

# 3. PLD & NPLD

- 01** Charakteristika, vznik a rozdělení PLD
- 02** Popis dílčích PLD obvodů
  - a)** PLA, PAL, GAL, CPLD, FPGA
- 03** Makrobuňka vs. logický blok
- 04** Vývojové prostředí WebPack
  - a)** PLD kompilátor
- 05** Charakteristika NPLD
- 06** Popis a využití vybraných obvodů
  - a)** 74164, 74166, 74595, 74573,  
74244, 74245, 74688, 74193

### 3. PLD & NPLD

## PLD – Programovatelné logické obvody (Programmable Logic Devices)

### Definice a základní vlastnosti

PLD je **číslicový integrovaný obvod**, jehož vnitřní strukturu lze uživatelsky konfigurovat – tedy „**naprogramovat**“ tak, aby vykonával konkrétní logickou funkci.

### Programování se provádí:

- **vytvářením nebo přerušením propojek** v místě křížení vodičů (např. pomocí pojistek),
- nebo **zápisem dat** do programovatelných paměťových buněk (např. EEPROM, Flash).

### Využití

- Nahrazují klasické kombinační i sekvenční logické obvody (ty by jinak byly tvořeny z hradel, čítačů, registrů atd.). - sestaveny z **obvodů střední inteligence**
- Umožňují **návrh a realizaci vlastních číslicových obvodů** bez nutnosti použít mnoho diskrétních součástek.

### Základní princip návrhu

- Logické funkce v PLD se obvykle zapisují pomocí **Booleových rovnic ve tvaru součtu součinů** (tzv. disjunktivní normální forma).
- Tyto rovnice jsou pak **implementovány jako zapojení hradel** uvnitř PLD.

### Struktura PLD

- Na čipu se nachází matice logických prvků, které lze **konfigurovat**:
  - Prvky jsou propojeny pomocí **programovatelné mřížky vodičů** (propojovací síť).
  - Tato mřížka obsahuje desítky **horizontálních a vertikálních vodičů**, které se kříží.
  - V místech křížení jsou programovatelné spoje, které lze **softwarově aktivovat nebo deaktivovat**, čímž určujeme, zda dané prvky budou propojeny.

### Výhody PLD

- **Přehlednost návrhu** – vše je v jednom čipu.
- **Snížený počet vývodů** (pinů) oproti použití samostatných logických obvodů.
- **Vyšší spolehlivost** – méně součástek, méně spojů.
- **Snadná modifikace návrhu** – úprava pouze změnou konfigurace, nikoli hardwaru.
- **Vysoká výkonnost** – rychlá odezva díky optimalizaci struktury.

### Nevýhody PLD

- **Nutnost ovládat** specializovaný **programovací jazyk** (např. VHDL nebo Verilog).
- Porucha čipu znamená **výměnu celého obvodu**, nikoliv jen jeho části.
- **Vyšší cena** oproti jednotlivým klasickým logickým obvodům (u malých projektů).

### Nejvýznamnější výrobci PLD

- Xilinx
- Altera (dnes součást Intelu)
- Lattice Semiconductor

# 3. PLD & NPLD

## Programovací jazyky

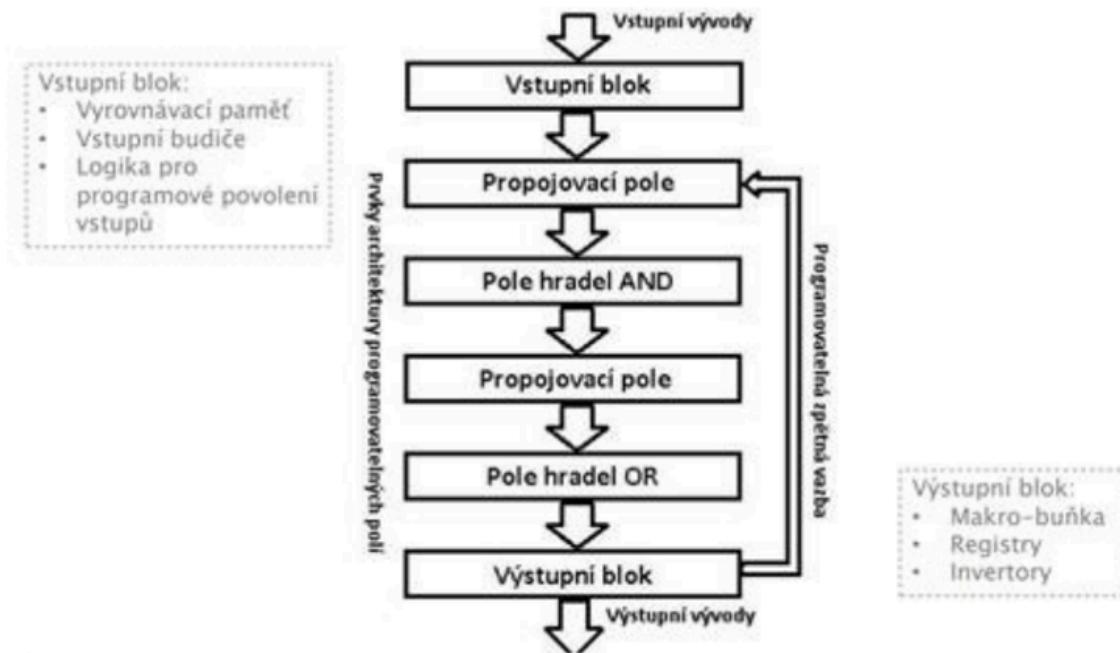
Programování PLD se provádí pomocí jazyků pro **popis hardware** (*HDL – Hardware Description Language*):

### VHDL (VHSIC Hardware Description Language)

- **Otevřený standard** – není třeba žádná licence.
- **Nezávislý na konkrétním čipu** – lze navrhnout obvod a cílový čip zvolit až později.
- **Podporuje simulaci** – umožňuje testovat funkčnost návrhu ještě před fyzickým nahráním do čipu.

### Verilog

- **Alternativa k VHDL**, syntaxí připomíná jazyk C.
- Obecně **považován za jednodušší** pro začátečníky, ale méně přesný než VHDL.



## Rozdělení PLD (Programmable Logic Devices)

Programovatelné logické obvody se dělí podle složitosti na **SPLD** a **CPLD**:

### SPLD – Simple Programmable Logic Devices

Základní typ PLD obvodů, které **obsahují jednoduchou programovatelnou logickou strukturu**, typicky **jedno pole AND hradel a jedno pole OR hradel**. Jsou určeny k realizaci **jednodušších logických funkcí**.

#### Obecné vlastnosti SPLD:

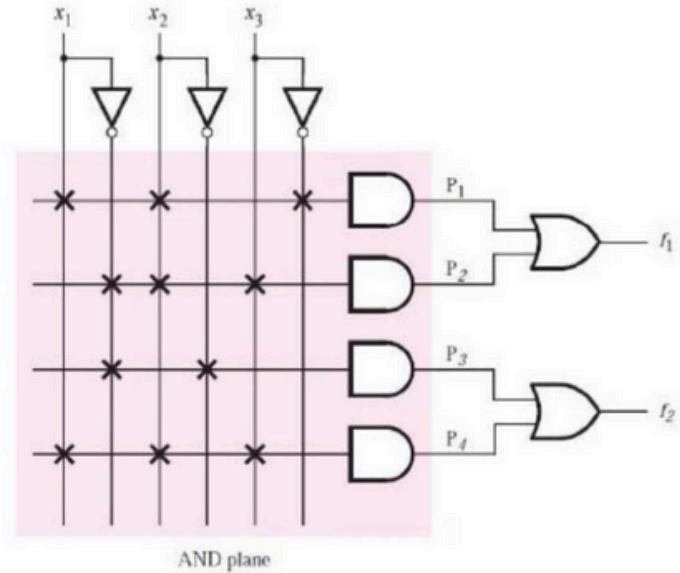
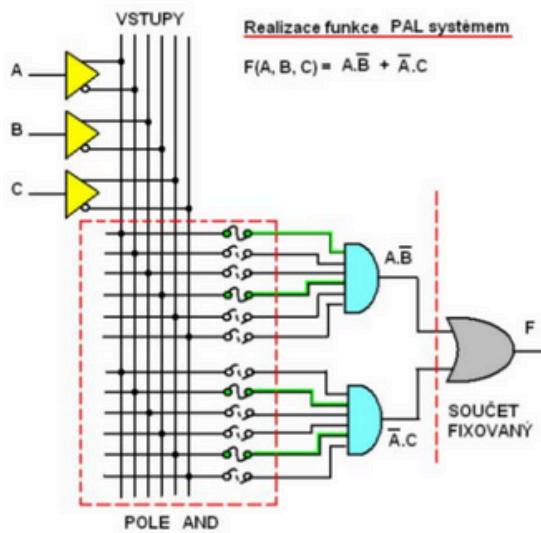
- Každá **vodorovná čára** v programovatelné matici **AND** odpovídá jednomu **součinovému (AND) hradlu**.
- Na výstup každého hradla lze připojit libovolnou kombinaci vstupních signálů, zpětných vazeb a jejich negací.
- Počet vstupů každého součinového hradla je omezen technologií čipu.
- Využívají se v PAL, PLA a GAL strukturách.

