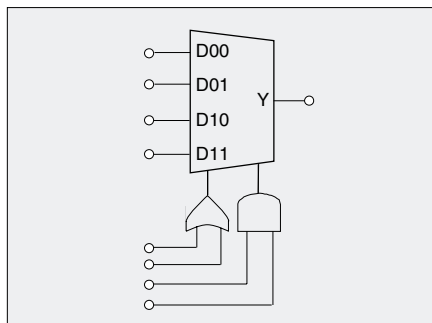


# Architektura programovatelných logických obvodů

## Úvod

V minulém čísle *ST* jsme představili programovatelné obvody, jejich vývoj a nasazení. Naznačili jsme, že se skládají z programovatelných bloků, které realizují naprogramova-



Obr. 1 Jednoduchý multiplexorový programovatelný blok (Actel Integrator)

né logické funkce a jsou propojeny programovatelným propojením. To jsou dva základní stavební kameny, z nichž jsou vystavěny všechny druhy programovatelných logických obvodů. Z nich lze vystavět obvody značně odlišné architektury.

## Realizace programovatelných bloků

Prakticky ve všech programovatelných blocích nalezneme kombinační obvody a klopné obvody typu D. Bylo by jistě možné klopné obvody vytvářet z obvodů kombinačních. Synchronní systémy, které většinou programovatelnými obvody realizujeme, potřebují hranové klopné obvody. To je struktura natolik složitá a přitom natolik opakovaná, že se vyplácí jí mít v obvodu hotovou. Navíc používané metody syntézy zřídka používají jiné klopné obvody než prosté registry.

Problémem tedy zůstává, jak realizovat programovatelný kombinační obvod. Možných metod je několik.

## Tabulky

Každou kombinační funkci můžeme zadat pravdivostní tabulkou, která udává hodnotu výstupu pro každou kombinaci hodnot vstupů. Tuto tabulku můžeme jednoduše nahrát do paměti, ať již ROM či RAM. Paměť bude mít tolik adresních vodičů, kolik má funkce vstupů a jeden výstup. Nevýhodou je, že pro  $n$  vstupů potřebujeme  $2^n$  buněk paměti a tabulková realizace je omezena na funkce o nevelkém počtu vstupů, v praxi ne více než pět. Jistou výhodou tabulek je, že se nevelkým doplněním obvodu dají použít jako malá paměť RAM nebo paměť ROM s obsahem daným konfigurací obvodu.

## Multiplexory

Multiplexor je velice univerzálním kombinačním prvkem, jak je známo již od dob multi-

plexorů MSI. Pokud na datové vstupy multiplexoru přivedeme konstanty a na výběrové vstupy vstupní signály realizované funkce, jsme opět schopni realizovat libovolnou funkci. Ve skutečnosti je to jenom jinak uspořádaná pevná paměť  $n \times 1$  se všemi nevýhodami. Mnoho funkcí se však dá realizovat i menšími multiplexory, přivedeme-li některé vstupní signály na datové vstupy multiplexoru.

Multiplexor je tedy pevný a programování vlastně obstarává programovatelné propojení. Takto lze pracovat s libovolnými jinými pevnými kombinačními obvody, případně s jejich kombinací. Typický multiplexorový programovatelný blok obvodů Actel je na obr. 1.

## Programovatelný montážní součin

Hlavní myšlenka obvodů PAL (Programmable Array Logic) vychází rovněž z představy paměti s kapacitou  $2^n$  buněk. Protože je v takovém případě výstupů adresního dekodéru příliš mnoho, udělejme adresní dekodér menší, ale programovatelný.

Potom budou jednotlivé výstupy dekodéru realizovat součiny proměnných a vlastní matice paměti součty těchto součinů. Zkušenost ukázala, že je výhodné součtovou matici zafixovat. Každý výstupní signál je pak součtem pevného počtu volitelných součinů.

