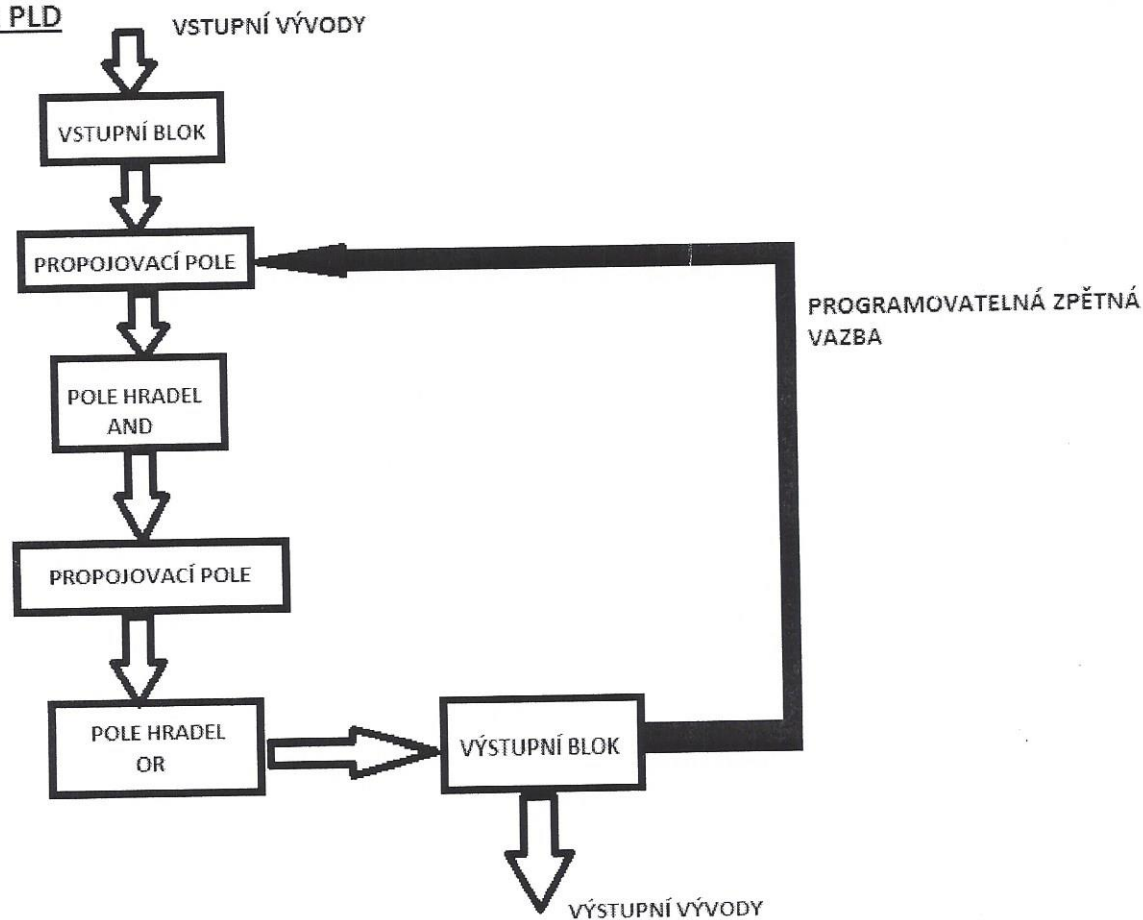


PLD

- číslicové obvody, které lze konfigurací naprogramovat
- programování probíhá vytvářením nebo přerušením propojek nebo paměťových buněk
- nahrazují kombinační a sekvenční logické obvody sestaveny z obvodu střední integrace (hradla, čítačky)
- PLD obvody představují prostředek pro návrh kombinačních a sekvenčních obvodů a díky vývoji prostředků, které lze pro práci s nimi použít umožňuje podstatným způsobem usnadnit práci návrháře
- filozofie PLD obvodu vychází z faktu, že jakoukoliv log. fci a log. rovnice lze vyjádřit pomocí součtu součinu boulovských proměnných => lze implementovat prostředí vhodného zapojení hradel AND a OR

ARCHITEKTURA PLD



- vstupní blok: - vyrovnávací paměť

- vstupní budiče

výstupní blok: - může být tvořen registry nebo může obsahovat poměrně složitou a dále programovatelnou logiku

