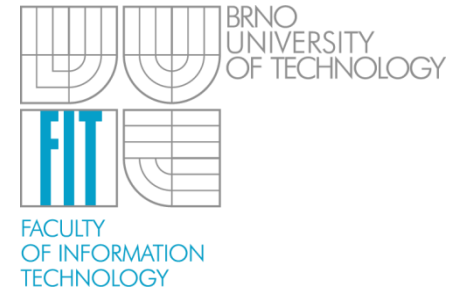


# Návrh číslicových systémů (INC)

Otto Fučík

Vysoké učení technické v Brně  
Fakulta informačních technologií  
Božetěchova 2, 612 66 Brno



## Použitá literatura

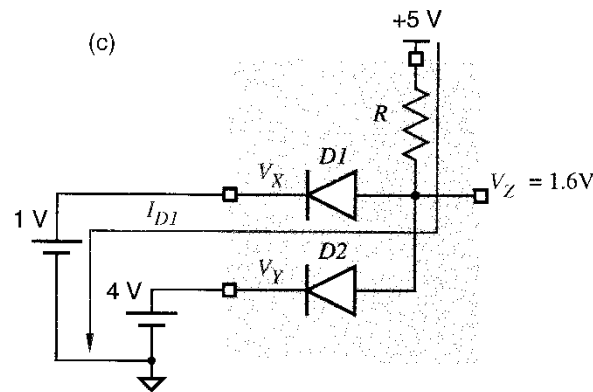
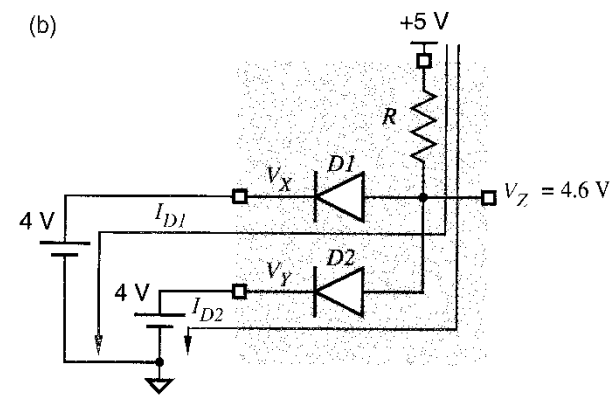
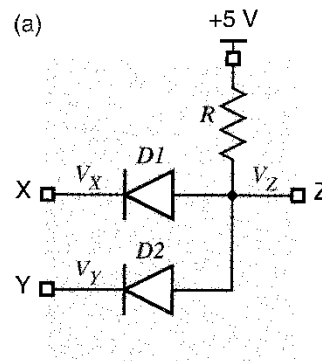
- N. Frištacký, M. Kolesár, J. Kolenička a J. Hlavatý: „Logické systémy“, SNTL Praha, 1986  
M. Eysselt: „Logické systémy“, SNTL Praha, skriptum VUT v Brně, 1985  
J. F. Wakerly: „Digital Design. Principles and Practices“, Prentice Hall, ISBN 0-13-769191-2, 2000  
V. P. Nelson, H.T.Nagle, B.D.Carroll, J.D.Irwin: „Digital Logic Circuit Analysis & Design“, ISBN 0-13-463894-8, 1995  
T.L.Floyd: „Digital Fundamentals“, Prentice Hall, ISBN 0-13-080850-4, 2000

Rodina	Popis	Zpoždění [ns]	Frekvence [MHz]	Příkon/hradlo [mW/MHz]	Napájecí napětí [V]	Rok	Poznámka
RTL	Resistor–transistor logic	500	4	10	3.3	1963	První CPU z IO (použito pro řízení Apollo)
DTL	Diode–transistor logic	25		10	5	1962	
CMOS	AC/ACT	3	125	0.5	3.3 nebo 5	1985	TTL kompatibilní úrovně
CMOS	HC/HCT	9	50	0.5	5	1982	TTL kompatibilní úrovně
CMOS	4000B/74C	30	5	1.2	10	1970	První CMOS
TTL		10	25	10	5	1964	Původní
TTL	L	33	3	1	5	1964	Low power
TTL	H	6	43	22	5	1964	High speed
TTL	S	3	100	19	5	1969	Schottky high speed
TTL	LS	10	40	2	5	1976	Low power Schottky high speed
TTL	ALS	4	50	1.3	5	1976	Advanced Low power Schottky
TTL	F	3.5	100	5.4	5	1979	Fast
TTL	AS	2	105	8	5	1980	Advanced Schottky
TTL	G	1.5	1125		1.65 - 3.6	2004	First GHz 7400 series logic
ECL	ECL III	1	500	60	-5.2	1968	Improved ECL
ECL	MECL I	8		31	-5.2	1962	První IO vyráběný komerčně
ECL	ECL 10K	2	125	25	-5.2	1971	Motorola
ECL	ECL 100K	0.75	350	40	-4.5	1981	
ECL	ECL 100KH	1	250	25	-5.2	1981	

Technologie	Příkon	Rychlost
Resistor-Transistor Logic (RTL)	***	*
Diode-Transistor Logic (DTL)	***	*
Transistor-Transistor Logic (TTL)	**	**
Emitter-Coupled Logic (ECL)	***	***
Positive Metal Oxide Semiconductor (pMOS)	**	*
Negative Metal Oxide Semiconductor (nMOS)	**	**
Complementary Metal Oxide Semiconductor (CMOS)	*	**
Gallium Arsenide (GaAs)	***	***

- **Logické úrovně**
  - Log. členy jsou konstruovány tak, aby byly za normálních podmínek (teplota, napájecí napětí, rušení, atd.) schopny generovat log. úrovně v jistém intervalu hodnot a též rozlišovat log. úrovně v určitém rozmezí hodnot
- **Odolnost proti rušení (DC Noise Margins)**
  - Je zajištěna v určitém rozmezí tak, že log. člen je schopen akceptovat větší rozptyl vstupních hodnot log. úrovní, než jaký generuje na výstupu
  - Rušení může být generováno např. kosmickým zářením, elektromagnetickým polem, kolísáním napájecího napětí apod.
- **Logický zisk (Fan-Out)**
  - Počet vstupů log. členů, které můžeme zapojit na výstup daného členu, při kterém jsou ještě zaručeny správné hodnoty log. úrovní pro celý rozsah pracovních podmínek (napájecí napětí, teplota)
- **Rychlost**
  - Doba, které je třeba k přechodu signálu ze vstupu na výstup
  - Dána dobami přechodů mezi log. úrovněmi a dobou průchodu signálu
  - Závisí na konstrukci log. členů, na počtu jiných log. členů zapojených na jeho výstup, na délce vodičů, na konstrukci desky s plošnými spoji atd.
- **Příkon**
  - Závisí na konstrukci log. členu, na počtu členů, frekvenci změn log. úrovní, parazitních atd.

- Realizace hradla AND (a)
- Ilustrace činnosti (b)
  - Oba vstupy jsou na H
  - Výstup je též na úrovni H
- Ilustrace činnosti (c)
  - Jeden vstup je na L
  - Druhý vstup je na H
  - Výstup je na úrovni L
- Pravdivostní tabulka (d)
  - Pomocí hodnot L a H
- Pravdivostní tabulka (e)
  - Pomocí hodnot 0 a 1



(d)

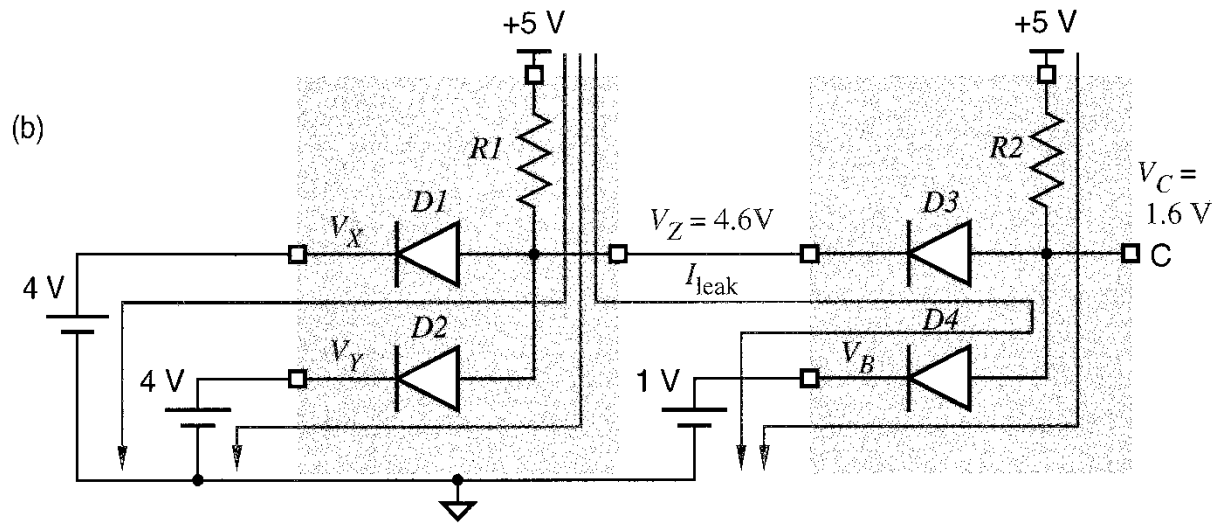
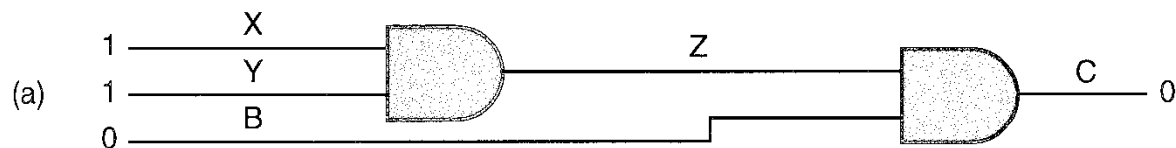
$V_X$	$V_Y$	$V_Z$
low	low	low
low	high	low
high	low	low
high	high	high