Jádrum obvodu je tedy součinnové hradlo s velkým počtem vstupů. Je realizováno příslušným počtem tranzistorů, pracujících do společné zátěže. Programování se děje odpojováním tranzistorů příslušejících jednotlivým vstupům jiným tranzistorem v sérii. Ten je řízen nejčastěji nábojem na svém plovoucím hradle (elektricky mazatelné programování). Jako obvykle, nesáz je v detailu. Zde je to společná zátěž, která při výstupu v nule má nezanedbatelnou spotřebu a není slučitelná s technologií CMOS. Je to také důvod, proč i dnes obvody GAL spotřebovávají desítky mA napájení.

## Realizace programovatelného propojení

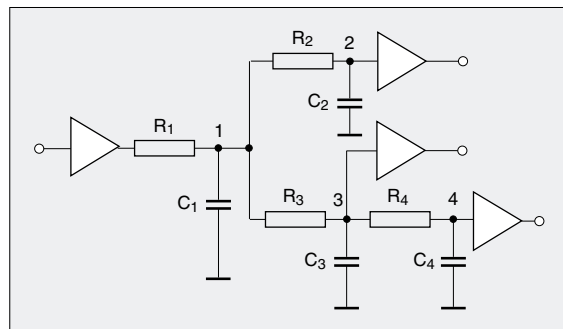
Propojení se vždy skládá z propojovacích vodičů různé délky a uspořádání, spojovaných programovatelnými spínači.

Technologie spínání se může lišit, pro použití je důležitější, zda spínací prvek zesílí (aktivní spínače) či nikoliv (pasivní spínače). Zpoždění na propojovacích cestách je v programovatelných obvodech mnohem významnější složkou než v jiných technologiích.

## Pasivní spínače

V reprogramovatelných obvodech jsou pasivní spínače odvozeny od přenosových hradel, známých z obvodů CMOS. Na rozdíl od nich je tvoří jen jeden tranzistor, nejsou tedy symetrické vzhledem k logickým úrovním. Každý takový spínač řídí jedna buňka statické konfigurační paměti, buď RAM nebo elektricky reprogramovatelné.

V obvodech jednorázově programovaných se užívají destruktivní prvky. Kdysi u malých obvodů PAL to byly zúžené části vodiče, v podstatě pojistky, které se přerušily programovacím proudem. Nynější prvky při programování naopak přecházejí do vodičového stavu, proto se jim říká antipojistky. Jsou založeny buď na protavení dielektrika ohřevem vodiče nad ním (Actel), nebo na průrazu vrstvy amorfního křemíku napětím (QuickLogic). Plocha, zabraná antipojistkou,



Obr. 2 Příklad přibližného výpočtu zpoždění v RC-modelu spoje:  
 $t_{d1} = R_1 C_1 + R_1 C_2 + R_1 C_3 + R_1 C_4 + R_2 C_2$ ;  
 $t_{d3} = R_1 C_1 + R_1 C_2 + R_1 C_3 + R_1 C_4 + R_3 C_3 + R_3 C_4$

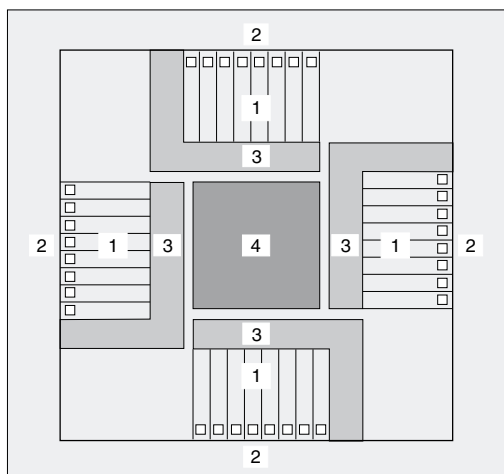
je asi desetinou plochy potřebné pro spínač a buňku pro jeho ovládání. Rovněž odpor v sepnutém stavu je o řád menší. Pro antipojistky je však třeba robustních tranzistorů k programování a přidavných kroků ke standardnímu výrobnímu procesu CMOS.

Pasivní spínače jsou významným zdrojem zpoždění. Vnější do cesty signálu podélný odpor o několik řádů větší než odpor přílehlého úseku vodiče a příčnou kapacitu. K nim je třeba připočítat výstupní odpor logických členů, kapacitu vodičů a vstupní kapacitu hradel. Dostáváme tak rozvětvené síť RC (obr. 2). Existuje mnoho způsobů jejich analýzy, pro výklad vystačíme s poměrně jednoduchou a přesnou Elmorovou metodou. Ta počítá příspěvek každého podélného odporu jako časovou konstantu toho odporu a všech kapacit, které leží za ním. Zpoždění signálu tedy závisí nejen na počtu spínačů v cestě signálu, ale i na větvení signálu a na vlastnostech ostatních větví.

