

6. Programovatelné logické obvody

Definice programovatelných hradlových polí (FPGA)

Programovatelná hradlová pole je v elektronice typ logického integrovaného obvodu, který je vyroben tak, aby mohl být naprogramován až u zákazníka.

Obsahuje pole programovatelných logických obvodů (PLD), logických bloků, umožňuje je navzájem propojit a tím vytvořit takřka libovolné číslicové zařízení (například mikroprocesor, řídicí obvod síťové karty a podobně). Tím se odlišuje od zákaznických integrovaných obvodů (ASIC), jejichž funkce je dána již při výrobě.

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

Logický obvod a jeho odvození z pravdivostní tabulky (mintermy, maxtermy) nebo z diagramu

Máme následující funkci:

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

$$F = (a \wedge b) + 0 + 0 + (a \wedge b) = (a \wedge b) + (a \wedge b)$$

Tato logická funkce je dosti jednoduchá takže nevyžaduje komplexní řešení za pomoci mintermů/maxtermů a dále nejde zjednodušit.

Způsoby programování FPGA

SRAM

Programovací buňka založená na technice SRAM se skládá ze šesti tranzistorů, tvořících statickou paměťovou buňku, která řídí programovatelný spínač, následně konfiguruje funkci logického bloku nebo topologii propojovacích vodičů. Tato metoda se vyznačuje možností opakovaného konfigurování FPGA, v některých případech dokonce za chodu. Funkci buňky lze plně ověřit při výrobě FPGA.

Programovací buňka SRAM má mnohem větší rozměry než antipojistka, nicméně, jak již bylo uvedeno, technologie výroby FPGA sleduje technologii výroby SRAM. Jelikož při ztrátě napájecího napětí se u této metody ztrácí informace o konfiguraci, musejí být tato FPGA doplněna trvalou, nezávislou pamětí o konfiguraci.

Antipojistka

Pokud projde antipojistkou vysoký proud tak pojistka sepne jinak je v neseprnutém stavu. Antipojistka se vyznačuje trvalým naprogramováním, malým odporem přechodu, malými rozměry a úplnou testovatelností při výrobě nebo v programátoru.

Její nevýhodou je, že je naprogramována jednou provždy.

Naopak potenciální výhodou tohoto způsobu je, že výsledný obvod neobsahuje paměť se souborem konfiguračních bitů (configuration bitstream), což je důležité z hlediska obrany proti tzv. zpětnému inženýrství.

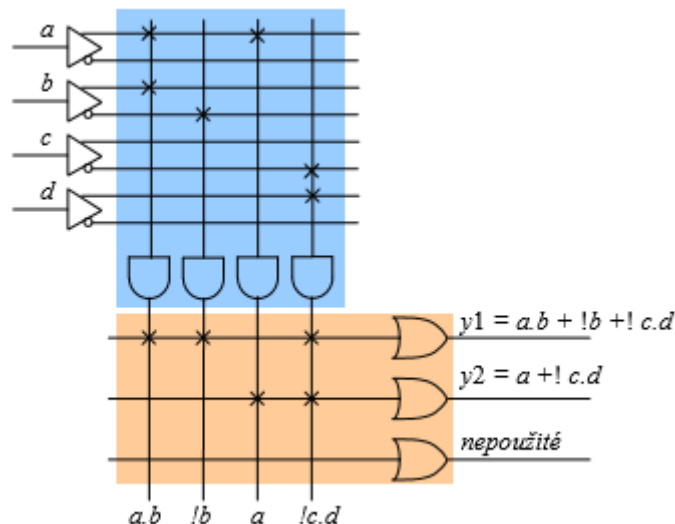
PLA (Programmable Logic Array)

Obvody PLA byly jedny z prvních, které se v minulosti začaly vyrábět. Avšak z trhu je vytlačily obvody PAL. Dnes se s principem PLA můžeme setkat u některých CPLD obvodů

Obvody PLA ze skupiny PLD, která vychází z teorie programovatelných matic AND a OR. Obě matice jsou programovatelné.

