjednodušené schéma typické programovatelné

buňky potom je uvedeno na obr. 3. Vycházíme zde z logického řezu obvodů firmy Xilinx [4]. Buňka obsahuje podporu pro implementaci kombinační logické funkce realizovanou pomocí bloků LUT4 (*Look Up Table*). Jedná se zde v principu o šestnácti-bitové paměti schopné realizovat všechny logické funkce čtyř proměnných. Moderní architektury obvodů FPGA často využívají i LUT s více vstupy. Dále vidíme pomocný multiplexer, který

umožňuje spojit oba bloky LUT dohromady a realizovat úplnou funkci pěti proměnných a dva výstupní registry, které slouží pro implementaci sekvenční logiky. Architektury FPGA doslova hýří registry. Jsou to tzv. *register-rich architectures*, a proto se při návrhu logiky nemusíme bát jejich použití v téměř libovolném množství. Dodejme ještě, že skutečná architektura programovatelného elementu FPGA je často složitější. Často obsahuje integrovanou podporu pro rychlé sčítání (*carry chain*), umožňuje nakonfigurovat LUT i jako posuvný registr či RAM, výstupní registr může být jak citlivý na hranu hodin (*D flip-flop*), tak na úroveň (*D latch*).

Kromě bloků realizujících základní logické kombinační a sekvenční funkce lze na FPGA najít i řadu dalších specializovaných bloků: generátory hodin (dělení či násobení hodin, spínání hodin), paměti RAM pro ukládání většího množství dat, hardwarové násobičky či celé dedikované bloky pro aritmetické operace (vhodné pro aplikace DSP), rozhraní pro sériovou komunikaci s rychlostí v řádu gigabitů za sekundu, celá procesorová jádra a dokonce i konfigurovatelné analogové bloky, AD převodníky či paměť typu flash. Takové dodatečné funkce umožňují lepší použití FPGA v nejrůznějších cílových aplikacích – jejich použití snižuje spotřebu energie výsledného systému a zlepšuje jeho parametry.

FPGA jednotlivých výrobců, ale i produktové řady od jednoho výrobce se od sebe významně odlišují právě touto „přidanou hodnotou“. Neméně důležitým parametrem obvodu FPGA je ovšem i charakter paměti pro uložení konfigurace obvodu.

Dodejme, že obvody CPLD mají obvykle omezenější architekturu postavenou na principu realizace logické funkce ze součtových a součinových termů a významně menší množství registrů. Důsledkem architektu-

ry je na jednu stranu snižená flexibilita a významně menší kapacita obvodu, ale na druhou stranu dobře predikovatelná zpoždění v logice.

## 7 FPGA s volatilní konfigurací

Technologie užitá pro uložení konfigurace hradlového pole je nejvýznamnější faktor pro výběr součástky pro finální aplikaci. Principiálně rozeznáváme dva základní typy FPGA podle uložení konfigurace: s volatilní a s nevolatilní konfigurací.

FPGA s volatilní konfigurací ukládají konfigurační informace do paměťových buněk typu SRAM. FPGA postavená na této technologii mají jednoznačnou výhodu ve snadné konfiguraci a rekonfigurovatelnosti i za běhu systému.

Další nespornou výhodou SRAM FPGA je technologický náskok. K výrobě FPGA s volatilní konfigurací je používán standardní technologický proces CMOS bez dodatečných kroků, a proto jsou obvody SRAM vždy o jednu až dvě technologické generace napřed před ostatními FPGA. Bez zajímavosti není, že SRAM FPGA se pro svou pravidelnou strukturu dnes stávají prvními návrhy vyráběnými v nových technologických procesech (například 90 nm UMC CMOS proces byl „vyládn“ na obvodech FPGA).

Na druhou stranu použití paměťových buněk SRAM přináší i nevýhody. Programovatelný obvod musí být po startu systému nakonfigurován – k tomu je obvykle třeba externí paměť a jednoduchý řadič, to znamená větší potřebný prostor na desce s tištěnými spoji a více součástek. Systém postavený kolem SRAM FPGA proto není schopen pracovat okamžitě po zapnutí napájecího napětí, nejprve je nutné FPGA nakonfigurovat. To může trvat řádově do několika set milisekund. Vlastní konfigurační proces způsobuje zvýšenou spotřebu elektrické energie po startu systému (tzv. *inrush current*). Na příslušný proudový odběr je pak nutné dimenzovat i napájení systému. Nevýhodou také představuje vyšší spotřeba energie za běhu zařízení. I když je FPGA v režimu s nízkým příkonem (*low-power mode*), odebírá stále proud ze zdroje na udržení konfigurace. Konečně, u SRAM FPGA je těžší zajistit zabezpečení intelektuálního vlastnictví, protože konfiguraci obvodu lze jednoduše vyčíst z konfigurační paměti (lze ovšem použít speciální podporu pro šifrování konfigurace, tu ale nemají všechna FPGA).