Zpoždění je značně velké, může dosáhnout i několikanásobku zpoždění kombinačního obvodu v programovatelném bloku.

Ještě větší problém nastane, pokud nevíme dopředu, přes kolik spínačů signál povede po rozmístění bloků a návrhu propojení.

Pak je zpoždění signálu nejen velké, ale také nepředvídatelné. Obvody zaměřené na rychlost se proto snaží nejen o spínače s malým odporem, ale i o takovou topologii propojení, kde by počet spínačů byl (alespoň přibližně) stálý. V praxi je někdy taková podmínka obtížně splnitelná.



Obr. 3 Uspořádání obvodů CPLD; 1 – jednotlivé bity programovatelných bloků; 2 – vývody čipu; 3 – lokální propojení; 4 – globální propojení

### Aktivní spínače

V moderních rychlých a velkých obvodech (např. procesorech) hraje zpoždění vodičů značnou roli i bez parazitních veličin spínačů. Jejich konstruktéři se dopracovali k empirickému poznatku, že signál na delší vzdálenost dopraví nejrychleji tehdy, bude-li se součet zpoždění na vodičích přibližně rovnat součtu zpoždění na mezilehlých zesilovačích členech.

Tento princip se dá použít i v programovatelných obvodech. Jistě bychom se zbavili vzájemné závislosti větví obvodu na obr. 2, kdyby tyto větve byly odděleny zesilovači. Nejlépe se to provádí v takovém místě, kde směr šíření signálu je dán konstrukcí programovatelného obvodu (např. na výstupu programovatelného bloku). Jinak musí návrhový systém znát směr šíření signálu za provozy naprogramovaného obvodu a podle toho aktivovat zesilovače příslušného směru. Signály, které se šíří obousměrně, se takto nedají ošetřit vůbec. Spojí na krátké vzdálenosti vycházejí lépe v pasivním provedení. Proto zpoždění i v obvodech s aktivními spínači zůstává velké a špatně předvídatelné.

### Pevné propojení (hodiny a jiné globální signály)

V rychlých obvodech je rozvod hodinového signálu časově velmi kritický. S rostoucí rychlostí a zvětšujícími se rozměry čipu roste nebezpečí fázových posuvů. Proto jsou hodinám zpravidla vyhrazeny zvláštní rozvody s programovatelností velmi omezenou nebo zcela vynechanou a s pečlivě navrženou topologií.

### Vybavení pro synchronní návrh

Popsané dynamické vlastnosti propojení prakticky vylučují návrh asynchronních obvodů. Pokročilejší návrhové prostředky sice mohou

zaručit, že zpoždění na daném spoji nepřesáhne zadanou mez (nebo alespoň ohlásit selhání), ale nedokážou implementovat složitě časově závislosti, které jsou výsledkem asynchronního návrhu. Ostatně asynchronní obvody velikosti desítek tisíc ekvivalentních hradel nejsou, a dlouhou dobu ještě nebudou, průmyslovou praxí.

Běžné programovatelné obvody dovolují rozdělení obvodu do několika domén s různými hodinovými signály. Návrhové prostředky pak umějí zkontrolovat časování uvnitř domény a není-li jich mnoho, vztahy mezi doménami lze ohlídat ručně.

K usnadnění synchronního návrhu je většina klopných obvodů vybavena signálem povolení hodin. Nulování a nastavení jsou často synchronní, asynchronní nulování se používá pouze globálně.

Hodinový rozvod programovatelného obvodu má sice mnohem menší zpoždění než ostatní propojení, ale i tak vznikají na vývodech obvodu problémy. Vlivem zpoždění hodin uvnitř obvodu jsou vnější signály vzorkovány se zpožděním proti aktivním hraně hodin na vývodu obvodu.