(e)

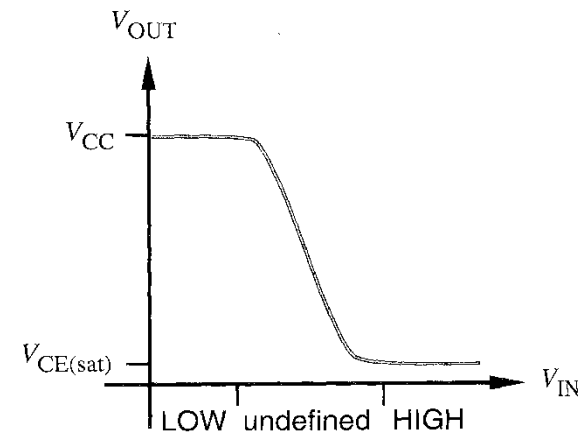
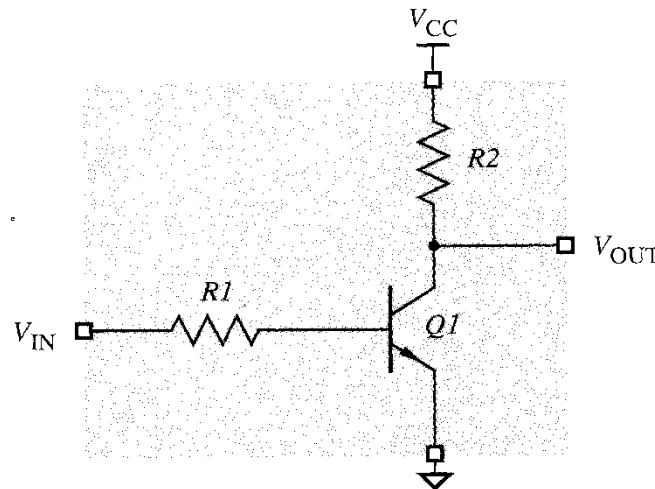
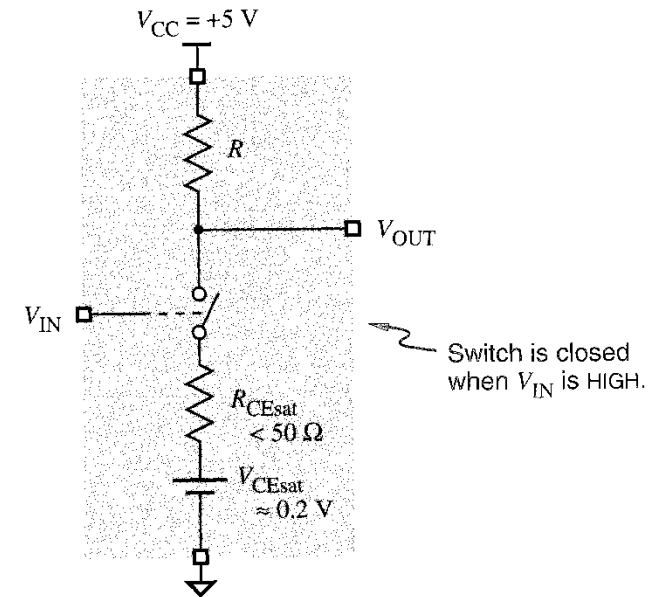
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

- Tabulka log. úrovní pro napájení 5 V
- Příklad realizace složitějších funkcí pomocí diodových log. členů
  - Log. schéma (a)
  - Podrobné schéma (b)
- Poznámka
  - Díky úbytkům napětí na diodě roste hodnota log. 0 => nelze zapojit více členů za sebe

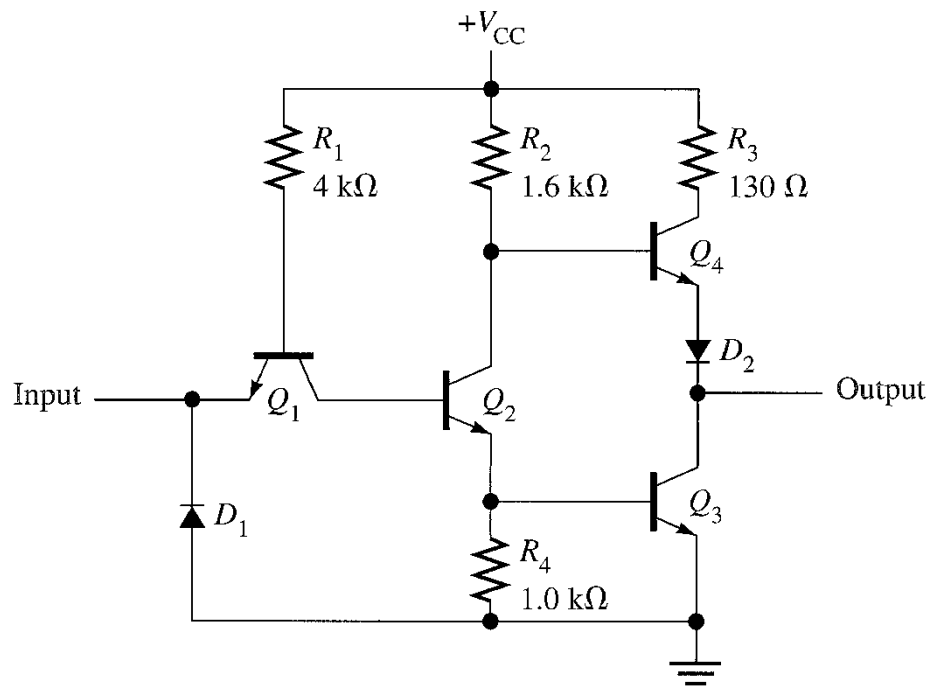
<i>Signal Level</i>	<i>Designation</i>	<i>Binary Logic Value</i>
0–2 volts	LOW	0
2–3 volts	noise margin	undefined
3–5 volts	HIGH	1



- Náhradní schéma tranzistoru jako spínače
  - $R$  - přídavný rezistor (pull-up)
  - $R_{CEsat}$  - odpor tranzistoru mezi kolektorem a emitorem při jeho plném otevření (v saturaci)
  - $V_{CEsat}$  - zbytkové napětí tranzistoru mezi kolektorem a emitorem při jeho plném otevření (v saturaci)
- Schéma invertoru
  - V bázi musí být rezistor  $R1$  omezující proud
  - Přenosová charakteristika

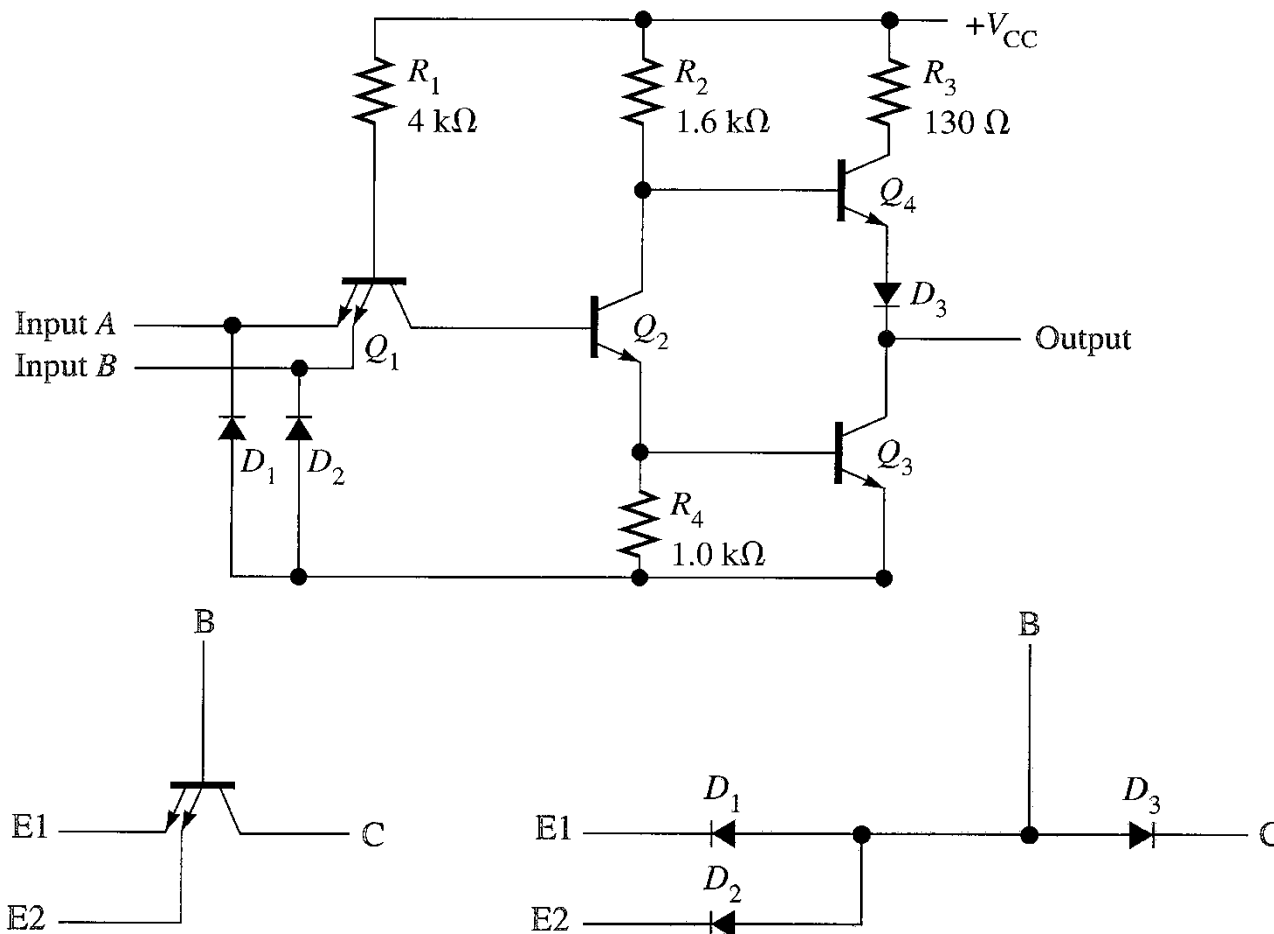


- TTL – Transistor-Transistor Logic
  - Členy jsou sestaveny z bipolárních tranzistorů, rezistorů, pomocných a ochranných diod
- Schéma invertoru TTL
  - Výstupní část se anglicky říká „totem-pole“
  - Úroveň L na výstupu bude mít lepší elektrické parametry než H