## 8 FPGA s nevolatilní konfigurací

FPGA s nevolatilní konfigurací ukládá konfigurační bity v nějakém typu nevolatilních paměťových buněk. Typicky se lze setkat s pamětí *flash*, EEPROM a s tzv. *antifuses* (antipojistkami). Jasná nevýhoda takového obvodu je obtížnější změna konfigurace – u *antifuses* je nemožná, u ostatních

je s ní třeba počítat při návrhu. Výhodou je ovšem nižší spotřeba energie výsledným zařízením (chybějící konfigurační fáze, mód *low-power* s nižší spotřebou) a lepší zabezpečení intelektuálního vlastnictví (konfigurace je uložena přímo v obvodu a obvykle lze propálením příslušné propojky znemožnit i její zpětné vyčtení). Nevolatilní obvody FPGA přinášejí proti volatilním obvodům FPGA i vyšší odolnost proti radiaci a nejvyšší úroveň ochrany intelektuálního vlastnictví – i po rozpouzdření čipu na specializovaném pracovišti je velmi obtížné zjistit skutečnou konfiguraci obvodu.

Jednotlivé technologie mají svá specifika:

- *Antifuse* je v nenaprogramovaném stavu rozpojená a programováním se propojí (opak pojistky) – vytvoří se rezistivní spojka. FPGA je ovšem konfigurovatelné pouze mimo cílovou aplikaci. V případě masové produkce je nutné zvážit dostupnost dostatečných množství obvodů (čas potřebný pro naprogramování obvodu může být kritický) a případně využít možnosti konverze FPGA do zákaznického integrovaného obvodu. FPGA konfigurované pomocí *antifuses* má výhodu ve zvýšené radiační odolnosti ve srovnání s ostatními technologiemi. Samotné *antifuses* ale pro zvýšení odolnosti nestačí. Je vhodné je dále kombinovat s trojnásobnou redundancí registrů a majoritním dekodérem (*triple-module redundancy* – TMR, např. výrobce Actel, rodina obvodů RTAX-S). Výroba obvodů FPGA s *antifuses* ovšem vyžaduje dodatečné kroky ke standardnímu procesu CMOS, a proto jsou tyto obvody obvykle o jednu až dvě generace za SRAM FPGA, což částečně odstraňuje jejich výhodu v nižší spotřebě a vyšší rychlosti kompaktnější logické matice FPGA (*antifuses* jsou menší než buňky SRAM).

- *EEPROM/flash* FPGA umožňují jak programování v aplikaci, tak před vlastním použitím. Spotřeba energie obvodem je zhruba mezi *antifuse* FPGA a obvody SRAM FPGA a stejně jako *antifuse* FPGA i obvody EEPROM/flash FPGA ke své výrobě potřebují proces CMOS s dodatečnými kroky. I tyto tedy technologicky poněkud zaostávají za SRAM FPGA.

Existují i FPGA, která jsou mezi oběma protipóly – obvody s SRAM konfigurací a *flash* pamětí integrovanou přímo v pouzdře FPGA obvodu (viz např. Xilinx Spartan 3AN). Výhoda kombinovaného přístupu je ve zmenšení plochy a složitosti desky plošných spojů zařízení.

## 9 Jak vybírat FPGA

### 9.1 Charakter operací prováděných obvodem FPGA

Výběr FPGA obvodu pro konkrétní aplikaci není vůbec jednoduchým úkolem.

Pokusíme se zde proto shrnout některé základní parametry budoucího systému, které by měly být při výběru obvodu brány v úvahu.

Je FPGA zamýšleno pro řídicí funkce – například pro řízení komunikace? Nebo bude provádět velké množství výpočtů (aplikace pro číslicové zpracování signálů)? Řada výrobců vyrábí svoje obvody v různých řadách podle charakteru použití – i stejné velká FPGA se pak liší v počtech integrovaných násobiček, pamětí atd.