# 3. PLD & NPLD

## Typy SPLD

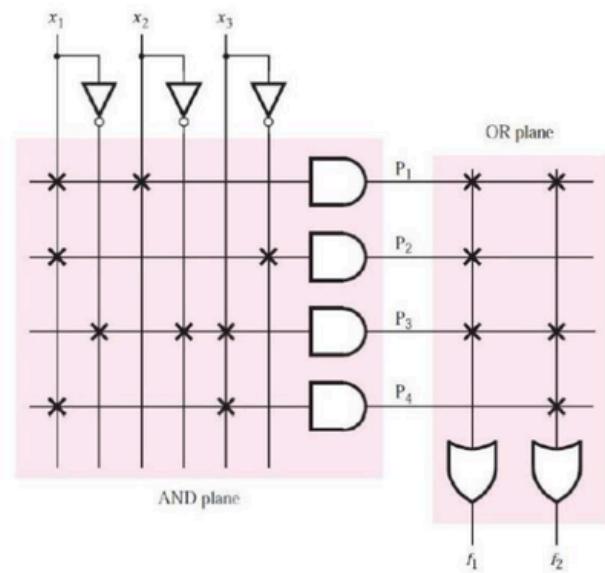
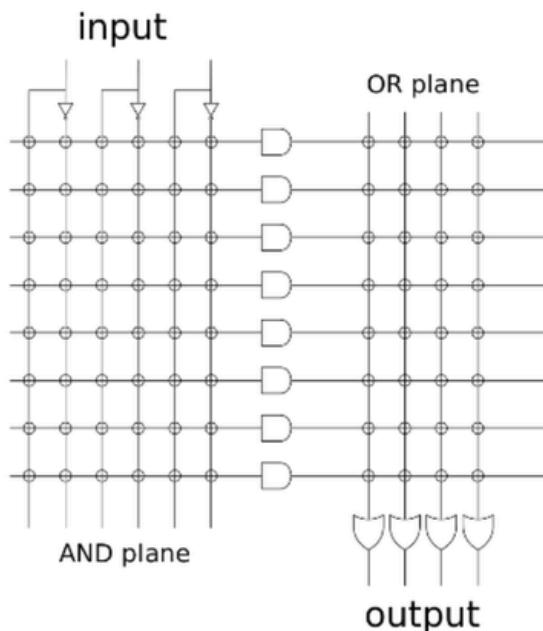
### PAL – Programmable Array Logic

- Obsahuje **programovatelné pole AND** a **pevně zapojené OR hradla**.
- Výstupy jsou fixně připojeny k OR hradlům → **omezená flexibilita**.
- Umožňuje realizovat **pouze funkce s omezeným počtem součinových členů** (např. max. 8).
- Výhoda:** rychlejší než PLA kvůli jednodušší struktuře.



### PLA – Programmable Logic Array

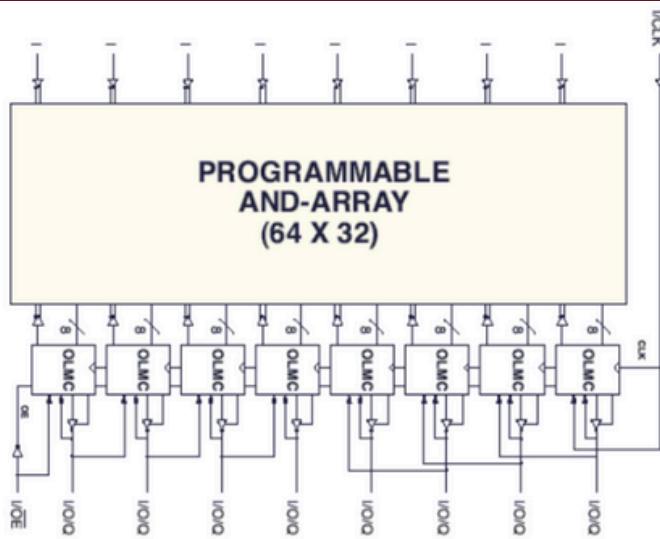
- Má **programovatelné pole AND i OR hradel**.
- Větší flexibilita** při návrhu logických funkcí.
- Vhodné pro implementaci **více logických rovnic** s různými vstupními kombinacemi.
- Např. pro **N** vstupních proměnných může mít až  **$2^n$  hradel AND**, a pro **M** výstupů je potřeba **M hradel OR** (každé s *programovatelným vstupem*).



# 3. PLD & NPLD

## GAL – Generic Array Logic

- Modernější alternativa k PAL.
- Umožňuje **mazání** a **přeprogramování** (typicky pomocí EEPROM technologie).
- Vhodné pro **vývojové účely** – více použitelný než PAL.
- Stále **relativně jednoduchý** a levný.



## CPLD – Complex Programmable Logic Devices

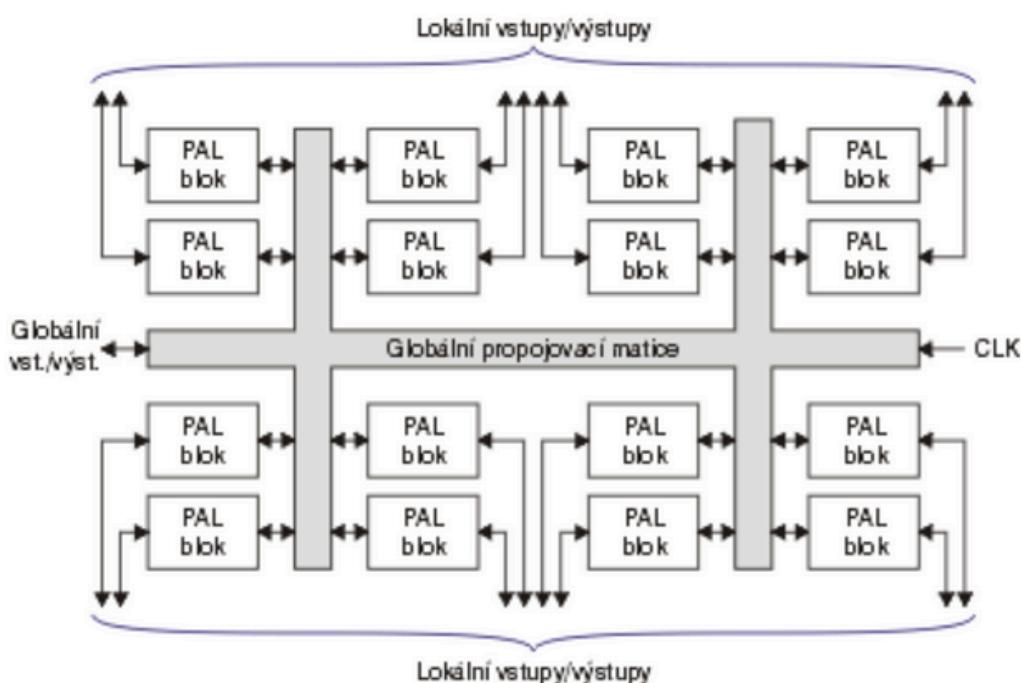
CPLD představuje **pokročilejší typ PLD s vyšší kapacitou a komplexností**.

### Vlastnosti CPLD:

- Spojuje více SPLD bloků (tzv. makrobuněk) do jednoho čipu.
- Umožňuje realizovat složitější logické funkce a i části sekvenčních obvodů.
- Počet makrobuněk: v řádech stovek, propojených pomocí interní propojovací logiky.
- Většina vstupní/výstupních pinů je univerzálních – lze je nakonfigurovat dle potřeby.
- Vhodné pro náročnější digitální systémy, často používané v průmyslových aplikacích.

### Příklad CPLD:

- COOLRUNNER (od Xilinx) – známý **nízkou spotřebou** a použitím ve **vestavěných systémech**.



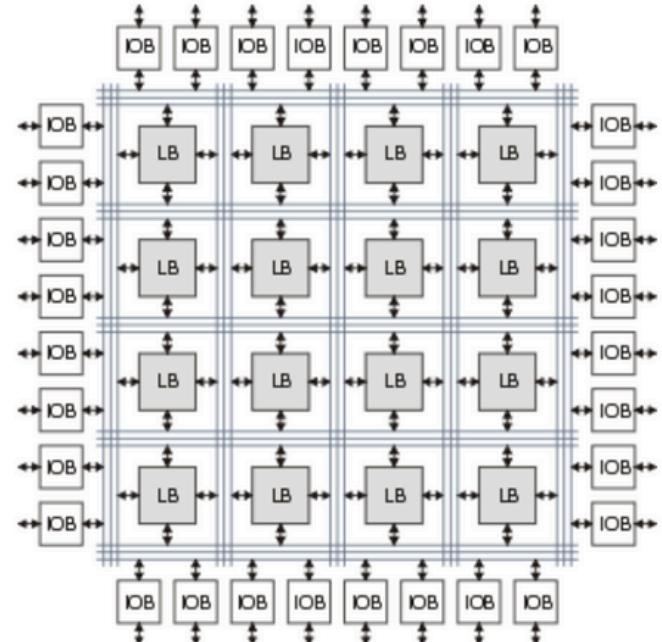
# 3. PLD & NPLD

## FPGA – Field Programmable Gate Array

FPGA jsou **nejvýkonnější a nejvícestrannější programovatelné logické obvody**. Umožňují realizaci **velmi složitých logických obvodů** díky své flexibilní a rozsáhlé architektuře.

### Vlastnosti a struktura FPGA

- Obsahují **největší množství logiky** ze všech PLD typů → až **miliony** logických hradel (např. AND).
- **Základní stavební prvky** nejsou makrobuňky jako u CPLD, ale tzv. **logické bloky (LB)**, konkrétně **CLB – Configurable Logic Block**.
- Vnitřní logika je **propojena pomocí globální propojuvací maticy (GPM)**, která umožňuje **flexibilní směrování signálů** mezi jednotlivými bloky.

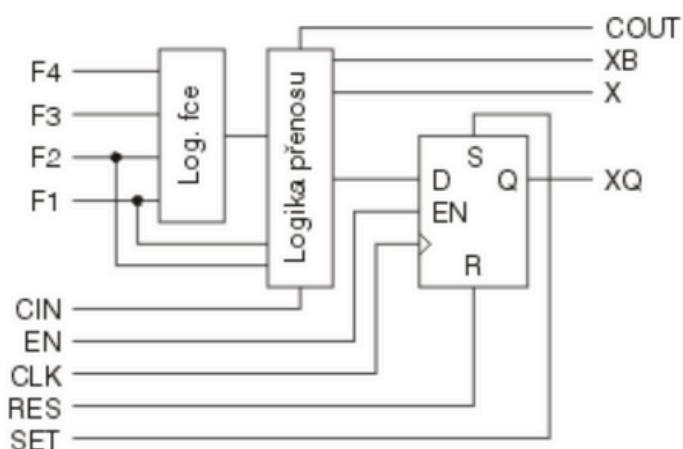


### Přímé propojení sousedních bloků

- Některé sousední logické bloky **lze propojit bez použití GPM**, což:
  - snižuje zpoždění signálu,
  - umožňuje realizaci **rychlých obvodů** – např. sčítáček a násobiček (*klíčové pro ALU, DSP jednotky atd.*).