Časové požadavky na vnější signály vycházejí nesymetricky – záporný předstih a velký přesah, což je nežádoucí. Jednodušší obvody používají volitelné zpoždění ve vstupních obvodech, náročnější obvody kompenzují zpoždění fázovou smyčkou.

### Problém granularity

Známe způsoby, jakým lze vytvořit programovatelné prvky, ale dosud jsme nezmínili strukturu a velikost programovatelného bloku samotného. Má to být kombinační logika o několika málo vstupech doplněná jedním klopným obvodem, nebo blok velikosti několikabitové operační jednotky s vnitřními registry? Oba tyto krajní případy se vyrábějí a používají, což naznačuje, že jednoznačná odpověď neexistuje. Granularita je nicméně dobrým vodítkem k přehledu konstrukčních řešení a k objasnění současných trendů v konstrukci programovatelných obvodů.

### Hrubozrnné architektury

Na obr. 3 je naznačena struktura typického obvodu CPLD (Complex Programmable Logic Device). Obvod je jakoby vystavěn z několika menších obvodů PAL, tedy z bloků, které mají široká hradla realizovaná technikou programovatelných montážních součinů a jejichž lokální zpětnovazební vodiče dovolují spojit každý bit s každým. Každý takový programovatelný blok má šířku 10–18 bitů a velké obvody CPLD jich obsahují až 48.

### Globální propojení

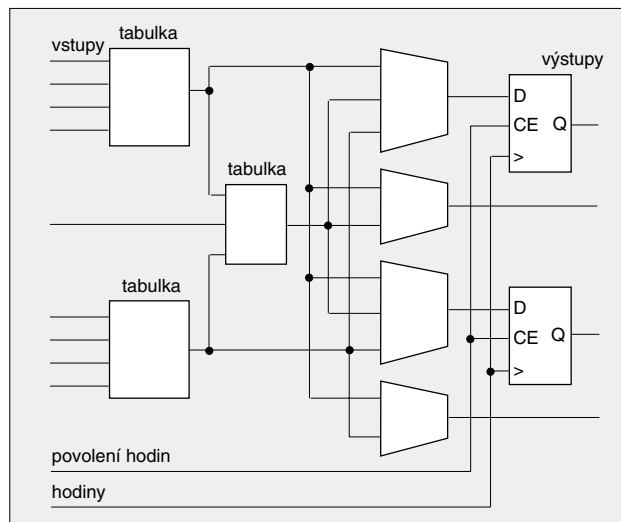
Druhý (globální) stupeň propojení spojuje všechny podobvody způsobem „každý s každým“. Zpoždění obou úrovní propojení je stále a vzhledem k malému počtu spínačů v cestě nízké. Plocha křemíku pro globální propojení ovšem roste s druhou mocninou počtu bitů všech bloků, což je hlavním omezením růstu velikosti těchto obvodů. Tyto vlastnosti vedou k relativně menším, ale rychlým obvodům.

### Vstupně-výstupní a vnitřní bloky

Střední a větší obvody CPLD mají strukturu bloku ještě o něco složitější. Programovatelné bloky nejsou spojeny přímo s vývody obvodu. Signály zpracovávají vstupně-výstupní bloky a teprve přes lokální propojení jsou přivedeny na vnitřní bloky a do globálního propojení.

### Paměti a čítače

Přes zmíněné překážky v konstrukci velkých obvodů jejich velikost stále roste. Při dnešních 50 000 ekvivalentních hradel se dostáváme k podsystemu, který je dost velký na to, aby byl relativně autonomní a s okolím



Obr. 4 Značně zjednodušený tabulkový programový blok (Xilinx Spartan); multiplexory jsou ovládané konfiguračním programem

spolupracoval přes paměť, frontu nebo registrové pole. Novější obvody jsou vybaveny jednobránovými nebo dvoubránovými paměťmi či frontami velikosti asi 4 kb i registrovými poli, které se dají konfigurovat jako čítače a časovače.

### Technologie

Typickou technologií obvodů je EECMOS. Vyrábějí je všichni významní výrobci (Xilinx, Altera, Atmel, atd.), ale za průkopníka na poli technologie i architektury je považována firma Lattice. Nejnověji se začínají objevovat obvody programované SRAM (Philips, Cypress).