Významy:

- Možnost skupinové minimalizace, tím že jeden součinný term mohl být použit ve více výstupních funkcích.
- Počet součinných termů pro výstupní funkci není omezen jako u obvodů PAL.



Obvody PLA realizují zapojení kombinační sítě, které vychází z DNF – součtové normálové formy.

V programovatelné součtinové matici se vytvářejí jednotlivé součtinové termy ze vstupních signálů v přímém a negovaném tvaru.

V programovatelné součtové matici se potom provede součet jednotlivých součtinových termů.

Na obrázku je PLA struktura určená pouze pro logickou kombinační síť. Dalším pokračováním je umístění paměťového elementu za součtové hradlo, a tím možnost vytvářet logické sekvenční AND a OR matice obvody

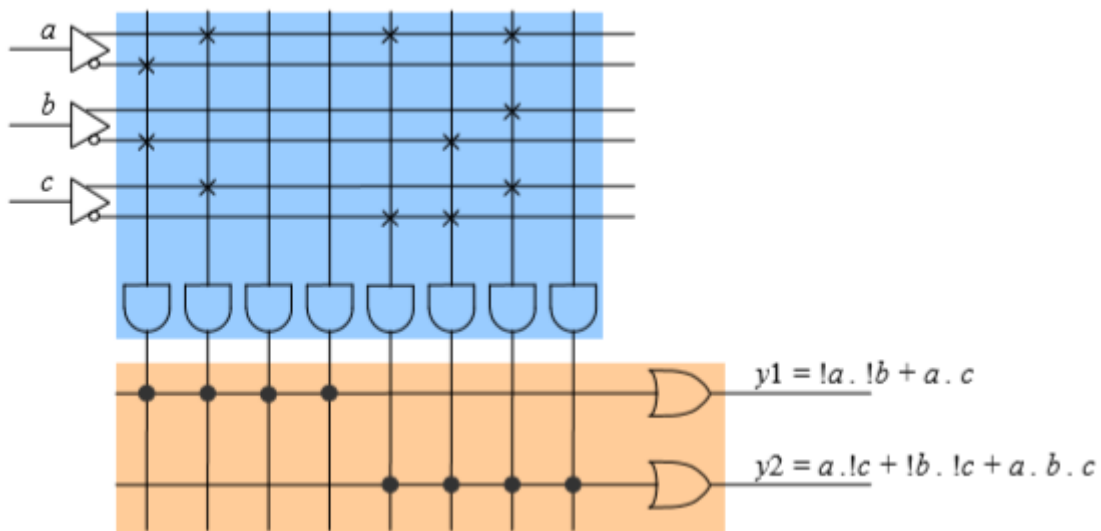
PAL (Programmable Array Logic)

Obvody PAL jsou variantou aplikace součtinové a součtové matice. V době jejich vzniku doznaly velkého praktického rozšíření a byly jednou programovatelné. Obvody PAL byly časem nahrazeny obvody GAL.

Obvody PAL, Programmable Array Logic je skupina obvodů vycházející z teorie matic AND a OR, kde matice AND je programovatelná a matice OR je pevná.

Významy:

- Význam PLA obvodů je dnes spíše historický, protože jako samostatná skupina se v minulosti neprosadily.
- Byly úspěšnější než PLA.
- Původně byly PAL obvody jednou programovatelné, dnes se objevují i reprogramovatelné a hlavně jsou nahrazeny obvody GAL.



Podle obrázku může každý výstup použít do logického součtu maximálně 4 součtinové termy. Výraz $y1$ má pouze dva součtinové termy, v zapojení se použijí pouze dvě součtinová hradla a dvě zůstanou nepoužita. Nepoužitá součtinová hradla neovlivňují výsledný výraz $y1$.

GAL (Generic Array Logic)

Obvody typu GAL, Generic Array Logic jsou obvody, které navazují na obvody PAL a rozšiřují je o dvě významné vlastnosti.