- výhody: - snížení počtu integrovaných obvodů

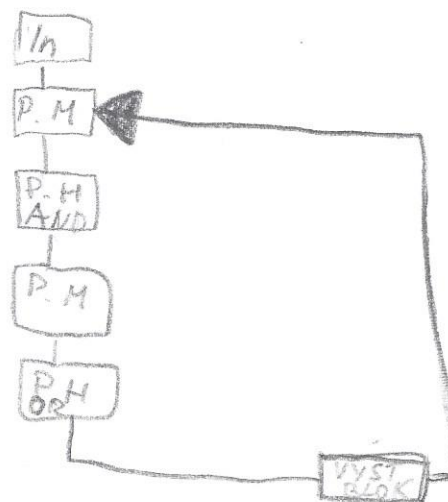
- snížení pracnosti výroby

- zvýšení spolehlivosti

- mikrokontrolery

- možnost snadné modifikace obvodů

- nevýhody: - musí se vyrábět sériově



PLD typu GAL (gate array logic)

- patří do skupiny E²PLD (elektronický mazatelné i programovatelné)

GAL 20V10

- nástupce PAL (80. léta)

GAL NAHRADIL AŽ 20 TYPŮ PAL:

- programovatelné propojovací AND a pevné pole hradel OR

- pro složitější úlohy je používán GAL 22(počet pinů)V10(počet pinů, které mohou být konfigurovány jako výstupy) postačí GAL 16V8

POČET PINŮ
MOHOU BÝT
JAKO VÝSTUPY

- skládají se z propojovacího pole, do kterého jsou napojeny hradla AND a OR, pomocí této sítě lze realizovat sekvenční obvod

- podstatou propojovacího pole je pam. buňka v technologii E²PROM (1 tranzistorový spínač)

- obrázek v sešitě

PARA
SERIO-PARA

Makro-buňky OLMC (output logic macro cell)

- každá makro-buňka může být nakonfigurována do kombinační nebo reg. fce

- základní částí bývá paměťový člen KO-D, který je doplněn o řadu paměťových hradel

- podoba makro-buněk se v každém obvodu liší, obsahuje několik konfiguračních míst

- makro-buňky se můžou v závislosti na prog. nacházet v několika režimech (i 6 režimu)

- 6. režimu:

- 1. registrový mód - vývody 1-11 jsou nastaveny jako vstupy CLK a ovladač tří stavového výstupu

- 2. kombinační výstup - komplexní mód



- vývody 1-11 plní fce vstupu OLMC, lze nastavit jako kombinační vstupy a výstupy k dispozici je 7 součinných termů, osmý slouží pro řízení 3 stavového výstupu
- **3. kombinační v registračním kódu**
- **4. kombinační výstupní režim se zpětnou vazbou**
- **5. kombinační výstup a jednoduchý mód**
 - vývody 1-11 mají fci vstupu
 - OLMC lze kombinovat jako vstup nebo výstup
 - k dispozici je 8 součinných termů, protože výstupy nelze 3 stavově ovládat
- **6. dedikovaný vstup a jednoduchý výstup**

Metodika návrhu

- vytvořit schem. editor, programovací jazyk (VHDL, very LOG, abel VHDL ad.), editor pro stavové diagramy
- postup:
 - formální zápis
 - určení vstupu a výstupu
 - formální popis problému (boulowské rovnice, funkční a přechodové tabulky, orientovaný graf, schéma zapojení, časové průběhy, minimalizace)
 - simulace
 - realizace
- při návrhu je velmi důležitá simulace nebo průběžné testování, zde se odladí nejvíce chyb aniž by bylo nutné využít skutečný obvod
- toto několikrát zopakujeme tzv. lazení (od textového zápisu po samotnou simulaci)

PLD kompilátor

- podobně jako překládač jazyku vyšší úrovně převádí zdrojový text do cílového strojového kódu procesoru, tak i PLD kompilátor převádí definovanou log fci do implementovaného prostředí konkrétního PLD obvodu
- vývojové systémy umožní definovat návrh číslicového obvodu, bez ohledu na konkrétní typ PLD obvodu, který bude použit pro cílovou implementaci
- výstupem PLD kompilátoru je soubor **jedek** (jed), který se programuje do konkrétního PLD obvodu a je to definiční rozhraní mezi vývojovým systémem a konkrétním PLD obvodem
- zápis programu je obvykle ve vyšším programovacím jazyce
- kompilér provede transformaci zápisu, včetně minimalizace => ta maximálně zjednoduší návrh, výsledkem je ještě dál optimalizace => přizpůsobení log struktury pro konkrétní log obvod
- zkompileovaný program je pak možné převést do přípravku přes programátor, je paralelní, kdy se vše "vypálí" a my můžeme používat přípravek
- optimalizace je proces pro přizpůsobení boulovských rovnic, pro co nejefektivnější využití vlastností zvolené architektury PLD obvodů => minimalizace spotřeby vstupu, výstupu, vnitřních termů a makro-buněk