### Jemnozrnné architektury

Multiplexorový blok na obr. 1 je příkladem programovatelného bloku s nejmenší granularitou. V dalších verzích byl doplněn klopným obvodem. Nicméně daleko typičtějším případem je zjednodušená struktura progra-

movatelného (vnitřního) bloku obvodu řady Spartan firmy Xilinx (obr. 4). Takové obvody jsou zpravidla nazývány FPGA (Field Programmable Gate Array).

Hlavními částmi programovatelného bloku jsou tři tabulky, realizující kombinační funkce, a dva klopné obvody. Tato struktura implementuje dvě libovolné kombinační funkce čtyř proměnných nebo jednu libovolnou funkci pěti proměnných nebo některou funkci až devíti proměnných. Výstupy obvodu jsou přímé nebo přes klopné obvody, a to i současně.

**Optimální velikost programovatelného bloku**

Je tato velikost a počet vstupů optimální? Na torontské univerzitě běží již dlouhou dobu výzkumný program, který se na podobné otázky snaží odpovědět experimentálně. Návrhové nástroje, vytvořené pro projekt, umí pracovat s velmi obecným popisem programovatelného obvodu. Díky tomu bylo možné navrhnout sérii obvodů, vzatých z praxe, čímž vznikly obvodové varianty s postupně měněnými parametry nebo strukturou. Za povšimnutí stojí některé zajímavé výsledky z hlediska spotřeby plochy křemíku:

- pro tabulky s jedním výstupem je optimální velikost 4 vstupy,
- tabulky s více výstupy jsou vždy horší než s jedním výstupem,
- máme-li k dispozici 32 buněk paměti, je nejlépe je uspořádat do čtyř tabulek po třech vstupech,
- přechod od tabulek k strukturám PLA ušetří jen nepatrně.

Závěry výzkumu z hlediska rychlosti obvodu:

- při průměrně rychlém propojení jsou nejlepší tabulky o pěti a šesti vstupech,
- pro rychlé propojení je lepší struktura z obr. 1,
- tato struktura je vždy lepší než čisté multiplexory a než pevná hradla AND-OR.

Výsledky experimentů jsou jistě závislé na způsobu práce návrhového systému a na dalších faktorech, přece jen však naznačují, že struktura z obr. 4 je vyhovující z mnoha hledisek.

**Vstupně-výstupní bloky**

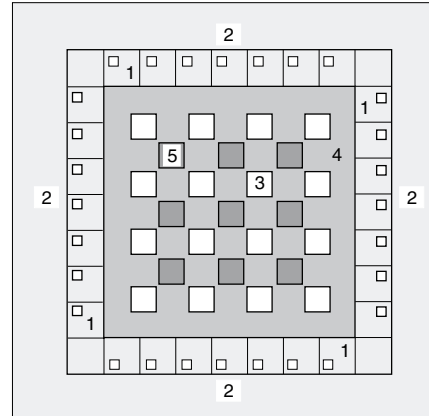
Dělení na vnitřní a vstupně-výstupní bloky je u obvodů jemnozrné architektury nutností. Programovatelné bloky již nejsou sdruženy s vývodem čipu a propojení nemůže přenášet proudy potřebné na vývodech. Vstupně-výstupní blok je možno programovat jako kombinační nebo s registrem, ve funkci výstupu je zpravidla možný třetí stav.

**Mřížkové a řádkové propojení**

Propojení má většinou podobu mřížky (obr. 5), někdy je uspořádáno do řádek. Hrany mřížky jsou tvořeny propojovacími vodiči. V místech vývodů po obvodu programovatelných bloků jsou rozmístěny programované spínače, které dovolují spojit každý vývod s více (ne nutně všemi) vodiči. V uzlech mřížky pak jsou spojovací uzly podobné křížovému přepínači. Od křížového přepínače se však v některých bodech liší:

- ve stejném směru vodič může pokračovat nebo může být přerušen,
- spojení není „každý s každým“ a ukázalo se, že počet spínačů lze podstatně omezit bez velkých následků v propojitelnosti.