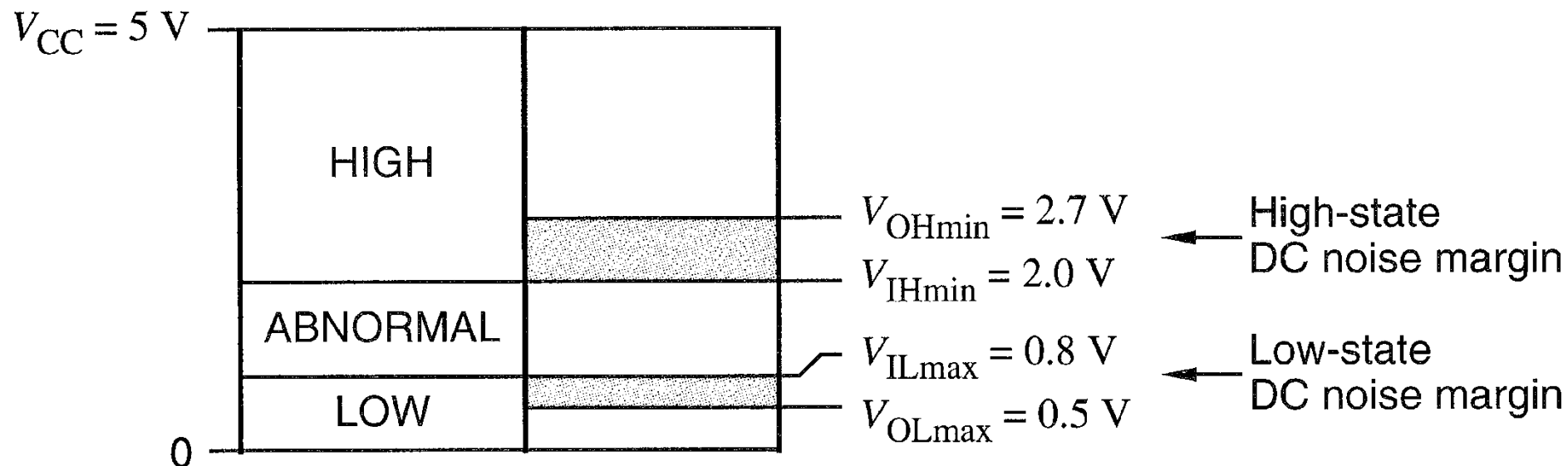




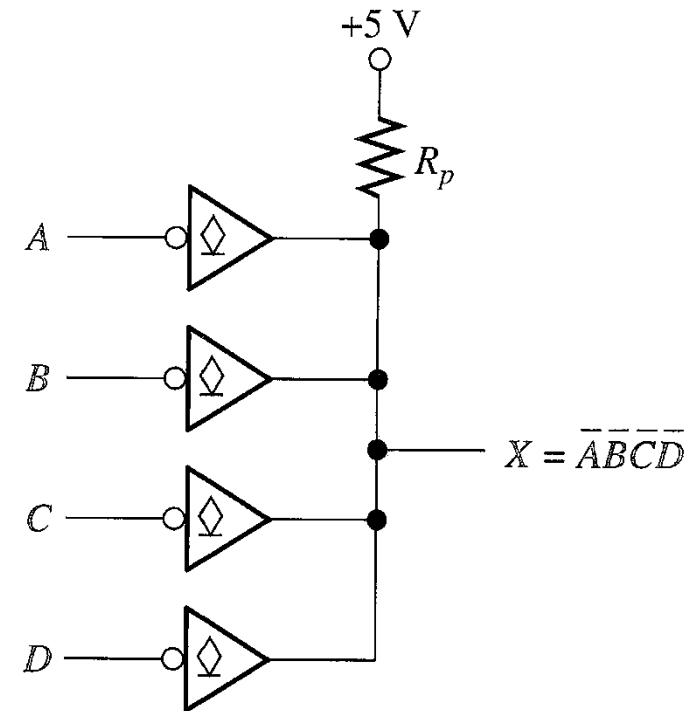
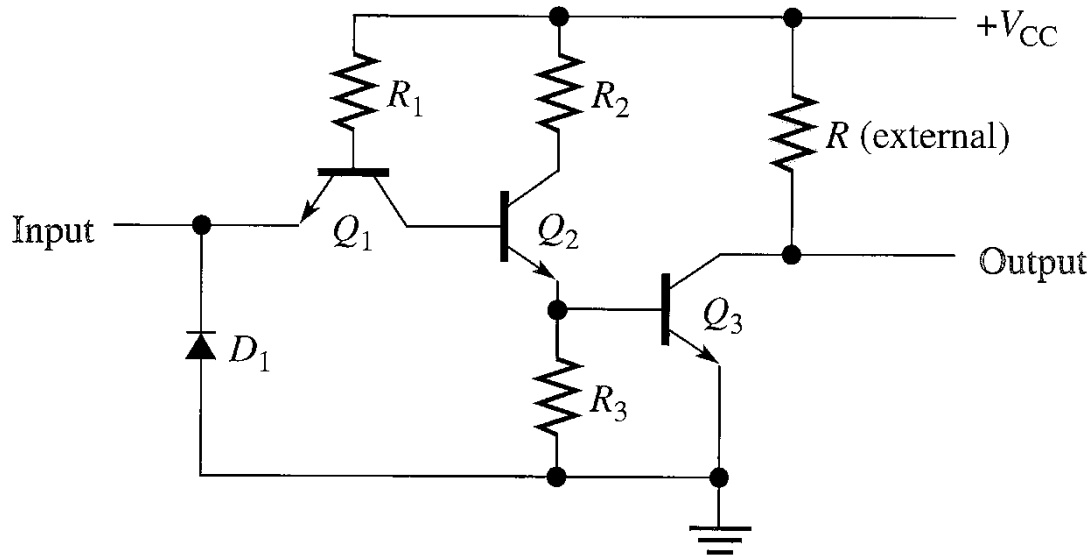
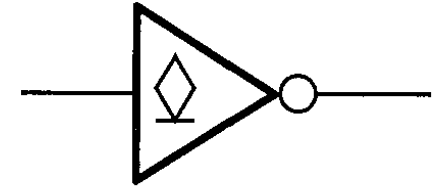
- Na vstupu je tranzistor s více emitory
  - Jedná se v podstatě o realizaci funkce AND pomocí diod



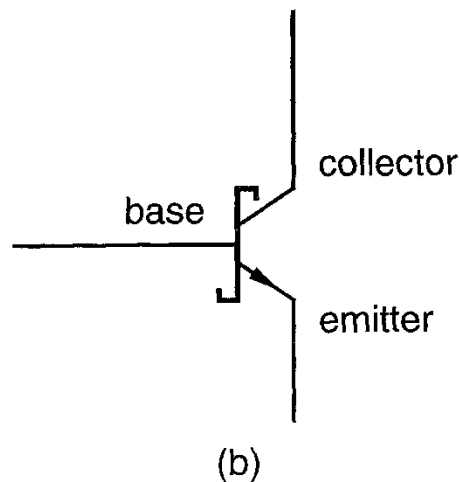
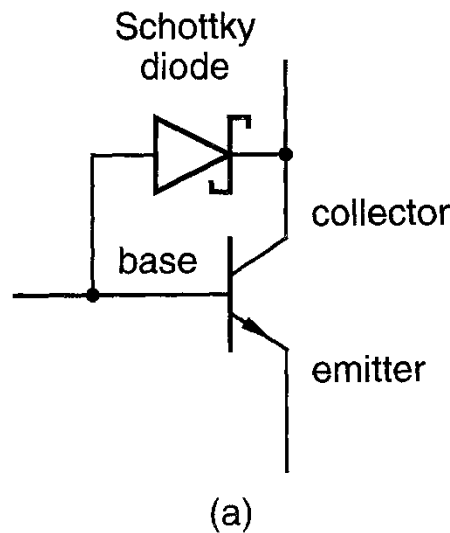
- Příklad základních rodin TTL obvodů s napájením 5V
  - 74LSxx – Low Voltage Shottky TTL
  - 74Sxx – Shottky TTL
  - 74ALSxx – Advanced Low Voltage Shottky TTL
  - 74ASxx – Advanced Shottky TTL
  - 74Fxx – Fast TTL



- Anglicky „open-collector“
  - Obvod nemá horní větev výstupní části „totem-pole“
  - Musí se použít externí rezistor
  - Lze využít např. k realizaci tzv. „montážní logiky“ (wired logic) - log. funkce je vytvořena zapojením součástek mimo IO