### Dodatečné bloky ve struktuře FPGA

- Rychlé **synchronní statické paměti (block RAM)**.
- **Hardwarové násobičky** pro DSP aplikace.
- **PLL / DLL (Phase Locked Loop / Delay Locked Loop)** – pro přesné řízení hodinových signálů.
- **Využití externí EEPROM** – pro uchování konfiguračních dat (*vlastní FPGA je typicky konfigurováno až po zapnutí*).



# 3. PLD & NPLD

## Architektura FPGA – Klíčové bloky

### IOB – Input/Output Block

- Řídí tok dat mezi **fyzickými vstupně-výstupními piny** a **vnitřní logikou FPGA**.
- **Obsahuje:**

- registry (*flip-flop*),
- budiče,
- multiplexory,
- ochranné obvody proti přepětí,
- volitelně zpožďovací obvody a paměťové členy.

### CLB – Configurable Logic Block

- Základní programovatelná jednotka logiky.

- **Obsahuje:**

- **LUT** – Look-Up Table, často postavenou na principu malých pamětí typu SRAM, ve kterých je zapsána logická funkce,
- **paměťové členy** (např. *D-klopné obvody*),
- **logiku pro řetězení** (*cascading*) sousedních CLB – užitečné pro tvorbu rozsáhlejších funkcí.

### Block RAM

- **Vysokorychlostní vnitřní paměť** integrovaná přímo v čipu.
- Použitelná např. jako:
  - zásobníky (*stack*),
  - FIFO buffery,
  - cache.

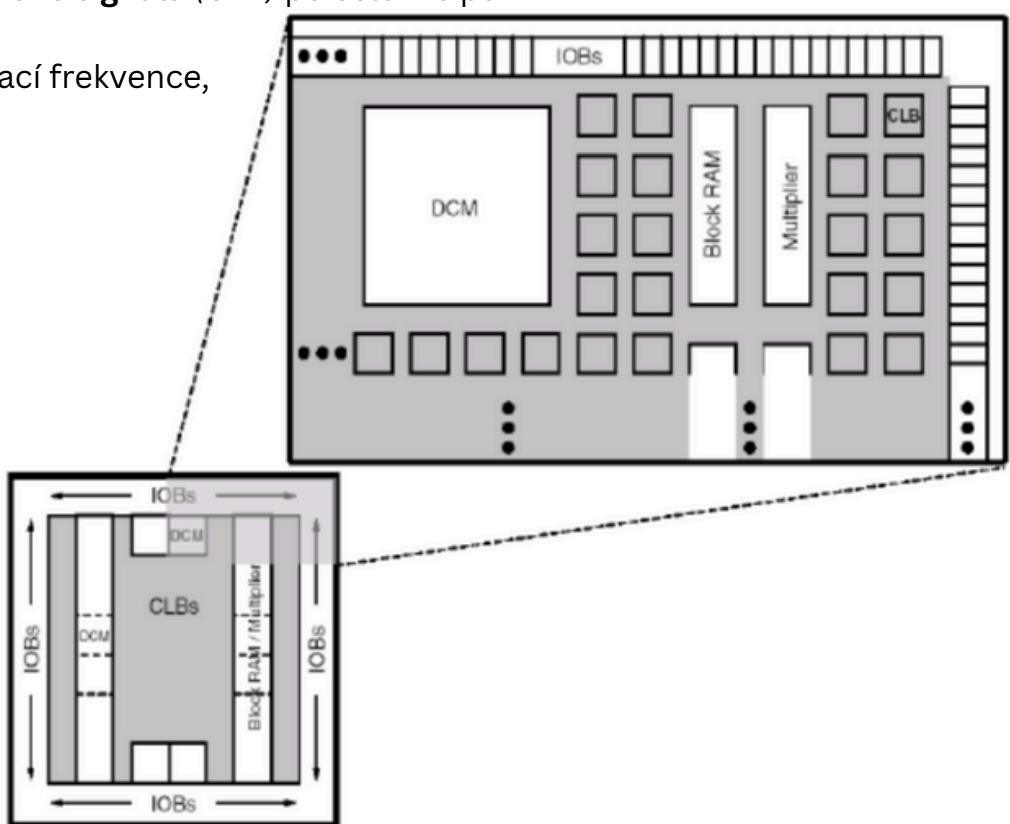
### Multiplier

- **Vestavěná násobička**, důležitá pro digitální signálové zpracování (DSP) nebo výpočty v reálném čase.

### DCM – Digital Clock Manager

- **Řízení a distribuce hodinového signálu (CLK)** po celém čipu.
- **Umožňuje:**

- generovat různé taktovací frekvence,
- potlačit rušení,
- synchronizovat signály.



# 3. PLD & NPLD

## Shrnutí:

**Vlastnost -** FPGA

**Logická složitost -** Velmi vysoká (až miliony hradel)

**Propojování -** Globální maticí + přímé propojení sousedních LB

**Paměť -** Interní RAM, LUT v CLB, externí EEPROM

**Flexibilita -** Maximální – vhodné i pro softprocesory, DSP atd.

**Příklad -** využití Vývoj CPU, DSP, řadičů, šifrovacích jednotek

## Makrobuňka vs. Logický blok

**Programovatelné logické obvody (PLD)** obsahují dvě základní struktury: makrobuňky a logické bloky. Obě slouží pro realizaci logických funkcí, ale liší se svou architekturou, použitím a výkonem.

### Makrobuňka (OLMC – Output Logic Macro Cell)

Makrobuňka je **základní konfigurovatelná jednotka** v SPLD a některých **CPLD obvodech**.

#### Charakteristika:

- **Zkratka OLMC** = Output Logic Macro Cell.
- Umožňuje **tvorbu jednoduchých logických operací (SLO)** – typicky kombinace kombinatoriky a sekvenční logiky.
- **Obsahuje:**
  - **kombinační logiku (AND, OR hradla),**
  - **D-klopný obvod (nejčastější typ flip-flopu),**
  - **pomocná hradla (např. pro výběr mezi přímým nebo zpožděným výstupem).**
- Každá buňka může reprezentovat malou **booleovskou funkci**.
- V jedné **makrobuňce lze vytvořit logickou funkci** až se **16 logickými součinami (AND termi)**.

#### Struktura makrobuňky:

- **Kombinační část:** zajišťuje výpočet **booleovské rovnice** (např. pomocí matic AND-OR).
- **Sekvenční část:** D-klopný obvod **uchovává výsledek a umožňuje jeho synchronizaci s hodinovým signálem.**
- Výstupní logika: může volit mezi přímým a registrovaným výstupem.

Více makrobuněk může být spojeno a **vytvořit tzv. logický blok**.

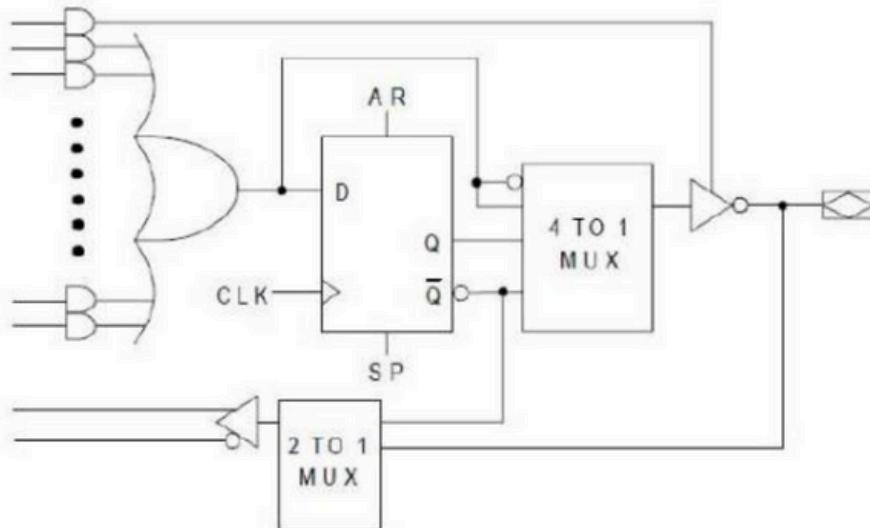
### Logický blok (Logic Block / Configurable Logic Block – CLB)

Logický blok je **základní stavební prvek FPGA**. Má větší flexibilitu a výpočetní sílu než makrobuňka.

#### Charakteristika:

- Může být stejně jednoduchý jako makrobuňka, ale **často je mnohem komplexnější**.
- **Obsahuje:**
  - **LUT (Look-Up Table)** – tabulka, která implementuje logické funkce pomocí RAM principu,
  - **flip-flop nebo registry** – pro ukládání stavů a synchronizaci,
  - **budiče, multiplexory a další pomocné obvody**.
- Lze je **propojit s jinými bloky** přes propojovací mřížku.
- **Logické bloky jsou rozloženy do matic (pole) → umožňují realizaci rozsáhlých logických struktur.**

### 3. PLD & NPLD



### ISE WebPACK – návrhové prostředí pro FPGA

ISE WebPACK je **bezplatná softwarová sada od firmy Xilinx**, určená pro **návrh, syntézu a programování FPGA a CPLD obvodů pomocí jazyků HDL (VHDL, Verilog)**.

#### Funkce a vlastnosti:

- Slouží k **překladu HDL kódu, syntéze, mapování na konkrétní PLD** a následné **konfiguraci FPGA**.
- **Vstupem je HDL kód** (např. *popis obvodu ve VHDL*), který je **převeden na konfigurační bitstream (.bit)**.
- Systém umožňuje spustit přímo poslední krok (*programování*), přičemž **automaticky provede všechny nutné předchozí kroky** (syntéza, mapování, implementace).

