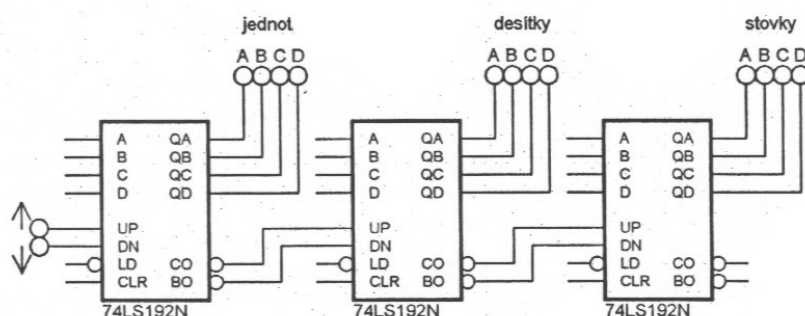


Následující obrázek ukazuje, jak jednoduše lze čítače této řady použít.



Jedná se o obousměrný čítač v rozsahu 0 až 999, tedy tří dekád. Pokud by vám to bylo málo, lze jednoduše přidat další obvody. Snad není třeba připomínat, že je třeba ošetřit zde nevyužité vstupy nulování a přednastavení.

6.9 Hradlová pole druhé a třetí generace

Hradlová pole, tak jak byla představena ve své první generaci, se samozřejmě stala základem pro vývoj druhé generace. Pro ni je typické použití tzv. makrobuněk. Do nich bylo převzato všechno co nabízela výchozí základna, dokonce zde byla snaha o určitou kompatibilitu. Jednalo se o obvody typu GAL 16V8 a GAL 20V8, které ve své době nahradily všechny obvody PAL z první generace. Protože však právě přílišná vazba na první generaci výrazně omezovala aplikovatelnost těchto obvodů, nutně musel následovat vývojový skok. Proto přišly obvody GAL 22V10, které přinesly architekturu pro univerzální použití. Obvod má 22 vstupů AND, 10 výstupů a jeho typickým prvkem je tzv. makrobuňka (OLMC – Output Logic Macro Cell). Makrobuňka umožňuje vytvořit kombinační i registrové funkce. V každé makrobuňce může být vytvořena logická funkce s až 16 mintermy. Uvědomte si, že každému výstupu odpovídá jedna makrobuňka. Makrobuňky však lze slučovat, takže pokud nevádí omezení počtu funkcí, může mít jedna funkce i více mintermů. Registrové funkce jsou zapojeny až za kombinační částí a umožňují vytvářet posuvné registry, čítače atd. Přesto, že jsou tyto obvody dnes velmi používány, budoucnost patří třetí generaci hradlových poli.

Ve třetí generaci jsou to obvody CPLD a FPGA.

Obvody CPLD – Complex Programmable Logic Devices jsou přímým pokračovatelem vývoje hradlových poli druhé generace. I zde je hlavním stavebním kamenem makrobuňka typu PAL nebo PLA. Tyto makrobuňky nejsou zcela nezávislé, ale sdružují se do bloků. Těch bývá obvykle dva až šestnáct. Tyto bloky jsou relativně nezávislé a jsou propojeny programovatelnou strukturou. Makrobuňky mohou fungovat jako skryté, tedy jejich výstupy nejsou vyvedeny na výstupy, ale tvoří vnitřní signál. Kromě toho tyto obvody obsahují další prvky rozšiřující možnosti použití. Jsou to obvody typu D, T, EX-OR, klopné obvody typu latch, případně obvody pro aritmetické operace. Jednotlivé

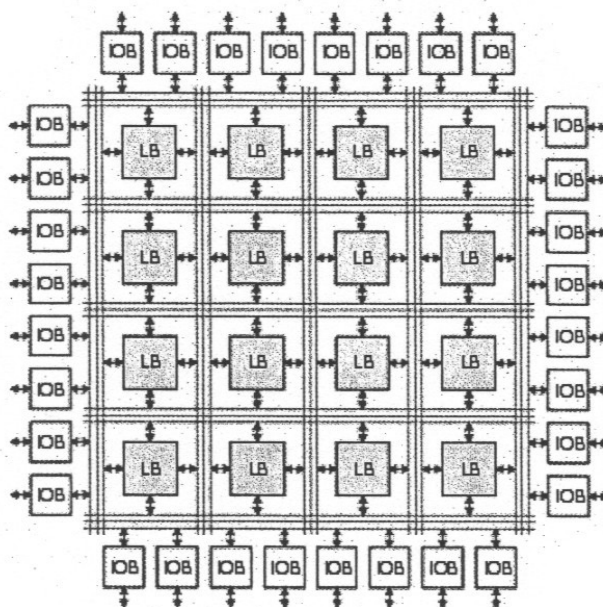
bloky mohou mít vlastní hodinový signál, vlastní nezávislé vstupy nastavení a nulování. Je možné také jejich řazení do kaskád.