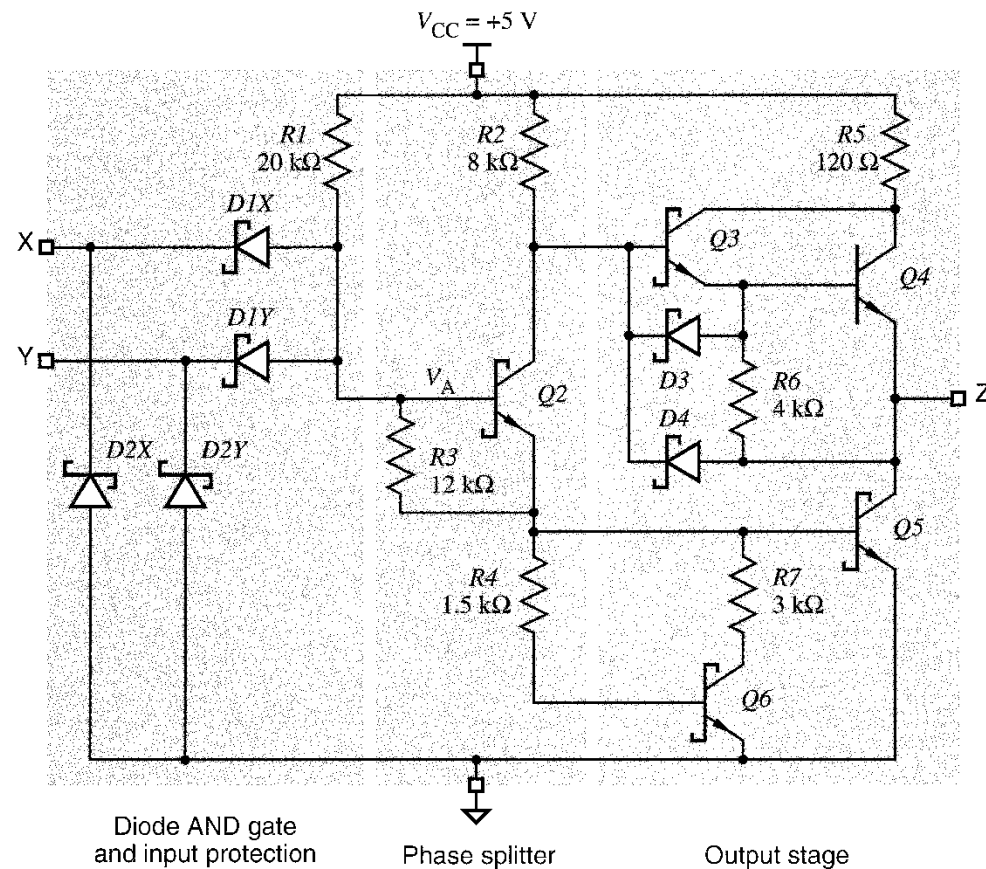


- Příklad vylepšování konstrukce log. obvodů s ohledem na rychlost
  - U normálního bipolárního tranzistoru je doba přechodu ze saturace do rozepnutého stavu poměrně dlouhá
  - Pokud se zabrání situaci, kdy se tranzistor dostane v sepnutém stavu do saturace, bude rychlost přepínání podstatně vyšší
  - Lze toho dosáhnout integrací tzv. Schottkyho diody mezi bází a kolektor (a) - výsledkem je tzv. Schottkyho tranzistor (b)

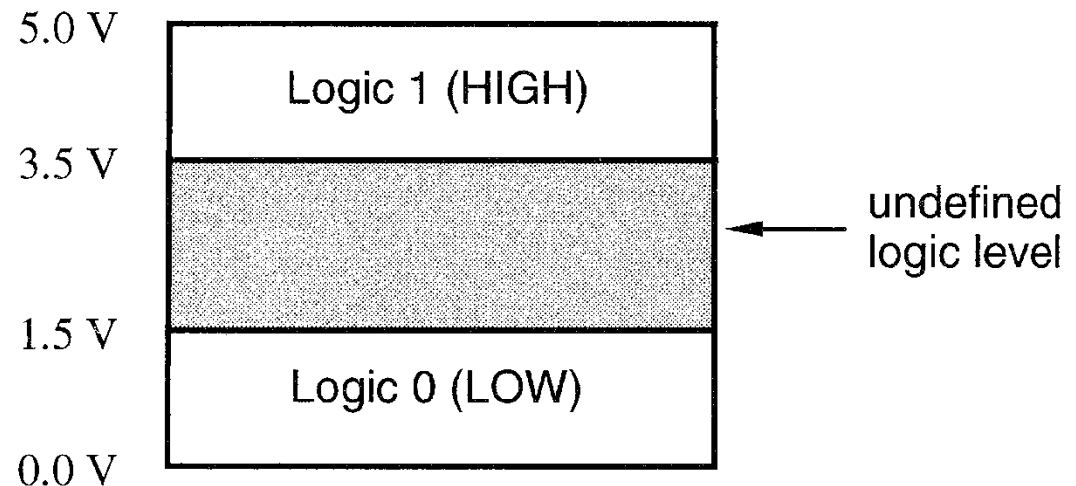
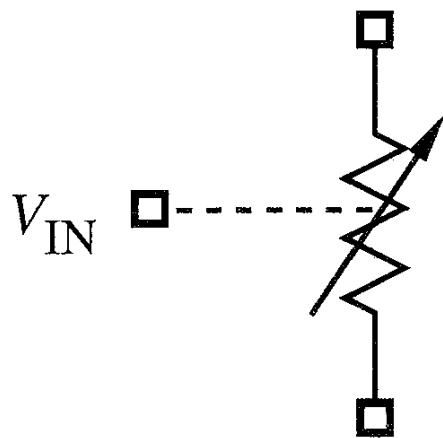


- Příklad vylepšování konstrukce log. obvodů s ohledem na rychlost a příkon
- Hradlo NAND LS-TTL
- Rychlost obvodu je dána použitím Shottkyho diod a tranzistorů
- Příkon obvodu je snížen díky použití rezistorů větších hodnot než je tomu u klasické technologie TTL
- Pozn: vstupní diody D1X a D1Y realizují log. funkci AND

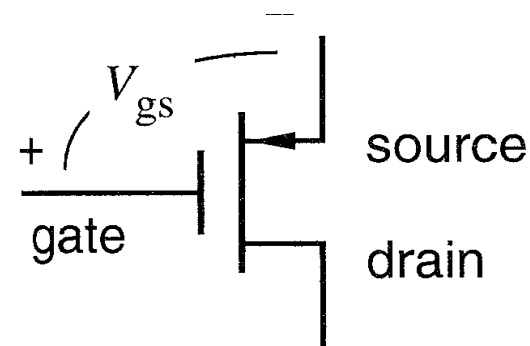
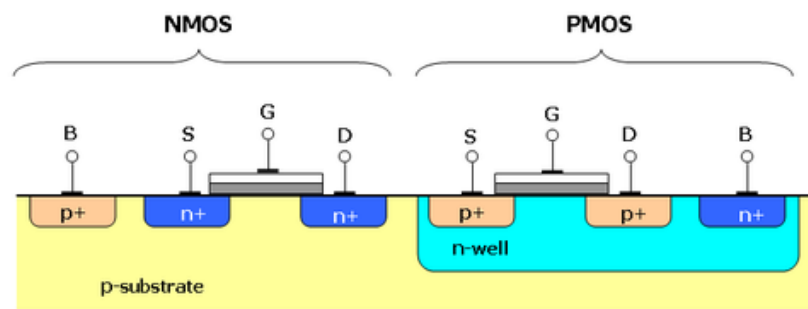
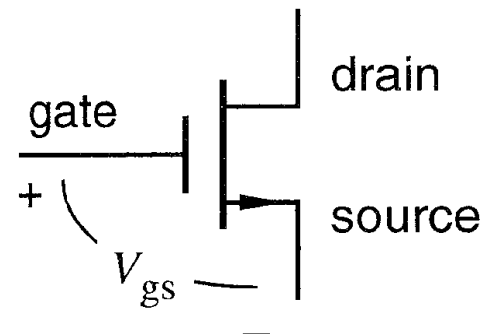
X	Y	$V_A$	Q2	Q3	Q4	Q5	Q6	$V_Z$	Z
L	L	$\leq 1.05$	off	on	on	off	off	2.7	H
L	H	$\leq 1.05$	off	on	on	off	off	2.7	H
H	L	$\leq 1.05$	off	on	on	off	off	2.7	H
H	H	1.2	on	off	off	on	on	$\leq 0.35$	L



- Význam zkratek
  - MOSFET = MOS Field Effect Transistor
  - MOS = Metal-Oxide Semiconductor ... zkratka z MOSFET
  - CMOS = Complementary MOS
- Princip činnost
  - MOSFET tranzistor pracuje jako „rezistor řízený napětím“
- Logické úrovně typického CMOS obvodu

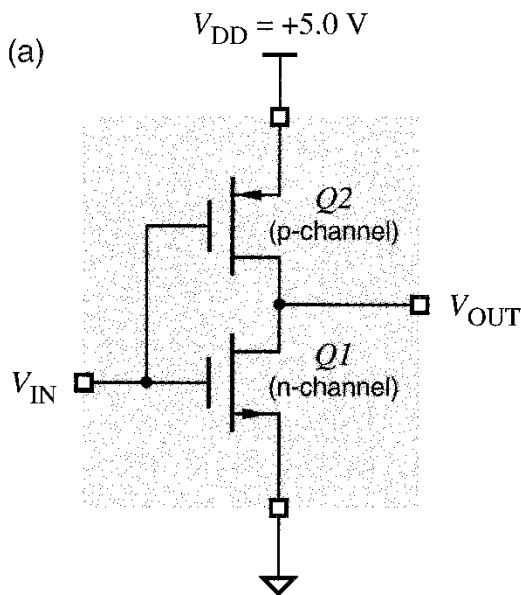
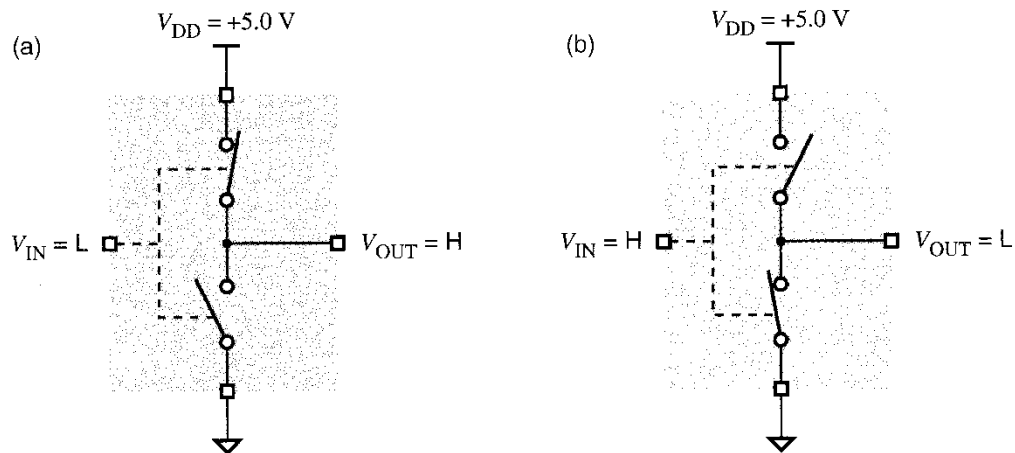