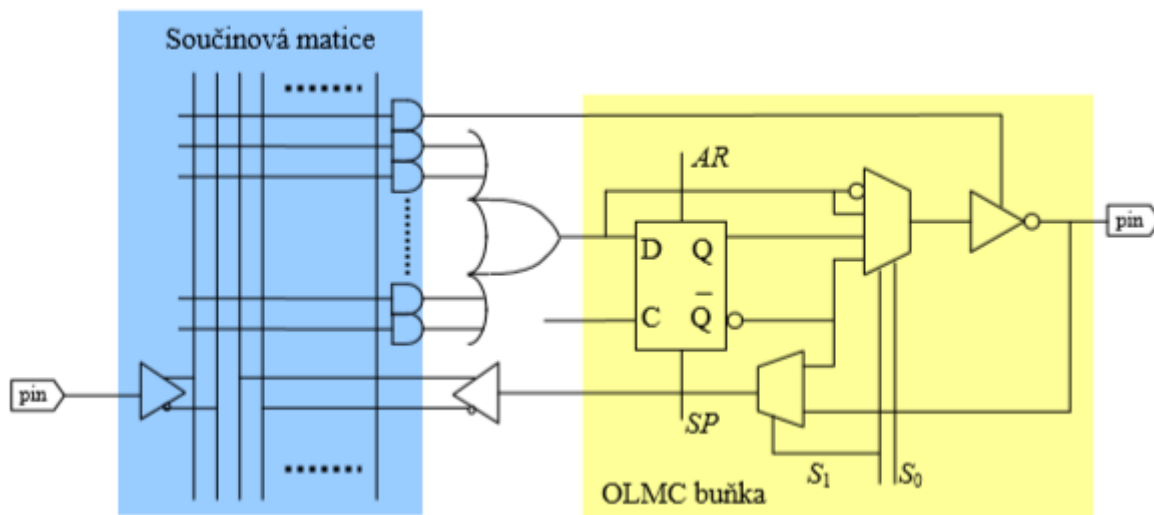
První vlastností je reprogramovatelnost GAL obvodů, čímž vzniká skupina elektricky mazatelných obvodů.

Druhou vlastností je rozšíření používání OLMC Output Logic Macro Cell, která umožňuje jeden obvod využívat jako kombinační nebo s paměťovými elementy.

Obvody GAL vycházejí z obvodů PAL, mají programovatelnou matici AND a pevnou matici OR a na výstup součtového hradla je zařazena OLMC.

Významy:

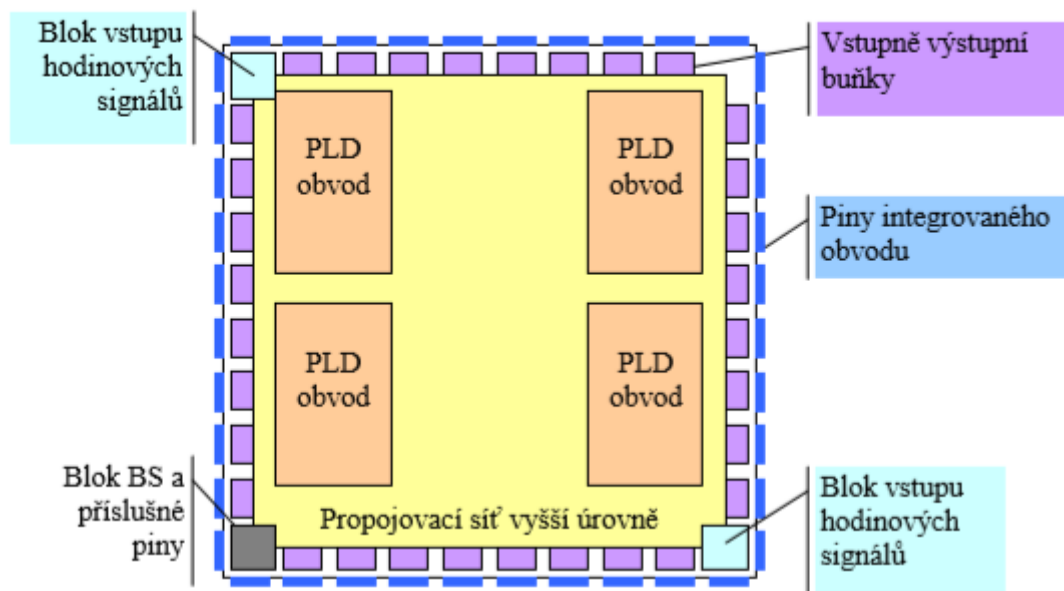
- Z aplikace OLMC buňky vyplývá možnost použití pouze jednoho typu GAL obvodů pro všechny PLD prvky na desce plošného spoje. Je to zmenšení počtu typů součástek na desce plošného spoje.
- Zavedení opětovného programování má význam jednak pro návrháře ale i pro výrobce. Reprogramovatelnost GAL obvodů umožňuje pozdější zavádění změn zapojení GAL obvodů. Je to významný rozdíl vůči PAL obvodům.
- Aplikace elektronického podpisu, který umožňuje identifikaci obsahu GAL obvodů.