Podstatnou novinkou je možnost naprogramovat přiřazení vstupních a výstupních signálů na zvolené vývody. To je spolu s variabilitou ve výběru pouzdra podstatná výhoda pro konstruktéry zařízení, resp. při návrhu plošného spoje. Protože podstatnou částí ceny každého integrovaného obvodu, pro hradlové pole to platí samozřejmě také, je cena pouzdra, zavedli výrobci takový postup, že pro stejný obvod je používáno různé pouzdro. Tedy s více nebo méně vývody. Každý vývod přitom znamená zvýšení ceny. Pokud tedy stačí pouzdro s méně vývody, ale přitom je třeba výkonově použít větší pole, nic nebrání v tom, aby to co není připojeno přímo na vývody, bylo použito jako skryté.

Ještě na jednu okolnost je třeba upozornit. Tyto obvody tradičně používají napájecí napětí 5 V, ale nověji se používá i 3,3 V nebo 2,5 V. Přitom i při nižším napájecím napětí jsou vstupy a výstupy tzv. 5 V tolerantní. Prakticky to znamená, že tyto obvody mohou bez problémů spolupracovat s obvody klasické TTL nebo CMOS logiky.

Obvody FPGA - Field Programmable Gate Array sice v naznačené klasifikaci patří do třetí generace, ale přesto se od CPLD v mnohém odlišují. Zásadní rozdíl spočívá v tom, že růst výkonu u CPLD je dosahován velikostí a složitostí makrobuněk při jejich omezeném počtu, u FPGA je použita cesta malých makrobuněk a výkon je dosahován jejich počtem.

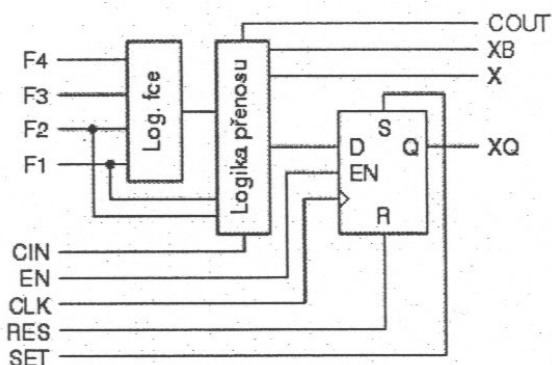
Základní struktura FPGA je dle následujícího obrázku.



Základem je matice logických bloků LB - Logic Block, které mají vstupy a výstupy programovatelně propojeny pomocí Programmable Interconnect. Propojení s vnějším světem pak zajišťují vstupně/výstupní bloky IOB - Input/Output Block. Je třeba mít na paměti, že toto je typická struktura, která se u různých výrobců v detailech liší a to případně i podle různých typů, resp. typových řad. Např. obvody Spartan, jejichž výrobce

je firma Xilinx, obsahují navíc čtyři bloky DLL – Delay Locked Loop, které slouží k rekonstrukci případně násobení či dělení vnějších taktovacích signálů.

Logické bloky jsou obvykle tvořeny dvojicí prvků LUT – Look-up Table, které jsou základem kombinanční struktury pole. Obvykle to bývá malá paměť typu PROM se čtyřmi vstupy doplněná o logiku přenosu a klopný obvod. Obrázek ukazuje typické zapojení prvku.



Již uvedený obvod řady Spartan pak má ve své konstrukci dvojici prvků LUT jako tzv. řez a v logickém bloku jsou dva tyto řezy, tedy čtyři prvky LUT. Některé logické bloky však tvoří specializované obvody typu paměť, registr, sečítačka, násobička atd. Právě tyto specializované bloky tvoří hlavní sílu výkonu FPGA. Jako zajímavost lze uvést, že filosofie malých buněk je využita i u těchto bloků. To znamená, že v obvodu je třeba několik desítek pamětí a ty lze použít nezávisle, případně je i dělit na dvě nezávislé poloviny, nebo naopak je slučovat.

Vstupně/výstupní bloky si můžeme představit jako oddělovací zesilovače, které definují napěťové úrovně dle standardu, tedy dle toho co je na obvod připojen. Můžeme si tak vybrat, zda budeme pracovat se standardem TTL, CMOS 5 V, CMOS 3,3 V, CMOS 2,5 V. To jsou ty nejobvyklejší, ale existuje spousta dalších. Součástí těchto bloků mohou být také registry, ale nikdy kombinační obvody.

Pokud z uvedeného není zřejmé, jsou tyto obvody spolu s moderními mikroprocesory to nejsložitější, co v oboru polovodičových součástek existuje. Navíc vývoj jde cestou zvyšování výkonů, tedy stále větší složitosti. To už samozřejmě není možné zvládnout klasickými postupy. Představa, že by se FPGA řešil ručně pomocí Karnaughových map, je víc než úsměvná. Prakticky k tomu existují speciální programy, tzv. **vývojová prostředí**. Je jistou komplikací, že každý výrobce má toto prostředí vlastní a s ostatními zcela nekompatibilní. Ve vývojovém prostředí máte možnost napsat pomocí VHDL celý program pro chování obvodu, jste v něm upozorněni na chyby, provedete si simulaci, připojíte vývody a nakonec nahrajete program do obvodu. To vše už je však nad rámec tohoto materiálu.