- „Gate“ - řídicí elektroda
  - Napětí mezi elektrodami gate a source ovládá činnost tranzistoru
- Tranzistor MOS s kanálem n
  - Elektroda drain je připojena na vyšší napětí než source
  - Pro  $V_{gs}=0$  má přechodový odpor mezi drain a source ( $R_{ds}$ ) desítky  $M\Omega$
  - Pro  $V_{gs}$  vyšší než prahová úroveň má  $R_{ds}$  jednotky  $\Omega$
- Tranzistor MOS s kanálem p
  - Elektroda source je na vyšším napětí než drain
  - Pro  $V_{gs}=0$  má přechodový odpor mezi drain a source ( $R_{ds}$ ) desítky  $M\Omega$
  - Pro  $V_{gs}$  nižší než prahová úroveň má  $R_{ds}$  jednotky  $\Omega$



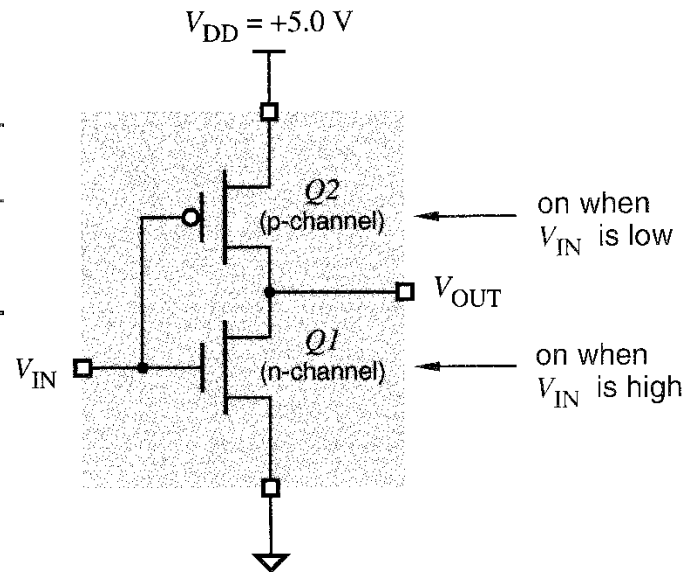
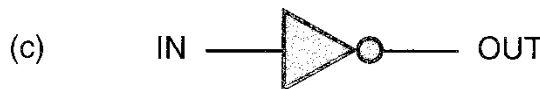
- CMOS Invertor

- Je sestaven ze dvou komplementárních tranzistorů MOSFET, z nichž vždy jeden je sepnut a druhý rozepnut



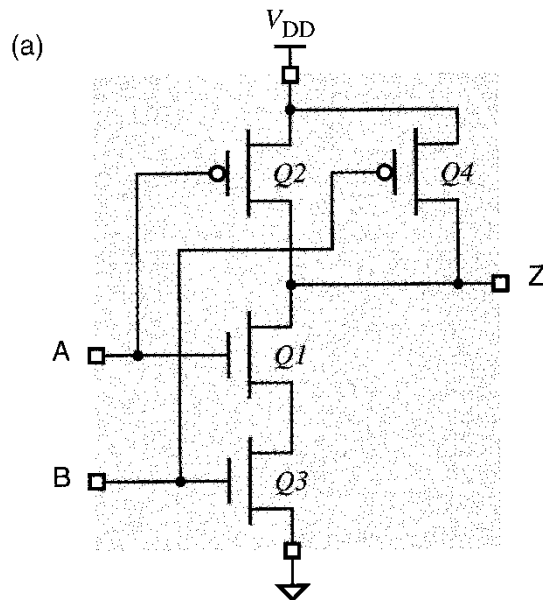
(b)

$V_{IN}$	$Q1$	$Q2$	$V_{OUT}$
0.0 (L)	off	on	5.0 (H)
5.0 (H)	on	off	0.0 (L)



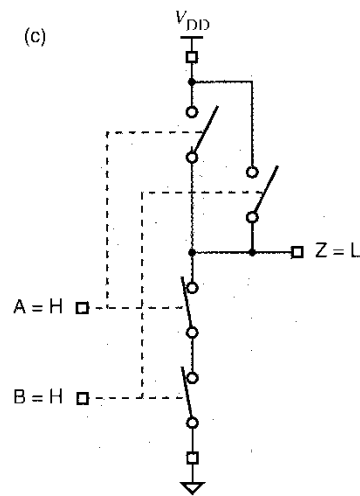
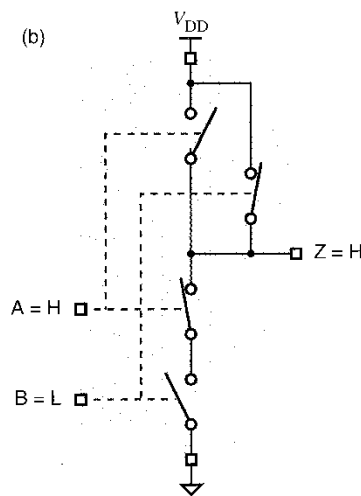
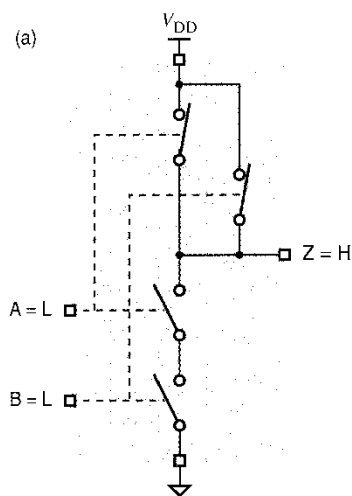
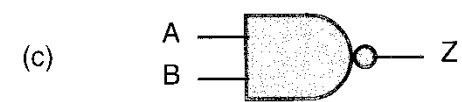


- Realizace pomocí komplementárních tranzistorů MOSFET
  - Schéma (a)
  - Pravdivostní tabulka (b)
  - Symbol (c)
- Ilustrace činnosti pomocí spínačů
  - Oba vstupy na úrovni L => výstup H (a)
  - Jeden vstup na úrovni L a druhý na úrovni H => výstup H (b)
  - Oba vstupy na úrovni H => výstup L (c)



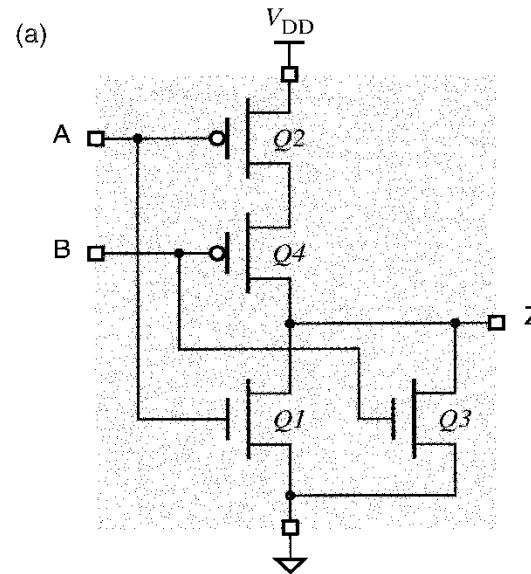
(b)

A	B	Q1	Q2	Q3	Q4	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	H
H	L	on	off	off	on	H
H	H	on	off	on	off	L



- Realizace pomocí komplementárních tranzistorů MOSFET

- Schéma (a)
- Pravdivostní tabulka (b)
- Symbol (c)



(b)

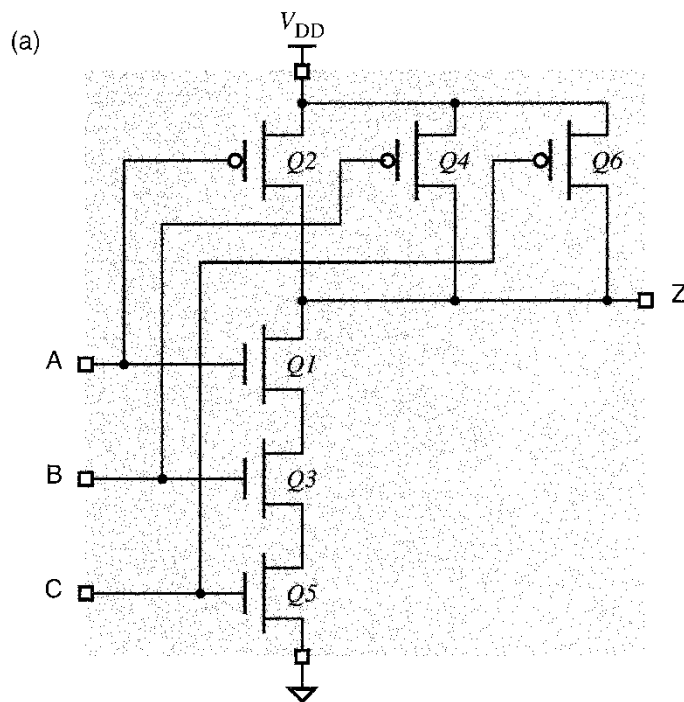
A	B	Q1	Q2	Q3	Q4	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	L
H	L	on	off	off	on	L
H	H	on	off	on	off	L