Pokud by každý spojovací vodič procházel spínačem v každém uzlu mřížky, delší spoje by měly neúnosné zpoždění. Proto



**Obr. 5** Uspořádání obvodu FPGA jemnozrné architektury s mřížkovým propojením; 1 – vstupně-výstupní bloky; 2 – vývody čipu; 3 – vnitřní bloky; 4 – propojení; 5 – redukované křížové spínače; (propojení zabírá i přes 70 % plochy čipu)

mřížkové obvody vždy nabízejí vodiče různé délky, např. dvěma hranami mřížky počínaje a celou šířkou či délkou čipu konče.

**Dlaždicové obvody**

Jestliže v multiplexorových blocích programovatelné propojení vytváří programovatelnost bloku, u dlaždicových obvodů je tomu přesně naopak. Jednotlivé programovatelné bloky na sebe nasedají přímo, bez odděleně programovatelných spojů (v praxi se zcela bez spojů neobejdeme). Spínače uvnitř buněk pak předávají signály mezi (typicky velmi jednoduchými) pevnými logickými členy a horním, dolním, levým a pravým okrajem, případně mezi vývody navzájem. Díky tomu se obvody snadno po částech rekonfigurují, neposkytují však příliš efektivní prostředky pro implementaci výpočetních funkcí.

**Prostředky pro iterativní obvody**

Iterativní obvody, například některé sčítačky nebo čítače, se vyznačují pravidelnou strukturou, která však obsahuje velmi dlouhou časově kritickou cestu signálu kombinačním obvodem. To je pro realizaci na FPGA velmi nepříjemné. Proto se v programovatelných blocích záhy objevily specializované obvody pro přenosy, které jsou navíc mezi jednotlivými bloky propojeny neměnným, a proto rychlým spojením. Rozložení jednotlivých bitů iterativního obvodu pak nemůže být libovolné, nýbrž musí sledovat připravené cesty pro přenos.

**Paměti a speciální obvody**

Uvedli jsme, že tabulky implementující kombinační funkce se dají používat jako paměti. To je u jemnozrných obvodů, zejména firmy Xilinx, časté. Například obě větší tabulky na obr. 4 jsou konfigurovatelné jako dvě jednobránové paměti 16x1, jedna jednobránová 32x1 nebo jedna dvoubránová 16x1.

Je mnoho případů, kdy obvod realizovaný FPGA je připojen přímo k adresované sběrnici. Rychlý dekodér adresy je další pro FPGA nepříjemný obvod. Proto se vstupně-výstupní bloky některých obvodů dají zapojit jako hranové dekodéry, tedy široká hradla AND.

**Technologie**

Obvody s jemnozrnou architekturou jsou nejčastěji programovány obsahem konfigurační SRAM. Jsou samozřejmě výjimky, např. použití technologie EEPROM u obvodů Gatefield. Nejznámější jsou řady XC4000 a Spartan firmy Xilinx.

**Hierarchické obvody (proměnná zrnitost)**

V obou diskutovaných architekturách – hrubozrné i jemnozrné – sledujeme jistou hierarchičnost. U obvodů CPLD je to především hierarchie propojení, u jemnozrných obvodů FPGA pak některé skupiny programovatelných bloků jsou úzeji svázány. V posledních letech přicházejí obvody někde uprostřed mezi těmito krajními body. Hierarchické uspořádání bloků a propojení je jejich typickým rysem, přitom celek zůstává natolik pružný, že se někdy mluví o variabilní granularitě.

**Příklady**

Strukturu CPLD silně připomínají obvody Altera řady FLEX 8K, 10K a APEX 20K. Vždy osm jednoduchých programovatelných bloků (jedna tabulka, obvody přenosu a nastavení, klopný obvod) je vybaveno lokálním propojením, které je pak teprve připojeno na globální propojení mřížkové topologie. Všechny osm bloků je propojeno cestami pro přenos a sdílí signály pro nulování, nastavení a povolení hodin. U obvodů APEX tvoří 16 těchto shluků modul vyššího řádu.