### 9.2 Velikost a rychlost budoucího obvodu

Je třeba mít alespoň hrubý odhad velikosti finálního návrhu a požadované rychlosti logiky (potřebná hodinová frekvence). Výrobci obvodů FPGA obvykle uvádí u produktů velikost hradlového pole v tzv. ekvivalentních hradlech. Jedná se o termín vypůjčený z oblasti zákaznických integrovaných obvodů udávající, jak přibližně velký návrh na zákaznickém obvodu se vejde do příslušného FPGA. Toto číslo je nicméně nutné brát s rezervou. Udávané velikosti mezi jednotlivými výrobci téměř nelze srovnávat a i u produktů od jednoho výrobce je třeba znát skutečnou konfiguraci matice FPGA (metodiky pro převod velikosti FPGA matice na ekvivalentní hradla bývají poněkud obskurní). Podobná je situace co se rychlosti obvodu týče – FPGA obvody jedné typové řady se typicky vyrábí v několika provedeních podle rychlosti (tzv. *speed grade*).

### 9.3 Počet vstupních a výstupních signálů

Počet potřebných v/v linek obvodu implementovaného do FPGA je třeba zohlednit při volbě vhodného pouzdra obvodu.

### 9.4 Spolehlivost finální aplikace

Bude FPGA pracovat v náročnějších podmínkách, nebo předpokládáme běžnou spotřební elektroniku? Požadujeme speciální teplotní rozsah? Případně zvýšenou radiační odolnost? Například pro automobilovou elektroniku existují speciální produktové řady obvodů FPGA. Obecně je v případě požadavku na vyšší spolehlivost vhodné zvážit použití nevolatilních obvodů.

### 9.5 Definice požadavků na finální aplikaci

Jestliže nejsou jisté implementační detaily finálního systému (například síťový přepínač má podporovat protokol, který stále ještě není plně specifikován standardizačním orgánem), může být výhodné užít SRAM FPGA s možností rekonfigurace finálního zařízení.

### 9.6 Aplikace typu startup time

Chceme-li aby systém byl plně funkční hned po zapnutí napájecího napětí, nemůžeme použít SRAM FPGA.

### 9.7 Spotřeba energie

Systémy s nevolatilní konfigurací mají obvykle nižší spotřebu než v případě FPGA s volatilní konfigurací. Dříve bylo nemyslitelné použít obvod FPGA v systému napájeném z baterie. Dnes už ale existují speciální řady hradlových polí pro nízkopříkonové aplikace (např. Actel Iglou). Se spotřebou energie ovšem souvisí i teplo disipované obvodem během normální práce systému (dynamický příkon související se spínáním hradel a statický příkon způsobeným svodovým proudem). Podle množství uvolňovaného tepla, prostorových nároků aplikace a potřebného množství vstupních a výstupních signálů obvodu je třeba zvolit vhodné pouzdro obvodu FPGA a případně navrhnout i dodatečné chlazení.

### 9.8 Zabezpečení intelektuálního vlastnictví

Nevolatilní FPGA poskytují vyšší úroveň zabezpečení intelektuálního vlastnictví, obvykle ale za cenu ztráty flexibility ve finálních aplikacích.

### 10 Krátce o návrhu obvodů na FPGA

V počátcích programovatelných obvodů se pro návrh používaly proprietární jazyky (např. ABEL) či kreslení schémat. Dnes je standardem pro návrh obvodů FPGA použití příslušného jazyka pro popis hardwaru (HDL) na úrovni RTL. Pro popis se používají nejčastěji dva jazyky – buď VHDL, nebo Verilog. Popisujeme-li obvod na úrovni RTL, znamená to, že ho popisujeme jako sadu registrů propojených kombinační logikou realizující logické operace. Výhodou použití úrovně RTL je práce na poměrně vysokém stupni abstrakce. Jeden řádek zdrojového kódu je v hardwaru reprezentován typicky desítkami/stovkami hradel, což zvyšuje produktivitu práce a zpřehledňuje vlastní návrh. Popis RTL systému je posléze konvertován do konfiguračního souboru obvodu FPGA v několika krocích (syntéza, mapování, rozmístění a propojení, generování konfiguračního souboru). Návrh zjednodušuje a zrychluje i široká dostupnost tzv. jader (IP

cores) – už navržených a zverifikovaných bloků realizujících specifické funkce, např. číslicové filtry, komunikační rozhraní, ale i celé procesory [5 a 6].

Nástroje pro návrh obvodů FPGA se nicméně stále vyvíjejí. Dnes pomalu získává popularitu tzv. syntéza na vyšší úrovni – HLS [7]. Zjednodušeně ji můžeme popsat jako přímou konverzi algoritmu napsaného v nějakém vyšším programovacím jazyce (např. C) do hradlového pole.