- Vlastnosti

- Hradlo CMOS NOR je pomalejší než hradlo CMOS NAND, neboť při stejných rozměrech mají tranzistory s p-kanálem větší přechodový odpor v otevřeném stavu
- Pokud jsou tedy tranzistory s p-kanálem (Q2 a Q4) řazeny v sérii, budou mít pomalejší odezvu díky pomalejšímu nabíjení parazitních kapacit

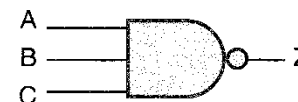
- Větší počet vstupů se realizuje přidáním dalších dvojic komplementárních tranzistorů
- Příklad tří-vstupového hradla NAND
  - Schéma (a)
  - Pravdivostní tabulka (b)
  - Symbol (c)



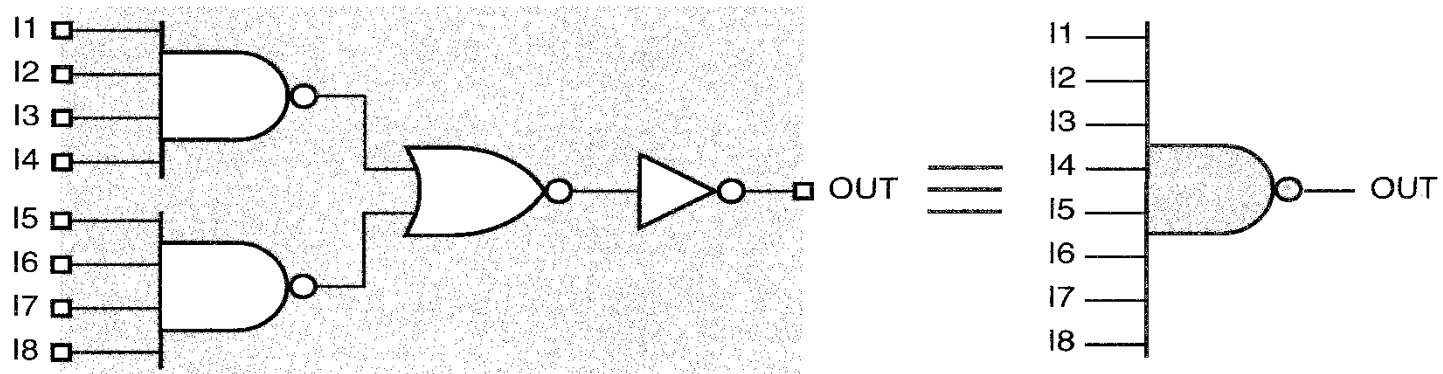
(b)

A	B	C	Q1	Q2	Q3	Q4	Q5	Q6	Z
L	L	L	off	on	off	on	off	on	H
L	L	H	off	on	off	on	on	off	H
L	H	L	off	on	on	off	off	on	H
L	H	H	off	on	on	off	on	off	H
H	L	L	on	off	off	on	off	on	H
H	L	H	on	off	off	on	on	off	H
H	H	L	on	off	on	off	off	on	H
H	H	H	on	off	on	off	on	off	L

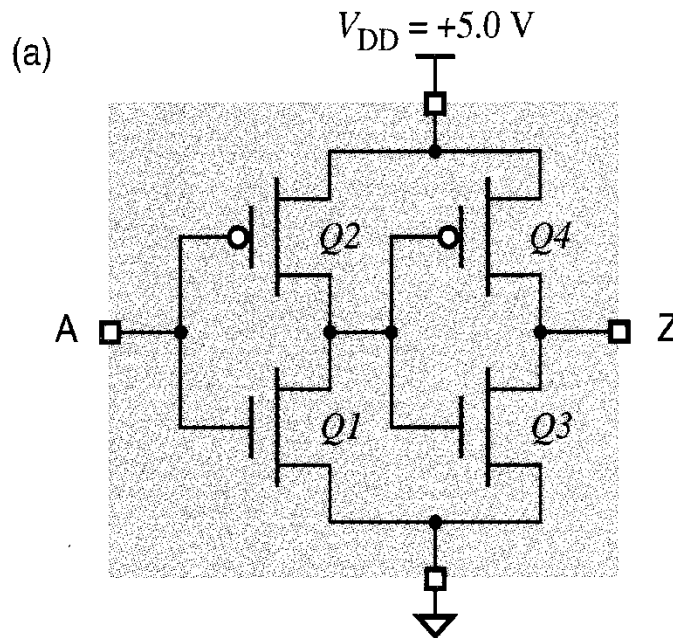
(c)



- Anglicky „fan-in“ - počet vstupů, které může mít log. člen vyrobený danou technologií
- Technologie CMOS
  - Teoreticky je možno sestavovat log. členy s více vstupy pouhým řazením více tranzistorů MOSFET
  - Prakticky je však, z hlediska konečných přechodových odporů a potřebných rychlostí odezvy, tato možnost limitována
    - U hradel NAND typicky na 6
    - U hradel NOR typicky na 4
  - Větší množství vstupů se realizuje pomocí kaskády log. členů s méně vstupy



- Zařazením dvou invertorů za sebe vznikne tzv. buffer
  - Slouží pro distribuci signálu tam, kde je např. třeba obnovit integritu signálu, budit více vstupů následujících log. členů (hodinový signál), apod.
  - Schéma (a)
  - Pravdivostní tabulka (b)
  - Symbol (c)

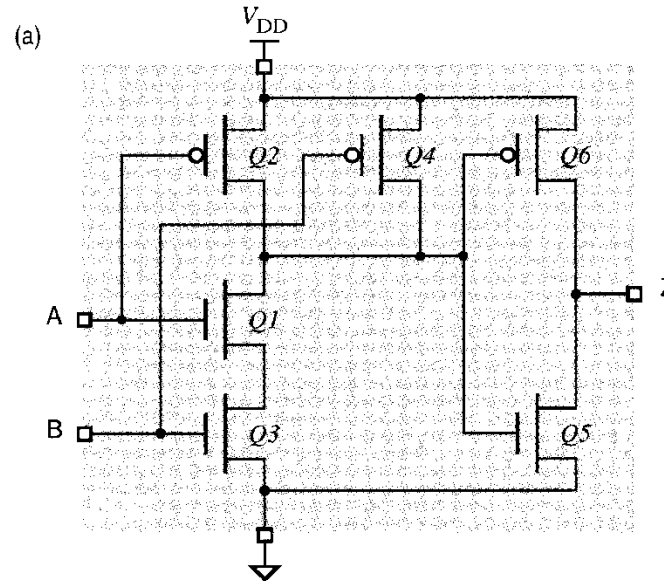


(b)

A	Q1	Q2	Q3	Q4	Z
L	off	on	on	off	L
H	on	off	off	on	H

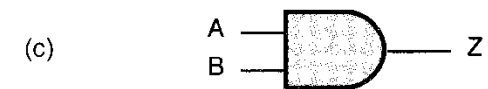


- Hradlo AND lze nejjednodušeji sestavit zařazením invertoru za hradlo NAND
  - Složitější, dražší, větší zpoždění než NAND
  - Schéma (a)
  - Pravdivostní tabulka (b)
  - Symbol (c)



(b)

A	B	Q1	Q2	Q3	Q4	Q5	Q6	Z
L	L	off	on	off	on	on	off	L
L	H	off	on	on	off	on	off	L
H	L	on	off	off	on	on	off	L
H	H	on	off	on	off	off	on	H

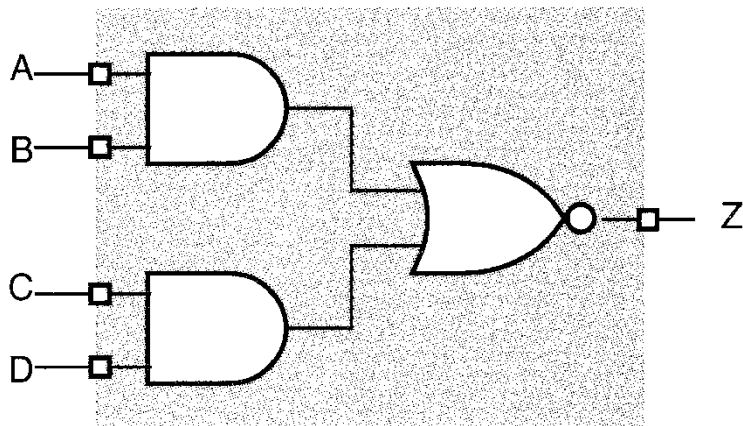


- => častěji se vyrábějí invertující hradla (NAND a NOR)...