Z opačného směru dospěly k hierarchii obvody Xilinx. Velikost programovatelného bloku u řady XC5200 byla zmenšena na jednu tabulku, přenosový obvod a klopný obvod. Je to vlastně polovina programovatelného bloku dřívějších řad. Čtyři takové obvody tvoří shluk, vybavený lokálním propojením. Teprve přes lokální propojení a speciální propojovací uzly se signál může dostat do globálního propojení opět mřížkové architektury. Novější obvody řady Virtex vypadají podobně. Dvě tabulky a dva klopné obvody jsou propojeny pomocnými signály a tvoří řez. Dva řezy, tj. 4 tabulky a 4 klopné obvody, tvoří programovatelný blok. Ten má místní propojení a je připojen do propojovacího uzlu globálního propojení.

**Optimální velikost shluku**

Výše zmíněný torontský projekt zahrnul i hierarchické struktury. Proměnný počet  $N$  buněk, každá s tabulkou o čtyřech vstupech a klopném obvodu, bylo sdruženo do shluku, který měl úplně lokální propojení. Zatímco minima zabrané plochy bylo dosaženo při  $N=3$ , optimum součinu plochy a zpoždění, což je realističtější a komplexnější kritérium, bylo dosaženo při  $N=7$ . Při větším

shluku se totiž více časově kritických spojů „schová“ do lokálního propojení shluku a obvod je rychlejší.

Dále byl zkoumán vliv omezeného počtu vodičů  $I$  které spojují shluk s globálním propojením. Ještě při  $I=2N+2$  bylo využití logiky 98 %, téměř úplné. Poznamenejme, že  $N=4$  a  $I=10$  přesně odpovídá obvodům ATT ORCA.

#### *Paměti střední velikosti a široká hradla*

Při diskuzi obvodů CPLD jsme konstatovali, že systémy jisté velikosti již vyžadují paměti řádově jednotek až desítek kb pro vytváření front a pracovních pamětí. Hierarchická struktura obvodu (podobně jako struktura CPLD) nabízí vhodnou úroveň velikosti bloku pro začlenění takových pamětí. Typicky bývají jednobránové i dvoubránové, synchronní i asynchronní.

Jestliže je tabulku možno použít jako RAM, je to jistě možné i obráceně. Některé obvody takto využívají i střední paměti RAM jako široká hradla. Jiné přidávají vždy několik širokých hradel ke shluku programovatelných bloků.

## Trendy vývoje

Hnacím motorem je vývoj polovodičové techniky. Programovatelné obvody jsou zhusta vyráběny nejmodernějšími polovodičovými procesy. Někteří výrobci polovodičů tyto obvody, díky jejich regulární struktuře používají k ladění výrobních procesů. Hustota a rychlost obvodů rostou, i když odstup od hustoty a rychlosti plně zakázkových obvodů se příliš nemění.

Tyto kvantitativní změny však znamenají kvalitativně jiné požadavky na strukturu obvodu. Podsystem o 100 000 ekvivalentních hradel jinak komunikuje s okolím, používá jinou strukturu paměti než menší obvody. Nutnost efektivního propojení velkých obvodů pak vede k integraci jemnozrnných a hrubozrnných architektur a ke vzniku hierarchií. Integrace se v současnosti neprojevuje unifikací, nýbrž tím, že se jednotlivé větve vývoje prolínají.

Dnešní programovatelný obvod je tedy nejen rozsáhlá, ale také složitá struktura s mno-

ha druhy logických i propojovacích prostředků. Jejich některé vlastnosti (např. charakteristiky zpoždění signálů) nejsou obvyklé v jiných technologiích. Návrh a ověření takového obvodu tedy vyžaduje specifické pracovní postupy a nástroje.

Ing. Jan Schmidt, CSc.  
katedra počítačů FEL ČVUT

#### LITERATURA

- [1] Trimberger, S.M., ed.: *Field-Programmable Gate Array Technology*, Kluwer Academic Publishers, Boston, Dordrecht, London, 1994
- [2] Brown, S.D. et al.: *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1992
- [3] Betz, V., Rose, J.: *Cluster-Based Logic Blocks for FPGAs: Area-Efficiency vs. Input Sharing and Size, Custom Integrated Circuits Conference 97*, Santa Clara 1997
- [4] <http://www.xilinx.com/>
- [5] <http://www.altera.com/>
- [6] <http://www.gatefield.com/>