#### Hlavní uživatelské rozhraní – Project Navigator:

- **Obsahuje:**
  - stromovou hierarchii návrhu (*moduly, entity*),
  - editor HDL kódu,
  - konzoli s výstupy,
  - přístup ke všem nástrojům z levého panelu.
- Komplexní prostředí pro celý návrhový cyklus.

#### Programování FPGA:

- Probíhá pomocí aplikace **iMPACT**, která je součástí ISE.
- **Bitstream (.bit)** je **nahrán do FPGA přes USB nebo LPT programátor**.



# 3. PLD & NPLD

## Metodika návrhu PLD

**Návrh číslicového systému pomocí PLD** (např. *FPGA*) probíhá v několika jasně definovaných fázích:

### 1. Formální zápis

- Definování vstupů a výstupů systému.
- Základní specifikace problémového zadání.

### 2. Popis problému

#### • Pomocí:

- Funkčních a přechodových tabulek,
- Booleových rovnic,
- orientovaných grafů,
- časových průběhů,
- minimalizace (např. pomocí *Karnaughových map*),
- schématu zapojení.

### 3. Simulace

- Slouží k ověření návrhu ještě před implementací.
- Pomáhá odhalit chyby v logice a funkčnosti návrhu.

### 4. Realizace

- Implementace návrhu do konkrétního programovatelného obvodu.

## PLD kompilátor

### Co dělá PLD kompilátor:

- Nezávislý na konkrétním typu PLD – návrh lze vytvořit **bez znalosti cílového čipu**.
- Kompilátor **převádí popis logických funkcí** do formátu, který je možné nahrát do fyzického PLD.

### Výstupní soubory:

- **Dříve:** .JEDEC (.jed) – pro CPLD.
- **Dnes:** .BIT – bitstream pro FPGA.

### Transformace zápisu + optimalizace:

#### • Provádí:

- **minimalizaci logických rovnic** (např. zjednodušení pomocí algebraických pravidel),
- **optimalizaci vstupů/výstupů a využití makrobuněk** – podle možností cílového čipu.

# 3. PLD & NPLD

## NPLD – Neprogramovatelné logické obvody

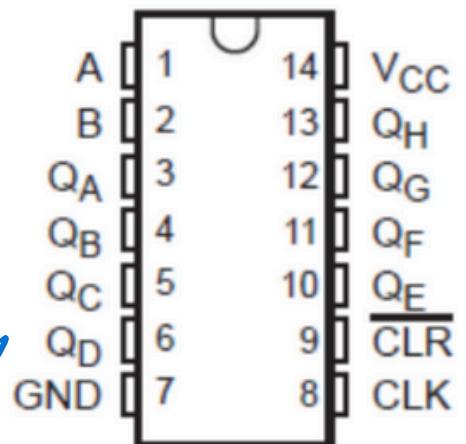
### Charakteristika:

- Funkce je pevně daná při výrobě – není možné ji uživatelsky měnit (na rozdíl od PLD).
- Jedná se o **hardware řešení** (pevná logická struktura).
- Typicky ve formě klasických **integrovaných obvodů řady 74xx**.

### Co obvykle obsahují:

- Jednoduché až středně složité logické struktury:
  - posuvné registry
  - záchytné registry
  - komparátory
  - čítače
  - posilovače sběrnic (*buffer*)
- Tyto komponenty se používají ve **složitějších číslicových systémech** jako stavební bloky.

### 74164 – 8bitový posuvný registr



### Základní informace:

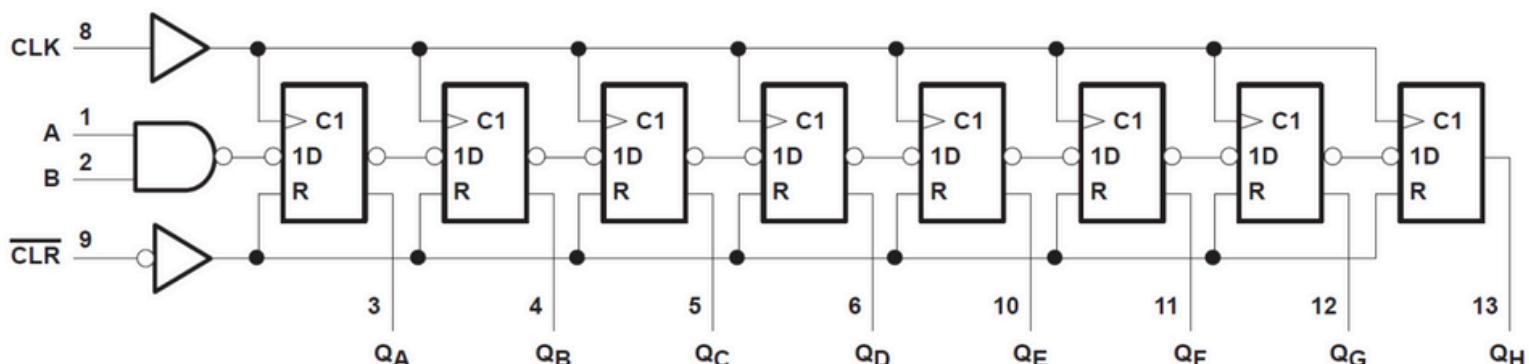
- **Typ:** 8bitový sériově–paralelní posuvný registr
- **Funkce:** Převod sériového vstupu na paralelní výstup
- **Využití:** Vysílací jednotky, rozšíření výstupů mikroprocesoru, řízení LED, atd.

### Vnitřní struktura:

- Obsahuje **8 kaskádově zapojených D-klopových obvodů**.
- Každým pulsem na vstupu **CLK** (*clock*) se data **posunou o jednu pozici doprava**.
- Informace je **posunována v sérii**, ale **čtena paralelně** na výstupech.

### Typické použití:

- Převodník **sériového signálu na paralelní** (např. při komunikaci s periferiemi).
- **Rozšíření výstupů mikroprocesoru** – např. když má procesor málo výstupních pinů:
  - 1 datový vstup + **CLK** → až 8 řízených výstupů ( $Q_a$ – $Q_h$ ).
- Řízení více **LED, LCD segmentů** nebo relé pomocí jen několika pinů.



- **A: Datový vstup** - zavádění nové informace
- **B: Povolení vstupu** - Data jsou povolená pouze pokud je B = 0
- **CLK: Hodinový signál** - Každý puls posune data o jednu pozici
- **CLR: Reset** - Aktivní v logické 0 – vymaže registry
- **Qa – Qh: Paralelní výstupy** - 8bitový výstup odpovídající obsahu D-klopových obvodů

### 3. PLD & NPLD

#### 74166 – 8bitový paralelně-sériový posuvný registr

##### Charakteristika:

- Integrovaný obvod typu **posuvný registr**.
- Umožňuje načíst 8bitová data paralelně a poté je vystupovat sériově.
- Funguje jako **paralelně-sériový převodník**.

##### Funkční popis:

###### Načtení dat (*LOAD* funkce):

- Pomocí řídicího vstupu SH/LD (*Shift/Load*):
  - **SH/LD = 0** → aktivní režim **LOAD** → načtení 8bitové hodnoty z paralelních vstupů **A–H** do registru.
  - Načtení dat proběhne s prvním náběžnou hranou **CLK**.

###### Posun dat (*SHIFT* funkce):

- **SH/LD = 1** → aktivní režim **SHIFT**.
- Při každém pulsu hodin (*CLK*) se data vregistrech **posunou o jednu pozici doprava**, poslední bit se objeví **na sériovém výstupu**.

###### Proces převodu:

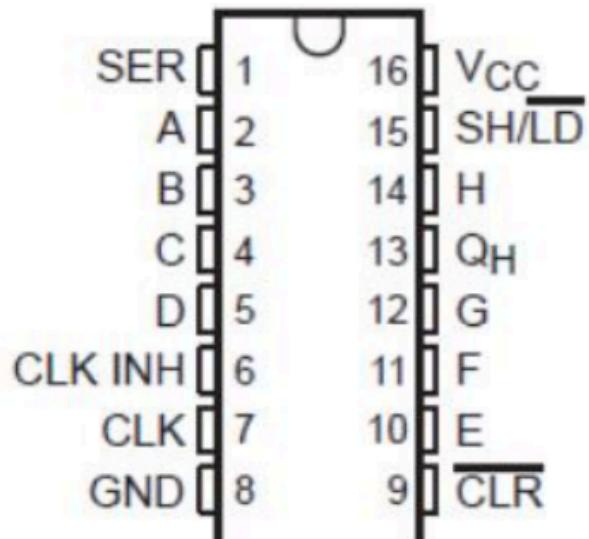
- **1x CLK** pro **načtení** dat (*SH/LD = 0*).
- **8x CLK** pro **výstup všech 8 bitů v sérii** (*SH/LD = 1*).
- Celkem tedy **9 hodinových pulsů pro úplný převod**.

###### Pozastavení hodin (*CLK INH – Clock Inhibit*):

- Slouží k **pozastavení posuvu** bez ovlivnění dat.
- **CLK INH = 1** → hodinové impulsy jsou ignorovány.