- Příklad implementace logických forem
  - V praxi se často využívají log. struktury, které realizují jistou formu disjunktivní či konjunktivní formy, a usnadňují tak implementaci kombinačních obvodů
  - Výhodou je skutečnost, že zpoždění těchto obvodů je podobné jako v případě členů NAND či NOR, i když realizují dvoustupňový logický obvod

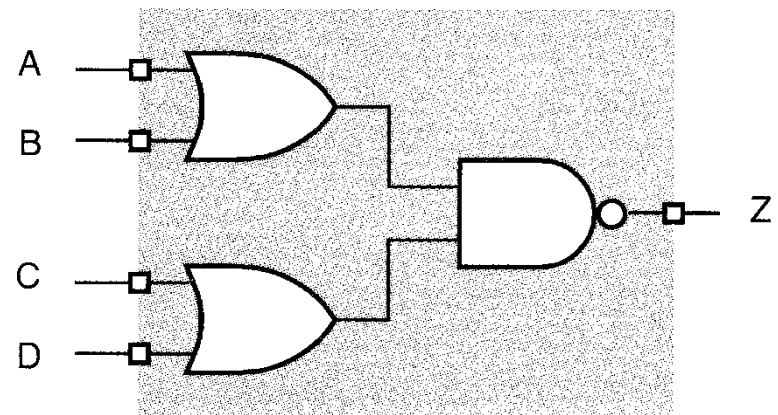
## • AND-OR-INVERT

- Disjunktivní forma s invertorem



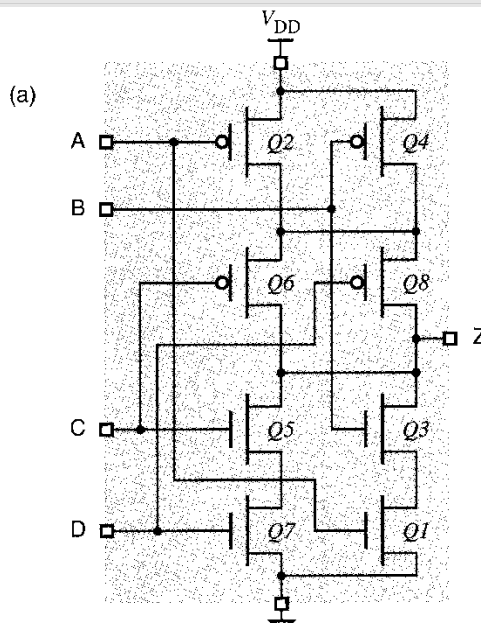
## • OR-AND-INVERT

- Konjunktivní forma s invertorem



- AND-OR-INVERT

- Schéma (a)
- Pravdivostní tabulka (b)

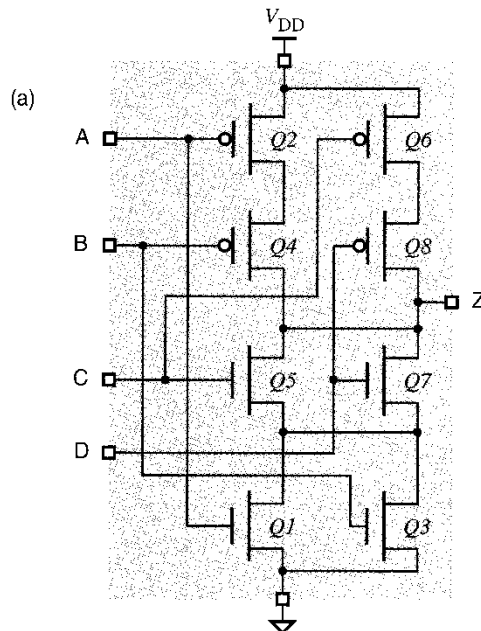


(b)

A	B	C	D	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Z
L	L	L	L	off	on	off	on	off	on	off	on	H
L	L	L	H	off	on	off	on	off	on	on	off	H
L	L	H	L	off	on	off	on	on	off	off	on	H
L	L	H	H	off	on	off	on	on	off	on	off	L
L	H	L	L	off	on	on	off	off	on	off	on	H
L	H	L	H	off	on	on	off	off	on	on	off	H
L	H	H	L	off	on	on	off	on	off	off	on	H
L	H	H	H	off	on	on	off	on	off	on	off	L
H	L	L	L	on	off	off	on	off	on	off	on	H
H	L	L	H	on	off	off	on	on	off	on	on	H
H	L	H	L	on	off	off	on	on	off	off	on	H
H	L	H	H	on	off	off	on	on	off	on	off	L
H	H	L	L	on	off	on	off	off	on	off	on	L
H	H	L	H	on	off	on	off	off	on	on	off	L
H	H	H	L	on	off	on	off	on	off	off	on	L
H	H	H	H	on	off	on	off	on	off	on	off	L

- OR-AND-INVERT

- Schéma (a)
- Pravdivostní tabulka (b)

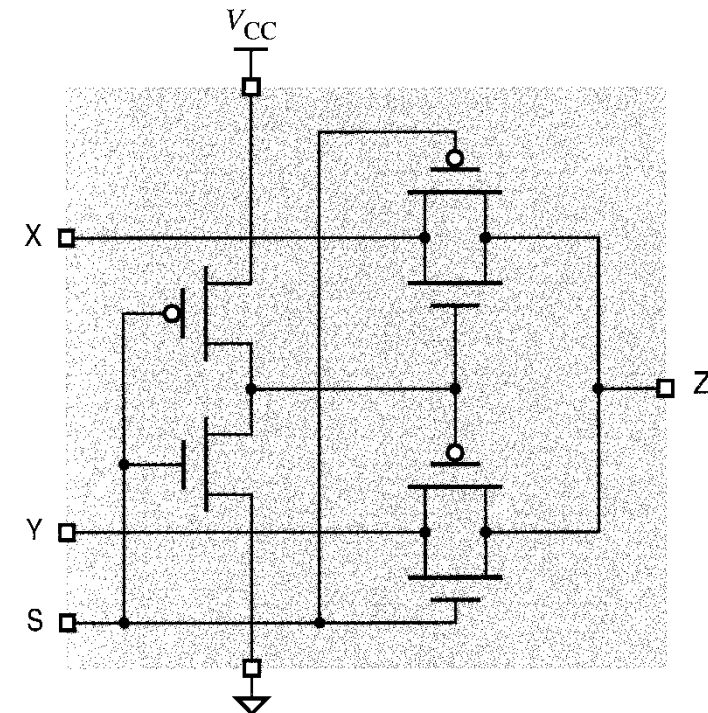
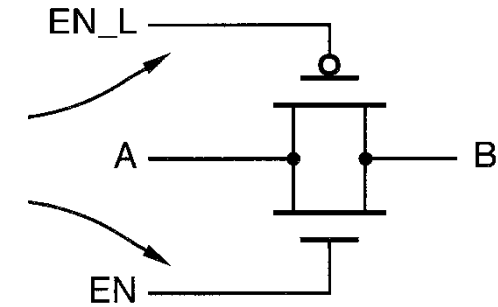


(b)

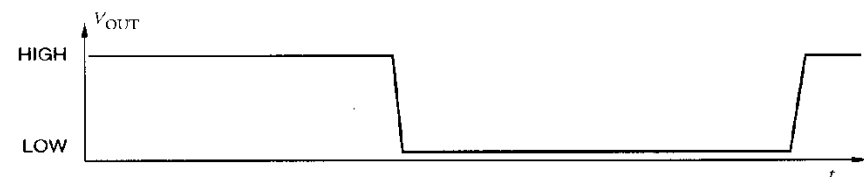
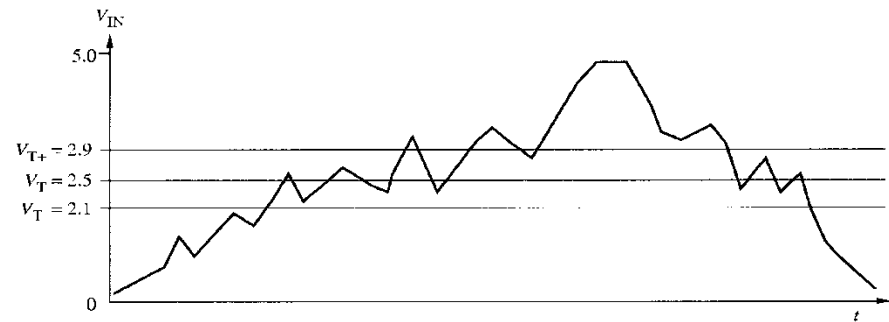
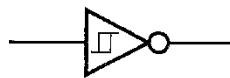
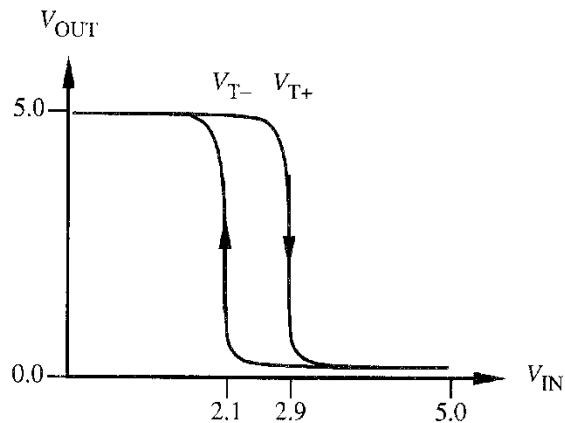
A	B	C	D	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Z
L	L	L	L	off	on	off	on	off	on	off	on	H
L	L	L	H	off	on	off	on	off	on	on	off	H
L	L	H	L	off	on	off	on	on	off	off	on	H
L	L	H	H	off	on	off	on	on	off	on	off	H
L	H	L	L	off	on	on	off	off	on	on	off	L
L	H	H	L	off	on	on	off	on	off	off	on	L
L	H	H	H	off	on	on	off	on	off	on	off	L
H	L	L	L	on	off	off	on	off	on	off	on	H
H	L	L	H	on	off	off	on	off	on	on	off	L
H	L	H	L	on	off	off	on	on	off	off	on	L
H	L	H	H	on	off	off	on	on	off	on	off	L
H	H	L	L	on	off	on	off	off	on	on	on	H
H	H	L	H	on	off	on	off	off	on	on	off	L
H	H	H	L	on	off	on	off	on	off	off	on	L
H	H	H	H	on	off	on	off	on	off	on	off	L