### 11 Závěr

Příspěvek krátce shrnuje historii, současnost i budoucnost programovatelných hradlových polí spolu s možnými aplikacemi, jejich technologií i pravidly pro výběr vhodného obvodu FPGA pro příslušnou cílovou aplikaci. Z uvedeného přehledu je zřejmé, že

trh s obvody FPGA již vyzpěl. Je k dispozici velké množství produktových řad pro nejrůznějších aplikací od mnoha výrobců a je jen na návrháři příslušné aplikace, aby zvolil architekturu, technologii a nakonec obvod vhodný pro jeho konkrétní aplikaci. Na článek bude volně navazovat série dalších příspěvků pokrývajících detailněji jednotlivé aspekty práce s programovatelnými hradlovými poli.

Ing. Jakub Štastný, Ph.D.  
katedra teorie obvodů  
FEL ČVUT v Praze,  
ASICentrum, s. r. o.

### LITERATURA

- [1] Internet: [www.mathstart.com/Products.php](http://www.mathstart.com/Products.php)
- [2] ŠTASTNÝ, J., BÍLÝ, P., An FPGA Microcontroller Design. *Elektrorevue*, no. 30, 2006. Dostupné na <http://www.elektrorevue.cz/clanky/06030/english.htm>
- [3] WIANGTONG, T., CHEUNG, P. Y. K., LUK, W., Hardware - software codesign. *IEEE signal processing magazine*, 20 (2005), no. 5, pp.14–22.
- [4] Xilinx, Virtex-4 user guide, v2.3, August 2007.
- [5] Internet: <http://amber.feld.cvut.cz/fpga>
- [6] Internet: [www.opencores.org](http://www.opencores.org)
- [7] ŠTASTNÝ, J., Syntéza na vyšší úrovni. [http://amber.feld.cvut.cz/fpga/prednasky/hll\\_synteza/hlls.html](http://amber.feld.cvut.cz/fpga/prednasky/hll_synteza/hlls.html)
- [8] Clive Maxfield, Design Warrior's guide to FPGA, Elsevier, 2004, ISBN-13: 9780750676045
- [9] JOHN, M., SMITH, S., *Application Specific Integrated Circuits*. Addison Wesley Publishing Company VLSI Design Series 1997, ISBN 0-201-50022-1
- [10] BROWN, S., ROSE, J., Architecture of FPGAs and CPLDs: A Tutorial. *IEEE Design and Test of Computers*, 13 (1996), no. 2, pp. 42–57. <http://www.eecg.toronto.edu/~jayar/pubs/brown/survey.html>
- [11] *FPGA Based prototyping: Why all ASICs should be prototyped using FPGAs and CPLDs Technical Report*, Synplicity, April 2005.
- [12] Computer History Museum, <http://www.computerhistory.org/semiconductor/timeline>
- [13] Xilinx, Programmable Logic Design QuickStart Handbook, [www.xilinx.com/publications/products/cpld/logic\\_handbook.pdf](http://www.xilinx.com/publications/products/cpld/logic_handbook.pdf)

### ■ Trend Micro ohlásil nové řešení

Společnost Trend Micro uvedla na trh řešení, které chrání podnikové uživatele před webovými hrozbami. Trend Micro InterScan Web Security Appliance nabízí ochranu sítě typu *all-in-one* před přichozími útoky, jako jsou viry, spyware, nebezpečné webové stránky a nevhodný webový obsah. Nástroj je nyní dostupný ve dvou verzích: InterScan Web Security Appliance Standard a InterScan Web Security Appliance Advanced, který poskytuje kromě funkcí standardní verze

plně integrované filtrování URL a zabezpečení appletů a prvků ActiveX založené na politikách. InterScan Web Security Appliance nyní obsahuje zjišťování důvěryhodnosti webových stránek v reálném čase od firmy Trend Micro. Funkce hodnotí bezpečnost dané webové stránky podle jejího chování i obsahu. Tato technologie pro zjišťování reputace stránek doplňuje výkonné skenování obsahu a filtrování URL, kterými produkt disponuje již několik let. InterScan Web Security Appliance nabízí možnost jednoduché ochrany webové brány typu *plug-and-protect*. Přichází a odchází provoz je

analyzován, a nástroj tak funguje jako chybějící článek bezpečnostního řetězce. Webová konzola umožňuje administrátorům centrální management a maximální flexibilitu při definování přístupových práv pro delegovanou správu. Stejně jako u všech ostatních podnikových řešení Trend Micro může být více nástrojů InterScan Web Security Appliance centrálně řízeno pomocí softwaru Trend Micro Control Manager. Tato centrální správa umožňuje administrátorům efektivně monitorovat a spravovat celé vícevrstvé bezpečnostní portfolio od Trend Micro.

podle tiskové zprávy mat