###### Základní parametry:

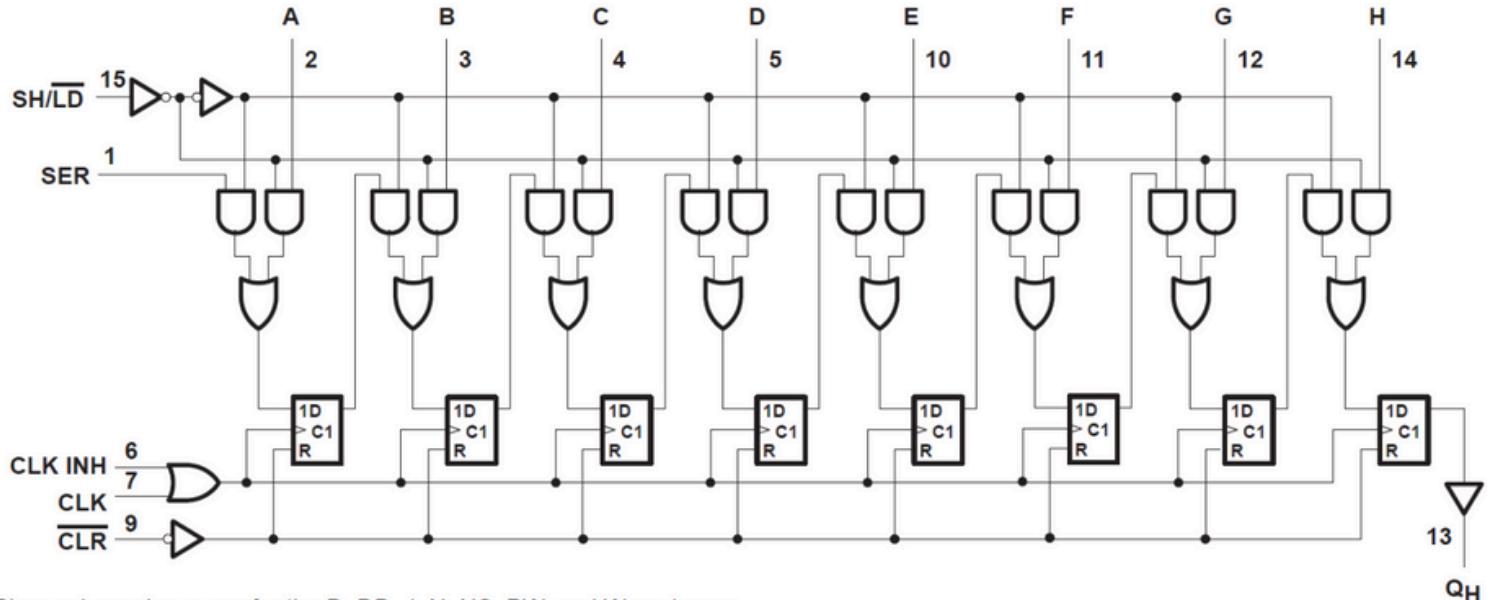
- **Bitová šířka:** 8 bitů
- **Vstup:** 8 paralelních vstupů (*A–H*)
- **Výstup:** 1 sériový výstup
- **Vnitřní struktura:** 8 D-klopných obvodů kaskádově propojených
- **Načítání a posuv:** Řízeno pomocí SH/LD a CLK signálů
- **Zastavení hodin:** Pomocí signálu CLK INH (*aktivní v log. 1*)



###### Využití v praxi:

- **Převod paralelních dat na sériový tok** (např. komunikace mezi obvody, sériové sběrnice).
- **Přijímací jednotky** (např. načtení bajtu a postupné dekódování).
- **Rozšíření vstupů mikroprocesoru** – mikroprocesor načítá data postupně přes jeden pin.

# 3. PLD & NPLD



Pin numbers shown are for the D, DB, J, N, NS, PW, and W packages.

- A–H: Paralelní vstupy** - Načítání 8bitových dat
- QH: Sériový výstup** - Výstup postupně posouvaných bitů
- CLK: Hodinový signál** - Řídí načtení a posun
- CLK INH: Clock Inhibit** - Pozastavení hodin (*aktivní v 1*)
- SH/LD: Shift/Load řízení** – 0 = načti, 1 = posouvej
- \CLR: Reset registru** - Vymaže obsah (*aktivní v log. 0*)

## 74595 – 8bitový sériově-paralelní posuvný registr s třístavovým výstupem

### Charakteristika:

- Jedná se o **8bitový posuvný registr**.
- Sériový vstup, paralelní výstup** (podobně jako 74164), ale navíc obsahuje:
  - třístavové výstupní budiče (*tristate*),
  - záchytný výstupní registr

### Základní parametry:

- Bitová šířka:** 8 bitů
- Vstup:** 1 sériový (DS)
- Výstup:** 8 paralelních (Q0–Q7)
- Vnitřní struktura:** Posuvný registr + výstupní registr + třístavové budiče
- Technologie:** CMOS / TTL kompatibilní
- Klíčová funkce navíc:** Třístavový výstup a záchytný registr

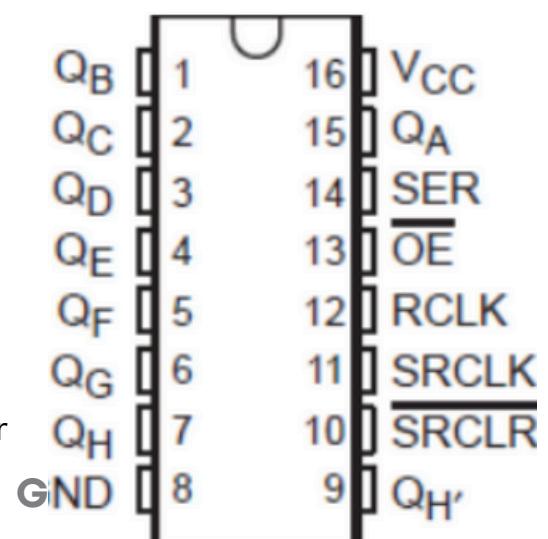
### Struktura a funkční bloky:

#### 1. Sériový posuvný registr (Shift Register):

- Uvnitř je **8 D-klopných obvodů** propojených kaskádově.
- SRCLK (Shift Register Clock)** posouvá data **doprava po každém pulsu**.

#### 2. Záchytný výstupní registr:

- Po dokončení posunu je možné **zachytit data do výstupního registru** pomocí signálu **RCLK (Register Clock)**.
- Tím se **odděluje posun dat od výstupu** – výstupy se mění jen při RCLK.



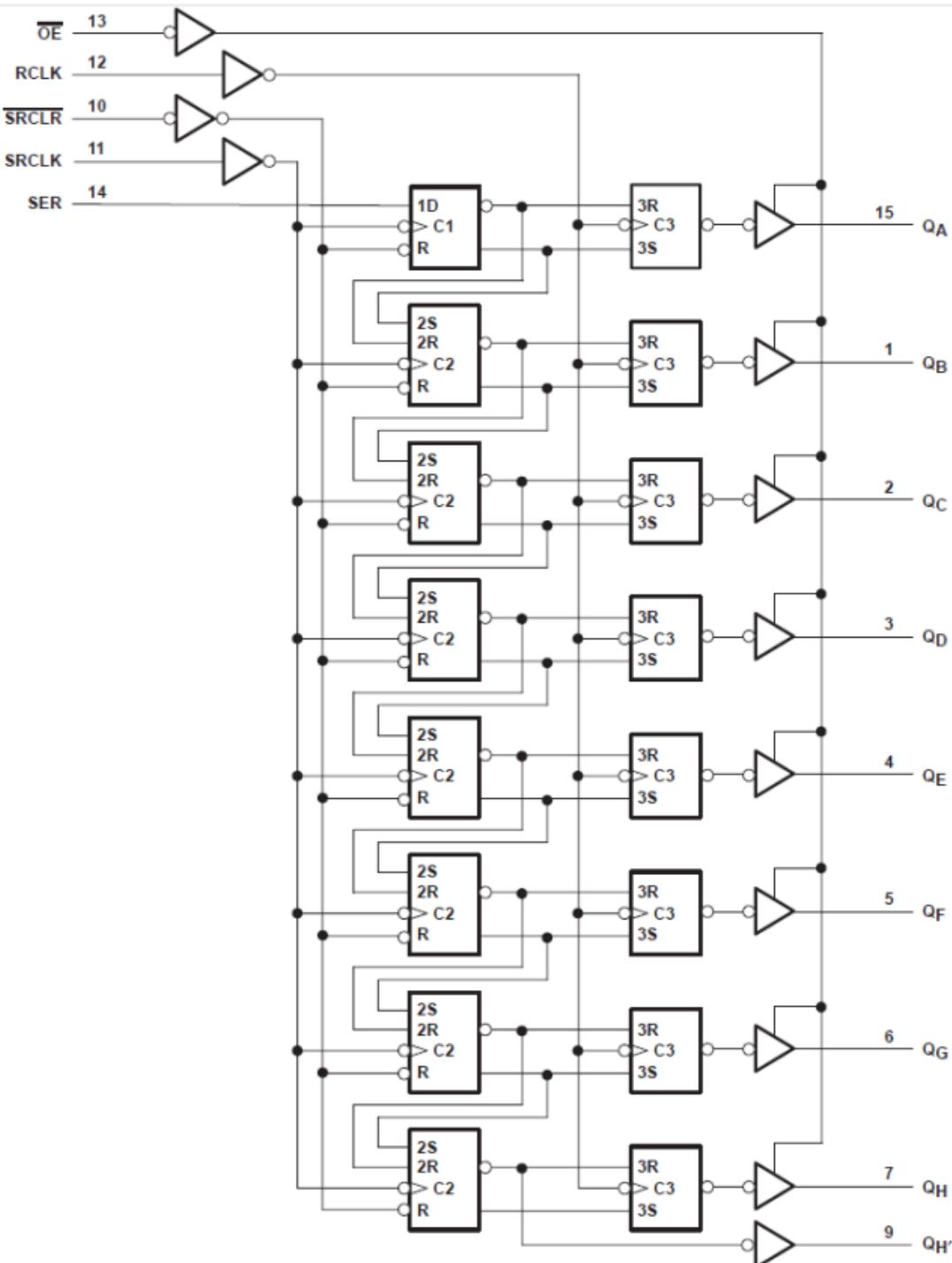
# 3. PLD & NPLD

## 3. Třístavové budiče (Tristate Drivers):

- Výstupy Q0–Q7 jsou připojeny přes zesilovače s třemi stavami:
  - 0 – logická nula,
  - 1 – logická jednička,
  - Z – vysoká impedance (výstup „odpojen“).
- Ovládány signálem OE (Output Enable):
  - OE = 0 → výstupy jsou aktivní,
  - OE = 1 → výstupy jsou ve stavu vysoké impedance.