U GAL obvodů je aplikován bezpečnostní bit, Security Fuse, který zabraňuje neautorizovanému čtení obsahu GAL obvodu. Bezpečnostní bit se dá vymazat pouze s vymazáním celého obsahu GAL obvodu.

CPLD (Complex Programmable Logic Device)

CPLD obvody lze jednak chápat jako další vývojový stupeň obvodů PAL a GAL, zvyšování integrace, protože nejdříve začínaly na principu AND a OR matice. Dnes přebírají některé prvky dříve typické pro FPGA, hlavně uplatněním buňky s LUT tabulkou.



Nejzákladnější popis CPLD obvodu je nejlépe při pohledu na křemík, jedná se o řez integrovaným obvodem.

Základ CPLD prvku tvoří PLD obvody, IO buňky a propojovací síť vyšší úrovně.

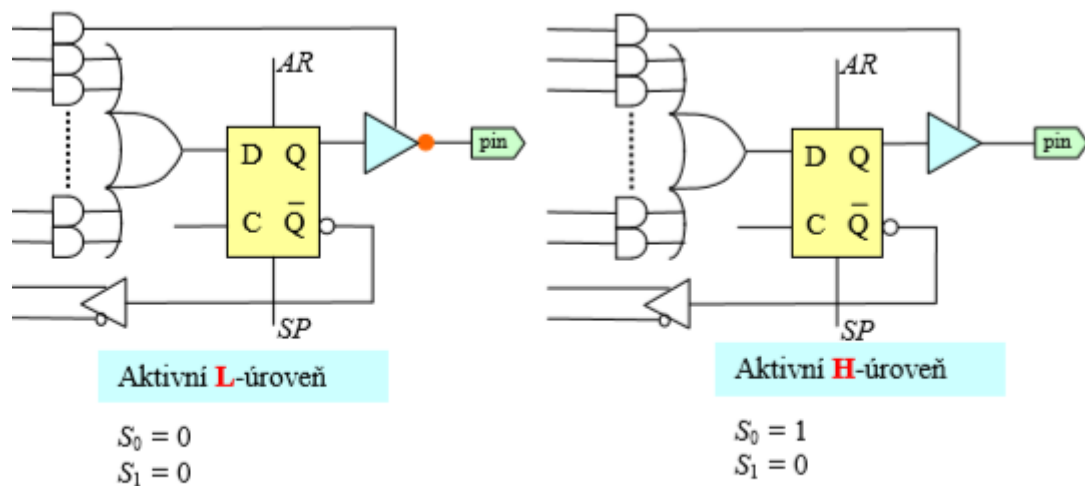
PLD obvod je v převážně typu GAL s OLMC buňkou. Dnes se na tomto místě objevují i zapojení pomocí buněk s LUT, které jsou známe s FPGA obvodů.

PLD obvod má vlastní programovatelnou propojovací síť a je možné v něm vytvářet zapojení logické sítě. Propojovací síť PLD obvodu nazýváme propojovací sítí nižší úrovně. Druhá také programovatelná propojovací síť je propojovací síť vyšší úrovně, která propojuje jednotlivé PLD obvody mezi sebou, a tím umožňuje vytvářet logické sítě větší složitosti.