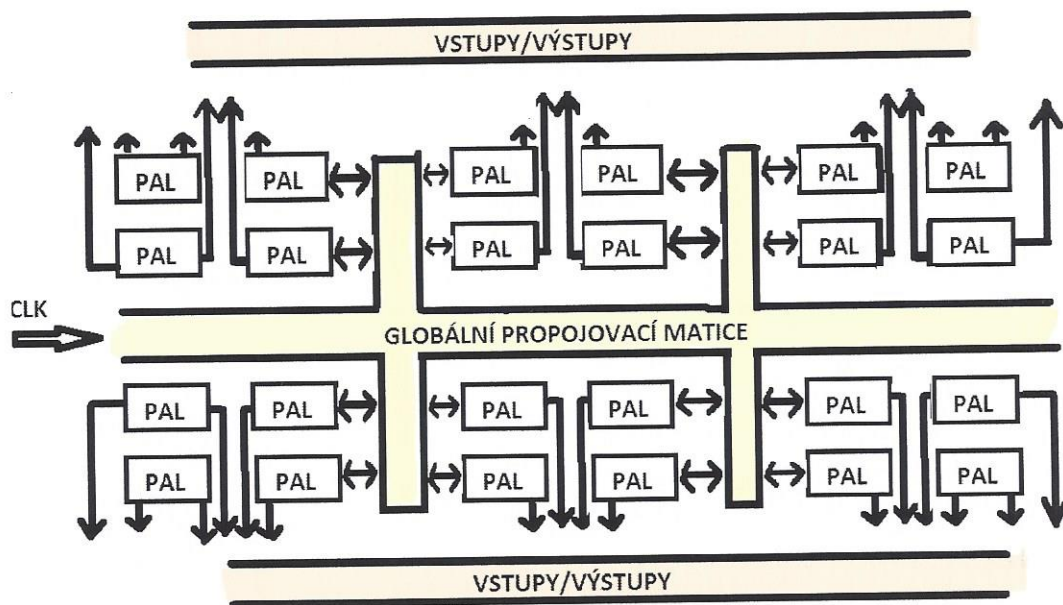
Obvody CPLD a FPGA

CPLD - součástky komplexnější než GALy

- větší možnost logiky
- charakterizovaný počet makro-buněk
- sdružují více obvodu typu PAL (GAL) na jednom čipu spolu s nutnými prostředky pro propojení
- až na pár speciálních pinů mohou být všechny konfigurace jako vstupní
- hodí se pro jednodušší aplikace
- výrobci: Lattice, Xilinx, Altera
- základní části: - vstupní a výstupní blok
 - makro-buňky rozdělené do skupin ve funkčních blocích
 - rozhraní jtag pro programování v systému
- makro-buňka: - obsahuje matici pro realizaci fce AND
 - obsahuje generátor logické fce (tzv. produkterm alokátor)
 - každá obsahuje 1 klopný člen KO-D