## Využití v praxi:

- Rozšíření výstupů mikroprocesoru – např. 3 piny (DS, SRCLK, RCLK) obslouží 8 výstupů.
- Ovládání LED, sedmsegmentovek, relé.
- Světelné a textové tabule (např. v dopravě, reklamě).
- Více registrů lze řetězit za sebe → umožňuje vytvořit třeba 16, 24 nebo 32bitový výstup.



- **DS:** Sériový datový vstup (Data Serial)
- **SRCLK:** Hodinový signál pro posuvný registr
- **RCLK:** Hodinový signál pro záhytný registr
- **OE:** Výstupní povolení (aktivní v 0) – řídí tristate
- **\SRCLR:** Asynchronní reset posuvného registru (aktivní v 0)
- **QA–QH:** Paralelní výstupy (výstupní data z výstupního registru)
- **QH':** Sériový výstup pro řetězení s dalším registrum (např. další 74595)
- **VCC:** Napájecí napětí (typicky +5 V)
- **GND:** Zem (společný potenciál)

# 3. PLD & NPLD

## 74573 – 8bitový záhytný registr (Latch Register)

### Charakteristika:

- Obvod 74573 je **8bitový záhytný registr typu LATCH**, známý také jako **střadač**.
- Používá se pro **dočasné uchování dat přivedených na vstupy a jejich stabilní výstup**, dokud nejsou přepsána.
- Zajišťuje **rychlé a synchronní zachycení dat** z datové sběrnice.

### Základní vlastnosti:

- Počet vstupů/výstupů:** 8 vstupů D, 8 výstupů Q (vždy v páru)
- Rídicí signály:** LE (Latch Enable), \OE (Output Enable)
- Výstupy:** Třístavové (0, 1, Z)
- Typ logiky:** D-latch – záhytné klopné obvody

### Funkční popis:

#### Latch Enable – LE (aktivní v 1)

- Pokud je LE = 1, data z D vstupů se propisují na výstupy Q.
- Je aktivní průběžné propisování – výstupy sledují vstupy.
- Jakmile LE = 0, aktuální stav dat se zachytí a na výstupech zůstane zachován – tzv. latched.

#### Output Enable – \OE (aktivní v 0)

- Řídí aktivitu výstupů **Q0–Q7**:
  - \OE = 0 → výstupy jsou **aktivní (0 nebo 1)**.
  - \OE = 1 → výstupy jsou ve stavu **vysoké impedance (Z)** – "odpojeny" od sběrnice.
- Poznámka:** Tento signál neovlivňuje uložená data – jen řídí výstup

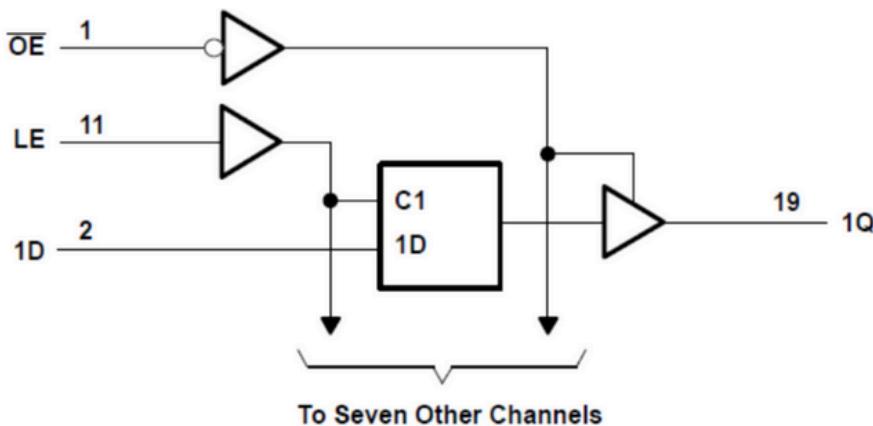
\OE	1	20	V <sub>CC</sub>
1Q	2	19	8Q
1D	3	18	8D
2D	4	17	7D
2Q	5	16	7Q
3Q	6	15	6Q
3D	7	14	6D
4D	8	13	5D
4Q	9	12	5Q
GND	10	11	LE

### Struktura obvodu:

- Obsahuje **8 D-latchů** (D-klopné obvody bez hodin), propojených s odpovídajícími výstupními piny.
- Každý **latch** má svůj vstup **D** a výstup **Q**.
- Výstupy jsou vybaveny **tristate** budíci řízenými signálem \OE.

### Využití v praxi:

- Zachytávání dat z datové sběrnice** – např. při komunikaci mezi mikroprocesorem a pamětí nebo periferií.
- Rozšíření vnější paměti mikroprocesoru** – jako dočasná vyrovnávací paměť.
- Synchronizace signálů mezi dvěma subsystémy s odlišným časováním.
- Vhodné tam, kde je potřeba krátkodobě podržet data a později je bezpečně odeslat dál.



**1D: Datový vstup** – Přivádí logickou hodnotu do záhytného registru

**LE: Latch Enable** – Uložení dat z D na výstup Q (aktivní v 1)

**\OE: Output Enable** – Aktivace výstupů (aktivní v 0), jinak vysoká impedance

**1Q: Výstup** – Uchovaná hodnota z registru dostupná na výstupu

**C1 / 1D (blok):** D-latch – Uchovává vstupní hodnotu při aktivním LE

**To Seven Other Channels:** Zbylých 7 bitů registru (toto schéma ukazuje jen 1/8 části)

# 3. PLD & NPLD

## 74244 – Posilovač sběrnice / Jednosměrný budič

### Charakteristika:

- Obvod 74244 je **čtyřbitový jednosměrný budič** (*bus driver*).
- Slouží jako **posilovač signálu** – zajišťuje **přenos dat mezi sběrnicemi** nebo mezi **CPU a periferiemi**.
- Umožňuje **řídit větší proudové zatížení** na výstupu než je možné na běžném výstupu logického obvodu.

### Základní vlastnosti:

- **Počet vstupů/výstupů:** 4 vstupy, 4 výstupy
- **Typ přenosu:** Jednosměrný (*vstup* → *výstup*)
- **Napěťová úroveň vstupu:** Tří stavový (*0, 1, Z*) – možnost odpojení
- **Řízení:** pomocí signálu OE (*Output Enable*)
- **Využití:** Zesílení signálu, adresové sběrnice, připojení výkonnějších prvků

### Funkce signálu OE (*Output Enable*):

- Řídí aktivitu výstupu:
  - **OE = 0** → výstupy jsou aktivní (*propouští hodnotu vstupu*).
  - **OE = 1** → výstupy jsou ve vysoké impedanci (*Z*) – „**odpojeny**“ od sběrnice.
- Díky tomu lze **více zařízení sdílet stejnou sběrnici**, aniž by došlo ke kolizi.

### Vlastnosti jako budič:

- Výstupy obvodu mohou dodávat **vyšší proud než vstupy běžných logických obvodů**.
- Chová se jako **proudový zesilovač, přizpůsobený pro řízení větší zátěže**.

### Typické použití:

- **Adresové sběrnice** (např. mezi CPU a pamětí).
- **Datové sběrnice** – přepínání mezi více zařízeními.
- Izolace a řízení signálů do zařízení s vyšším odběrem (např. LED pole, relé).
- **Ochrana citlivých vstupů procesoru** – přes budič lze řídit výstup s větším proudem.

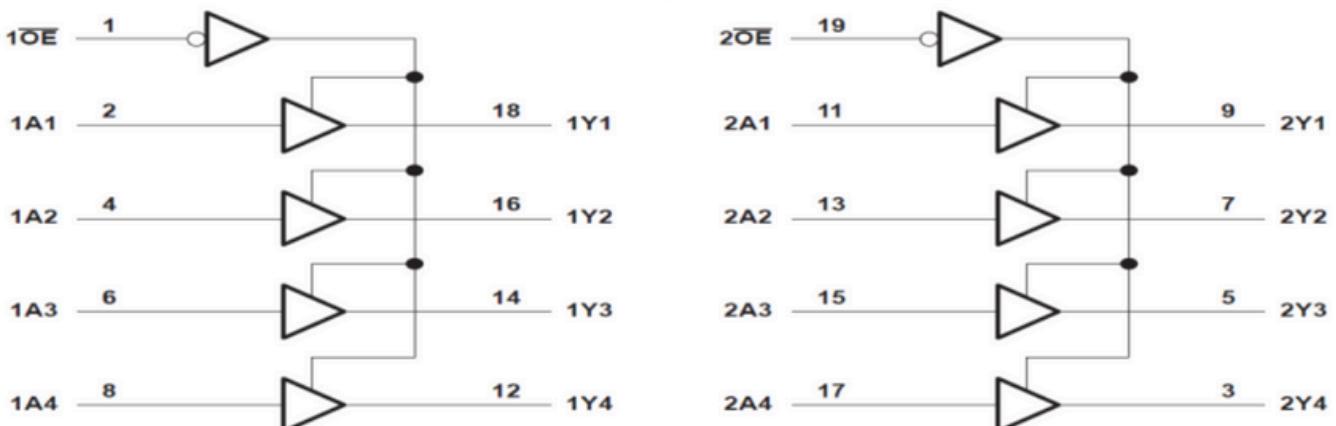
### Shrnutí signálů (v obecném stylu):

- **A0–A3: Vstupy** - Přijímají logickou hodnotu (*0 nebo 1*)
- **Y0–Y3: Výstupy** - Replikují vstup, pokud je OE aktivní
- **OE: Output Enable** - Řídí výstup: 0 = aktivní, 1 = vysoká impedance
- **VCC: Napájení** - +5 V
- **GND: Zem** - 0 V

FUNCTION TABLE  
(EACH BUFFER)

INPUTS		OUTPUT Y
OE	A	
L	H	H
L	L	L
H	X	Z

LOGIC DIAGRAM (POSITIVE LOGIC)



# 3. PLD & NPLD

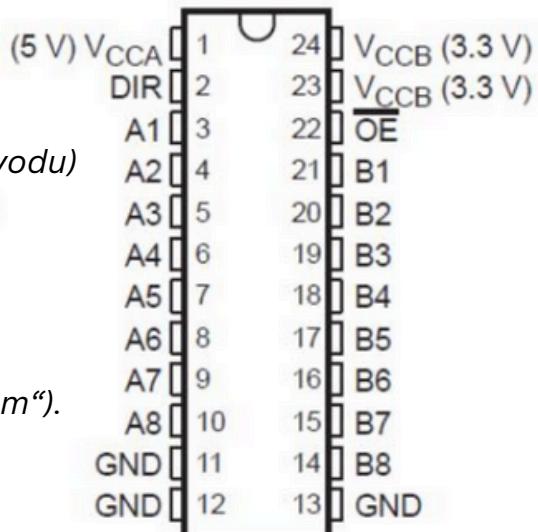
## 74245 – Oboustranný budič sběrnice (Bus Transceiver)

### Charakteristika:

- Obvod 74245 je **osmibitový oboustranný budič sběrnice s třístavovými výstupy**.
- Umožňuje **obousměrný přenos dat mezi dvěma částmi systému**, typicky mezi **sběrnicí a procesorem** nebo mezi dvěma sběrnicemi.
- Funguje i jako **napěťový převodník** – např. z **3,3 V na 5 V** a naopak (*dle varianty*).