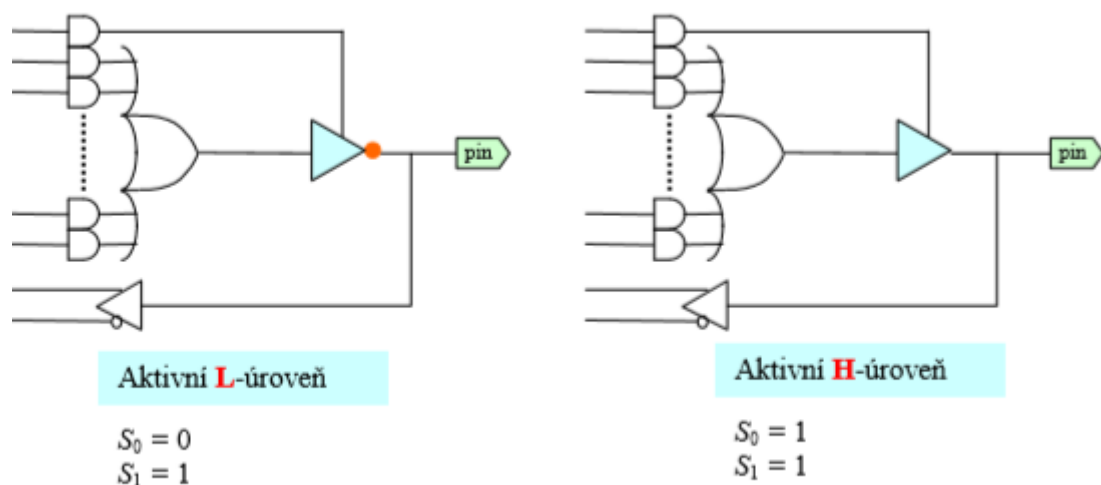
Základní rozdíl mezi jednotlivými úrovněmi propojovacích sítí je ve zpoždění signálu. Zapojení, které využije pouze propojovací síť PLD obvodu bude mít menší zpoždění než zapojení, které využije i propojovací síť vyšší úrovně.

Makrocell/OLMC

OLMC buňka je konfigurovatelná pomocí programovatelných propojek S0 a S1, které umožňují konfigurovat buňku jako kombinační nebo registrový výstup. Jedná se o výstup s paměťovým elementem nebo jako kombinační. Další rozdíl je v aktivitě výstupu, v přímém tvaru výstup nemá invertor a v negovaném tvaru jej obsahuje.



Obr. 6 - 2 Možné zapojení OLMC buňky s registrem na výstupu



Obr. 6 - 3 Možné zapojení OLMC buňky jako kombinační výstup

Jako paměťový element je většinou D klopný obvod. Signál AR – Asynchronous Reset je asynchronní nulování a signál SP je Synchronous Preset – synchronní nastavení. Signály AR, SP a CLK mohou být odvozeny od pinu nebo generovány jako součin v programovatelné AND matici.

Současná nabídka FPGA

V současnosti FPGA vyrábějí zejména firmy Actel, Altera, Atmel, Lattice a Xilinx. Zde se soustředíme na dvě z nich, na firmy Altera a Xilinx.

Oba výrobci nabízejí jak obvody zohledňující nízkou cenu řešení (Altera Cyclone/CycloneII, Xilinx Spartan-2/Spartan-3), tak obvody s velkou výkonností pro technicky náročné úlohy (Altera APEX/Excalibur Xilinx Virtex2/Virtex4).

Programování Výstupů

K popisu hardwarové struktury PLD obvodů slouží v dnešní době vyšší programovací jazyky. Těch existuje několik, jmenujme například Verilog, ABEL nebo VHDL.

VHDL původně vzniklo jako jazyk pro simulace hardwaru. Až později se začal používat k popisu reálných obvodů. Jeho výhodou je hardwarová nezávislost – můžeme psát kód ve VHDL a simulovat jej, aniž bychom předem věděli jaký hardware nakonec použijeme. Jeho nevýhodou je ale relativně velká „ukecanost“ a časté opakování stejných bloků.