- umožňuje možnost způsobu ovládání paměťových členů

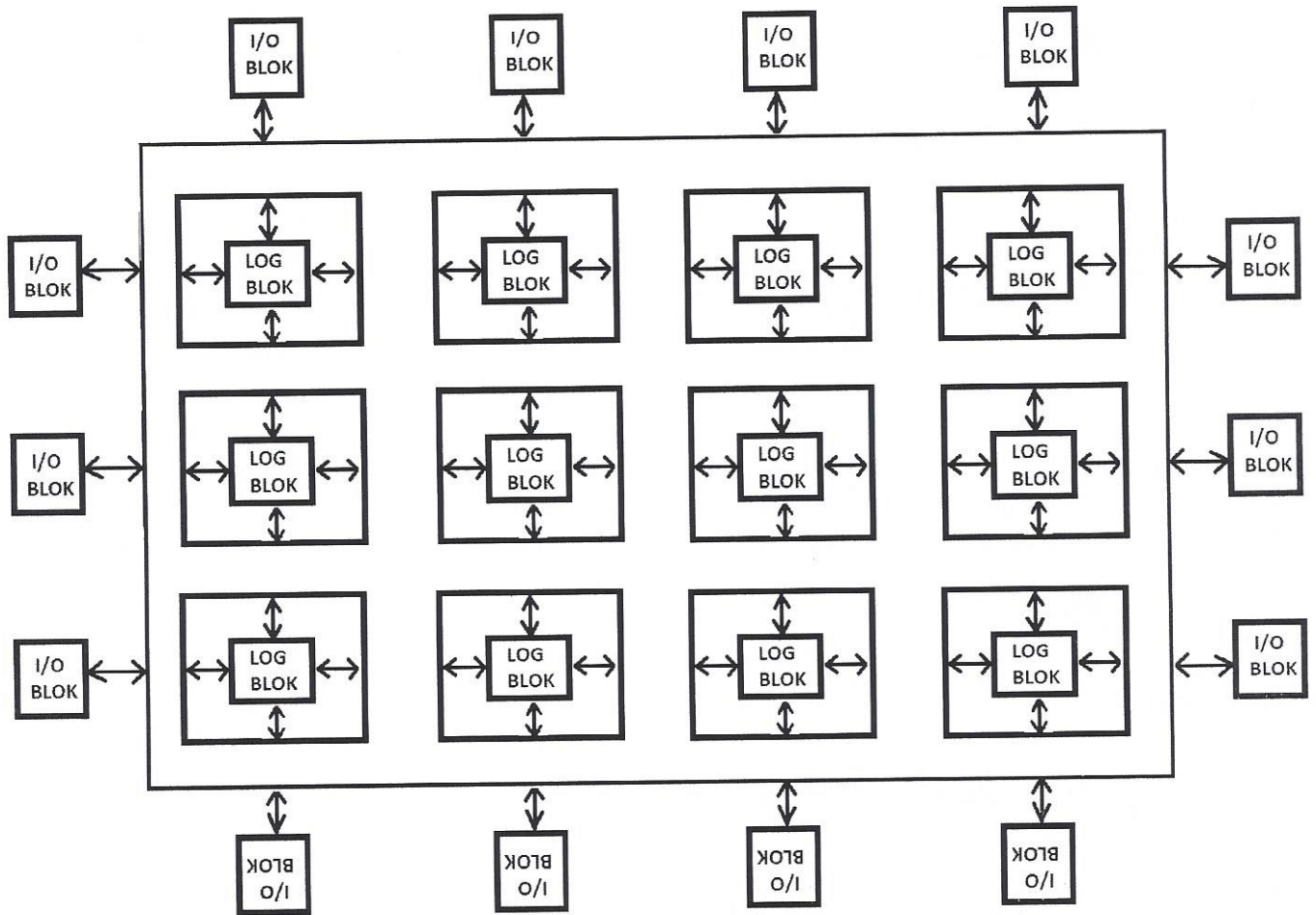
Struktura PLD



FPGA - programovatelné logické pole s jednoduchými funkčními bloky

- od firmy Xilinx
- je zde možno naprogramování (přeprogramování)
- obsahují až 25 tisíc hradel a až 2560 KO
- taktovací frekvence až 125 MHz
- jsou nejsložitější, ale zároveň nejobecnější PLD obvody
- místo makro-buněk obsahují tzv. logické bloky a obsahují až milióny ekvivalentních hradel (dvouvrstvé hradla AND)
- jednotlivé log bloky jsou propojeny globální propojovací maticí
- kromě základních bloků vstupu, výstupu a logických bloků obsahuje většina obvodu FPGA rychle statické RAM a SRAM (až speciální bloky) a hardwarové násobičky
 - pro svou konfiguraci mají integrovanou paměť RAM => po každém zapnutí nutné nakonfigurovat, vy užití externí EEPROM paměti => po zapnutí nakonfigurují RAM

Struktura obvodu FPGA



PROM - AND - pevně OR - pg

PAL - AND - pg

OR - pevně

PLA

Dodatek ke 4 – GAL, PAL, PLA a další pičoviny

PLD obvody

-založeny na dvoustupňové struktuře realizace kombinačních logických funkcí

-programovatelné prvky struktury realizovány bloky AND a OR, prostě buď programuju pole AND a OR je napevno nebo obráceně

Struktury PLD obvodů:

- PROM – stupeň AND zapojen pevně, struktura OR programovatelná
- PAL- stupeň AND programovatelný a stupeň OR pevně
- PLA – stupeň AND a OR programovatelné
- programovatelný stupeň realizuje ve formě matice programovatelných spínačů
- pevný stupeň zapojen jako logický člen

PROM

- používá v pamětech se stejným označením
- v sérii s každou diodou zapojena propojka, která se programováním přeruší nebo ponechá mintermy jejichž propojka byla přerušena, se ve výsledku neobjeví
- funkce realizována a základě úplné součtovém tvaru a není potřeba minimalizace
- universální, lze realizovat jakoukoliv funkci
- paměti EPROM – mazání světlem, EEPROM – mazání elektronicky

PAL

- obdobné zapojení jako pro PROM
- programovatelné AND
- funkce se realizuje na základě minimálního tvaru zápisu
- oproti PROM menší universálnost
- nejčastěji používána struktura
- odstraňuje neuspornost struktury PROM
- struktura vytváří log součiny

PLA

- nejúspornější řešení
- součinnové termy mohou být současně využity pro více vytvořených funkcí
- nevýhodou velký počet programovatelných propojení
- všechny AND a OR programovatelné