### Základní vlastnosti:

- Počet vstupů/výstupů:** 8 vstupů a 8 výstupů (A0–A7 ↔ B0–B7)
- Přenos dat:** Obousměrný (*směr se volí pomocí DIR*)
- Výstupy:** Třístavové – možnost odpojení (*Z stav*)
- Rídící signály:** DIR (Direction), \OE (Output Enable)
- Převod úrovní:** Napěťový převod **3,3 V ↔ 5 V** (*varianty obvodu*)
- Typ logiky:** CMOS / TTL kompatibilní



### Směr přenosu dat (DIR):

- DIR = 1** → přenos z portu A do portu B
- DIR = 0** → přenos z portu B do portu A

Tedy vždy: signál teče ve směru DIR (→ *směr určuje „odkud kam“*).

### Výstupní povolení (\OE):

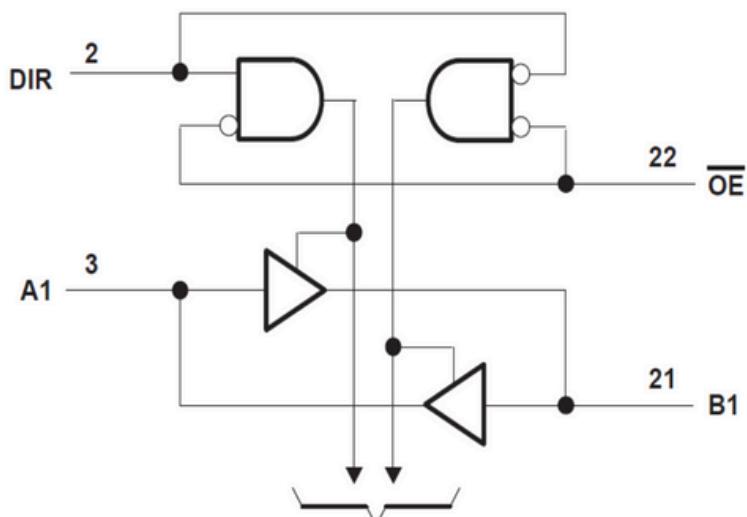
- Řídí aktivitu výstupů na obou stranách:
  - \OE = 0** → výstupy jsou **aktivní** (*normální funkce*).
  - \OE = 1** → výstupy jsou ve **vysoké impedanci** (*Z*) – „odpojeny“ od sběrnice.
- Umožňuje více **zařízení sdílet stejnou sběrnicí, bez kolizí**.

### Použití v praxi:

- Obousměrná komunikace** mezi dvěma datovými sběrnicemi.
- Spojování zařízení s různou logickou úrovní (*např. převod mezi 3,3 V a 5 V*).
- Izolace sběrnic** – zabránění nechtěnému průniku signálu.
- Zesílení proudové kapacity výstupů** – podobně jako 74244, ale obousměrně.

### Shrnutí signálů (v obecném stylu):

- A0–A7: Port A** – Jeden ze sběrnicových kanálů (směr podle DIR)
- B0–B7: Port B** – Druhý sběrnicový kanál
- DIR: Direction** – Určuje směr dat: 1 = A→B, 0 = B→A
- \OE: Output Enable** – Aktivní v 0, povoluje výstupy; 1 = vysoká impedance
- VCC: Napájení** – Typicky 5 V
- GND: Zem** – Společný potenciál



# 3. PLD & NPLD

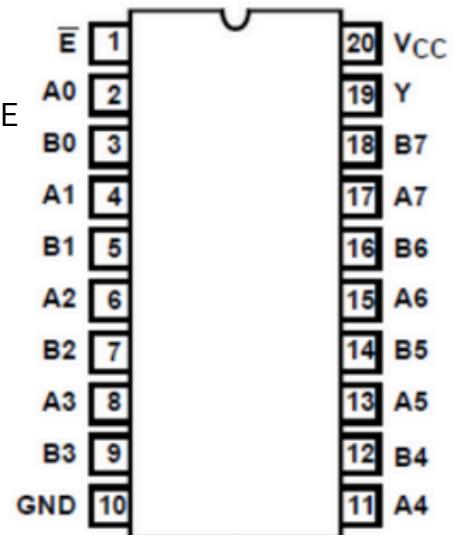
## 74688 – 8bitový komparátor

### Charakteristika:

- Obvod 74688 je **8bitový číslicový komparátor**.
- Slouží k **porovnání dvou 8bitových binárních hodnot (slov)** – vstup A a vstup B.
- Výstupem je signál, který informuje o **shodě nebo neshodě vstupních hodnot**.

### Základní vlastnosti:

- **Bitová šířka:** 8 bitů
- **Vstupy:** 8bitový vstup A a 8bitový vstup B
- **Výstup:** Třístavový (0, 1, Z) – závisí na výsledku porovnání a OE
- **Typ výstupu:** Aktivně nízký (*shoda = 0*)
- **Vnitřní struktura:** Hradla XNOR + AND + NAND
- **Řídicí signál:** OE (Output Enable) – výstupní povolení



### Směr přenosu dat (DIR):

- **DIR = 1** → přenos z portu A do portu B
- **DIR = 0** → přenos z portu B do portu A

### Princip funkce:

- Každý bit vstupů A a B je **porovnán pomocí XNOR hradla**:
  - XNOR vrací **1**, pokud jsou **bity stejné** ( $A_i = B_i$ ).
- Výsledky všech **8 XNOR hradel jsou spojeny přes AND hradla**:
  - Pokud všechny bity **souhlasí**, výstup z **AND = 1**.
- Výsledek je **negován pomocí NAND hradla**:
  - Pokud **všechny bity souhlasí**, výstup **komparátoru je 0** (aktivně nízký výstup).
  - Pokud se liší **jediný bit**, výstup = **1**.

Výstup je tedy **logická 0 pouze tehdy, když A = B**.

### Výstupní povolení – OE:

- **OE = 0** → výstup je aktivní.
- **OE = 1** → výstup je ve vysoké impedanci (Z) – "odpojen" od sběrnice.

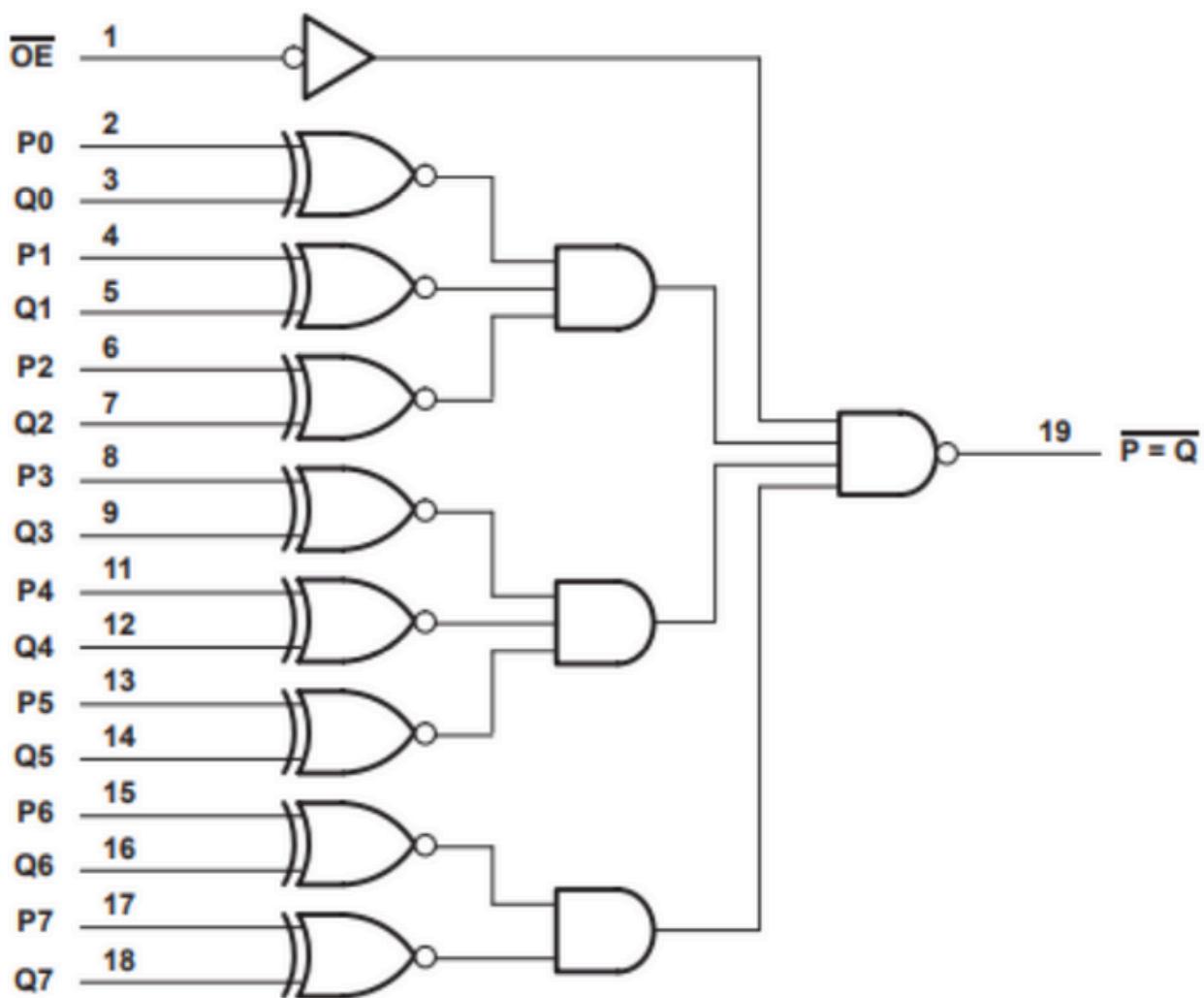
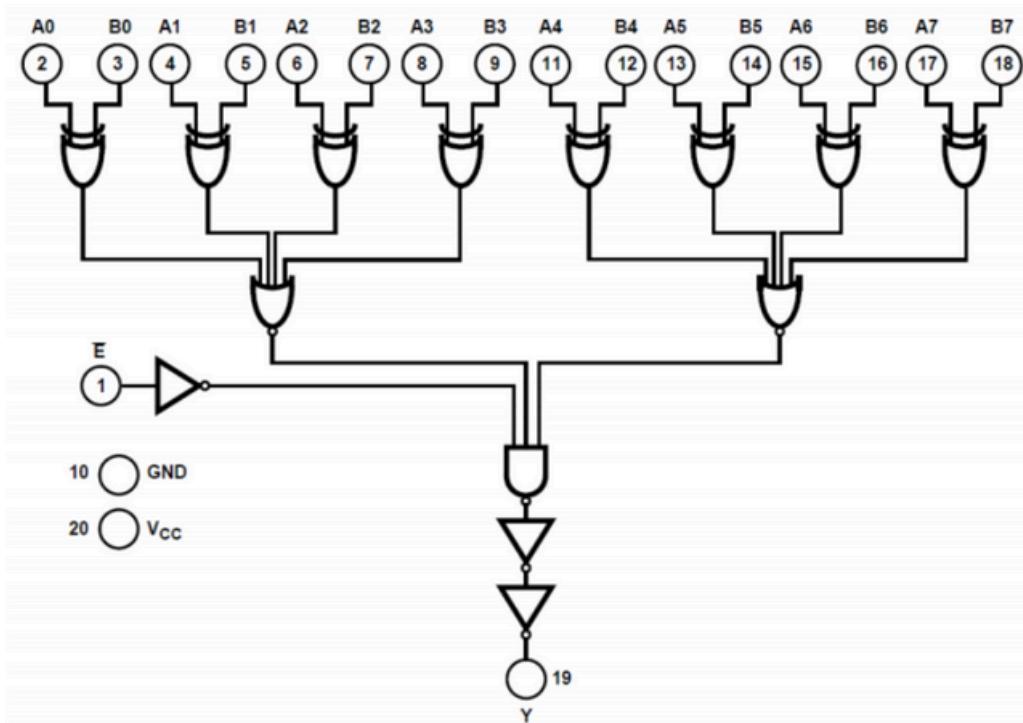
### Typické použití:

- **Porovnání dvou 8bitových slov** – například kontrola rovnosti registrů nebo dat.
- **Adresní dekodér** – jedna strana vstupu je pevně nastavená, druhá přichází z adresové sběrnice:
  - Když se adresa rovná očekávané, výstup komparátoru se aktivuje.
- **Řídicí logika, stavové automaty, přístupové systémy**, dekódování specifických kódů.

### Shrnutí signálů:

- **A0–A7:** Vstupní slovo A (např. pevná hodnota)
- **B0–B7:** Vstupní slovo B (např. dynamická hodnota ze sběrnice)
- **OE:** Output Enable – Řídí výstup: 0 = aktivní, 1 = vysoká impedance
- **Q (výstup):** Výsledek porovnání: 0 = A = B, 1 = A ≠ B

### 3. PLD & NPLD



# 3. PLD & NPLD

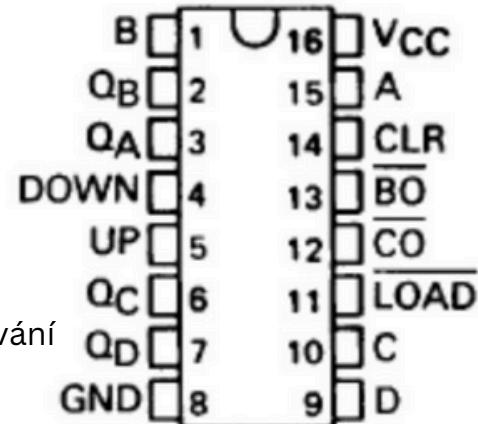
## 74193 – 4bitový obousměrný čítač (Binary Up/Down Counter)

### Charakteristika:

- 74193 je **4bitový binární čítač**, který umožňuje **čítání nahoru (inkrementaci)** i **dolů (decrementaci)**.
- **Je synchronní** – všechny změny probíhají současně s hranou hodinového signálu.
- **Obsahuje funkci předvolby (preset)** – umožňuje čítač nastavit na libovolnou hodnotu.
- Umožňuje **kaskádové zapojení** pro více než 4 bity.

### Základní vlastnosti:

- **Počet bitů:** 4 bity (*počítá od 0 do 15*)
- **Typ čítání:** 8bitový vstup A a 8bitový vstup B
- **Klopné obvody:** Vnitřně složen z 4 JK-klopých obvodů
- **Řízení směru:** Pomocí dvou oddělených vstupů hodin (*CLK*)
- **Přednastavení:** Možné pomocí vstupů P0–P3 a signálu LOAD
- **Výstupy:** Q0–Q3 – binární výstupní hodnota čítače
- **Indikátory přetečení:** Borrow (*BO*) a Carry (*CO*) – pro kaskádování



### Funkční bloky a signály:

- **A–D: Paralelní vstupy** – Načítání 4bitové hodnoty při aktivním signálu LOAD
- **QA–QD: Paralelní výstupy** – Aktuální 4bitová hodnota čítače
- **UP: Vstup pro čítání nahoru** – Při náběžné hraně inkrementuje čítač
- **DOWN: Vstup pro čítání dolů** – Při náběžné hraně dekrementuje čítač
- **LOAD: Paralelní načtení** – Při 0 se načtou hodnoty z A–D do čítače
- **CLR: Asynchronní reset** – Vymaže obsah čítače (aktivní v log. 0)
- **CO: Carry output** – Indikace přetečení při čítání nahoru (pro kaskádu)
- **BO: Borrow output** – Indikace podtečení při čítání dolů (pro kaskádu)
- **GND: Zem** – Společný potenciál
- **VCC: Napájení** – Typicky +5 V

### Popis funkce:

#### 1. Čítání nahoru:

- Aktivace CLK UP způsobí inkrementaci (*např. 0111 → 1000*).
- CO = 1, pokud čítač přeteče z 1111 na 0000 (*pro spojení dalšího čítače*).

#### 2. Čítání dolů:

- Aktivace CLK DOWN způsobí dekrementaci (*např. 1000 → 0111*).
- BO = 1, pokud čítač „**podteče**“ z 0000 na 1111.

#### 3. Přednastavení hodnoty:

- Při LOAD = 0 se na výstupy Q0–Q3 okamžitě načtou hodnoty z P0–P3.

#### 4. Reset (CLR):

- Při CLR = 0 se čítač vynuluje (*0000*) bez ohledu na ostatní signály.

### Použití v praxi:

- **Událostní čítač** – například počítání výrobků na lince.
- **Odpocítávací časovač** – dekrementace k nule (časový limit).
- **Pořadník / sekvenční řízení** – např. výběr výstupního kanálu.
- **Adresování paměti** – čítač jako generátor adres.
- **Kaskádové čítání** – snadné rozšíření na 8, 12 nebo 16 bitů.

# 3. PLD & NPLD

## Hradla

### AND

Funkce	$Y = A \cdot B$			
Značení		Pravdivostní tabulka		
norma	symbol	$X_1(A)$	$X_2(B)$	$Y$
ANSI/MIL		0	0	0
IEC		0	1	0
DIN		1	0	0
		1	1	1

### NAND

Funkce	$Y = \overline{A \cdot B} = \overline{A} + \overline{B}$			
Značení		Pravdivostní tabulka		
norma	symbol	$X_1(A)$	$X_2(B)$	$Y$
ANSI/MIL		0	0	1
IEC		0	1	1
DIN		1	0	1
		1	1	0

### OR

Funkce	$Y = A + B$			
Značení		Pravdivostní tabulka		
norma	symbol	$X_1(A)$	$X_2(B)$	$Y$
ANSI/MIL		0	0	0
IEC		0	1	1
DIN		1	0	1
		1	1	1

### XOR

Funkce	$Y = A \oplus B = \overline{A} \cdot B + A \cdot \overline{B}$			
Značení		Pravdivostní tabulka		
norma	symbol	$X_1(A)$	$X_2(B)$	$Y$
ANSI/MIL		0	0	0
IEC		0	1	1
DIN		1	0	1
		1	1	0

### NOR

Funkce	$Y = \overline{A + B} = \overline{A} \cdot \overline{B}$			
Značení		Pravdivostní tabulka		
norma	symbol	$X_1(A)$	$X_2(B)$	$Y$
ANSI/MIL		0	0	1
IEC		0	1	0
DIN		1	0	0
		1	1	0

### XNOR

Funkce	$Y = \overline{A \oplus B} = A \cdot \overline{B} + \overline{A} \cdot B$			
Značení		Pravdivostní tabulka		
norma	symbol	$X_1(A)$	$X_2(B)$	$Y$
ANSI/MIL		0	0	1
IEC		0	1	0
DIN		1	0	0
		1	1	1