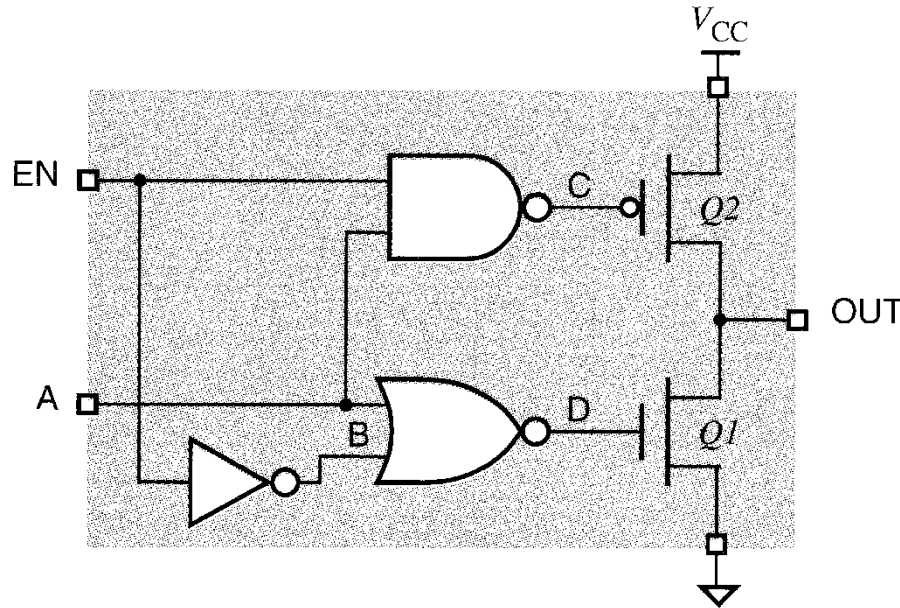
- Přenosový člen (anglicky „transmission gate“)
  - Funguje jako řízený spínač
  - Je sestaven z komplementárních tranzistorů
    - Oba jsou otevřeny – člen přenáší signál (malá impedance mezi A a B)
    - Oba jsou zavřeny – člen nepřenáší signál (velká impedance mezi A a B)
- Používá se v složitějších strukturách
  - Např. klopných obvodech
  - Multiplexorech (viz obr. vpravo)



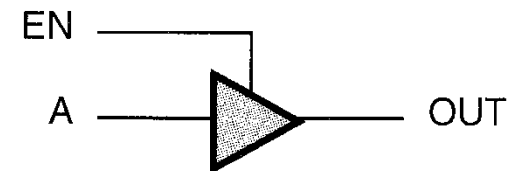
- Využívá vnitřní zpětné vazby pro posun prahových úrovní podle toho, do které úrovně přechází
  - L do H ... větší hodnota (např. 2,9 V)
  - H do L ... menší hodnota (např. 2,1 V)
- Tento jev nazýváme hysterezí
  - Zlepšuje šumovou odolnost obvodů
- Přenosová charakteristika
- Symbol
- Časový diagram



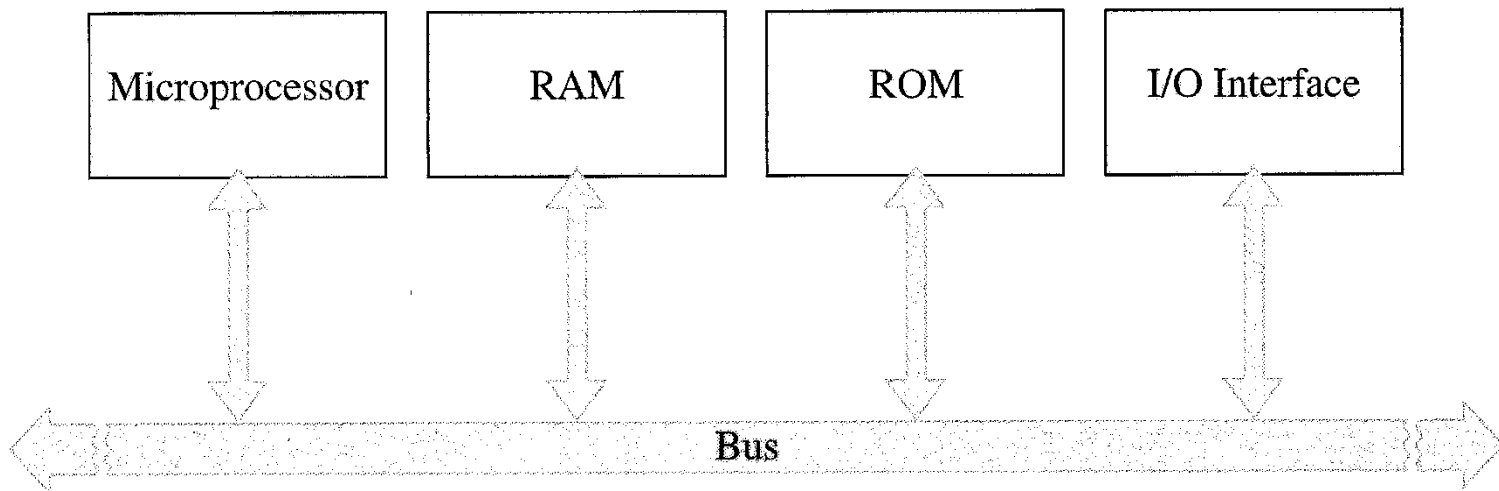
- Anglicky „three-state buffer“ nebo „tri-state buffer“
  - Umožňuje odpojit výstup obvodu
  - Tímto uvedeme obvod do tzv. třetího stavu či stavu vysoké impedance („high impedance“, Z)
  - Obvod má tedy tři stavy – L, H a Z
  - Typicky se používá pro realizaci sběrnic
  - Schéma, pravdivostní tabulka a symbol



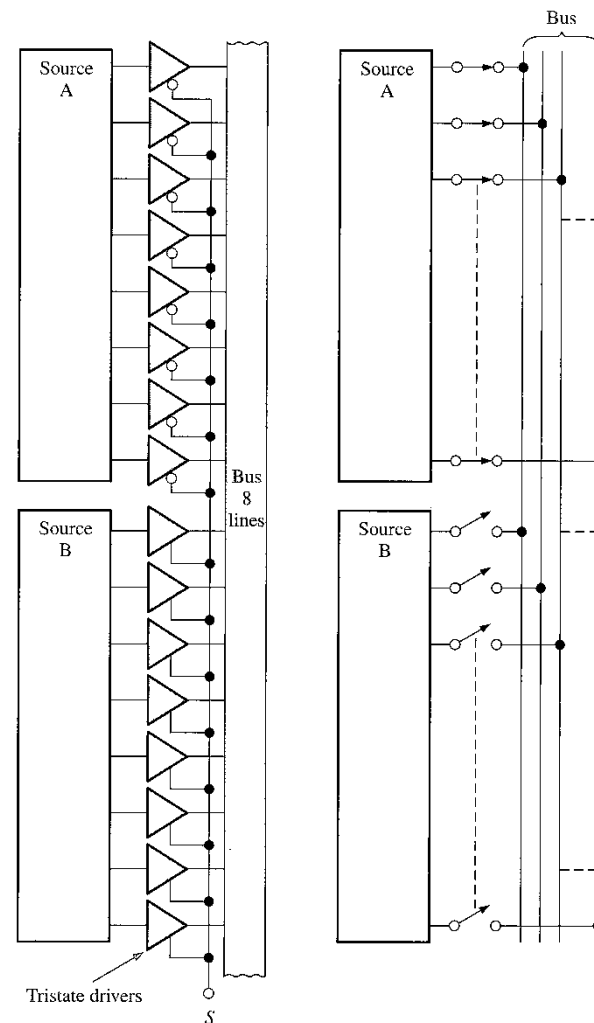
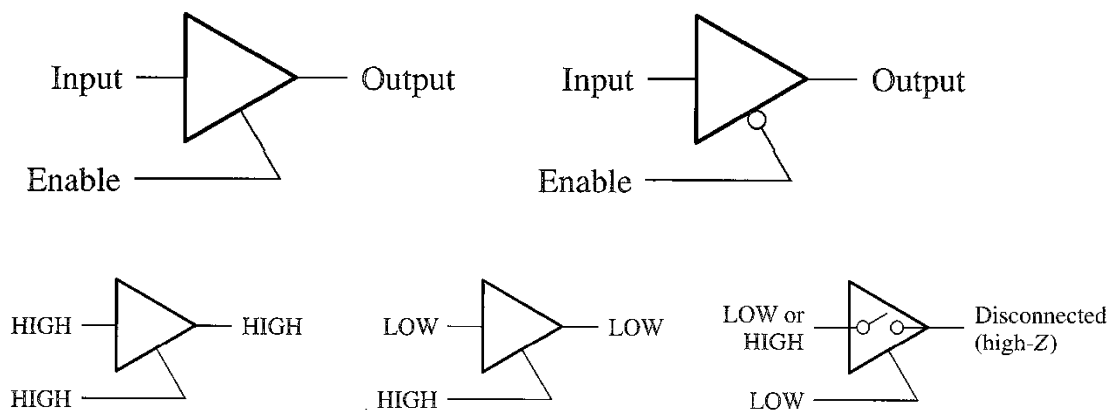
EN	A	B	C	D	Q1	Q2	OUT
L	L	H	H	L	off	off	Hi-Z
L	H	H	H	L	off	off	Hi-Z
H	L	L	H	H	on	off	L
H	H	L	L	L	off	on	H



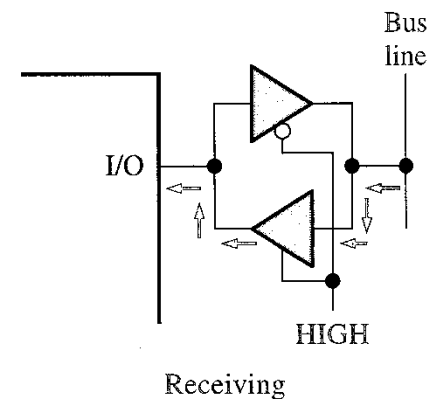
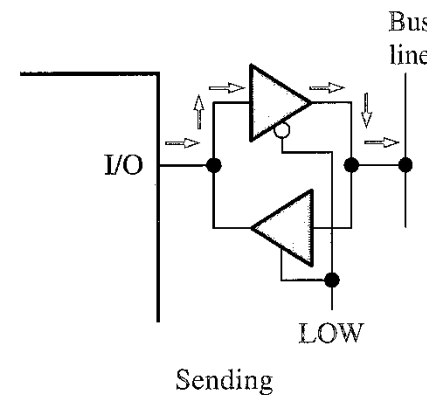
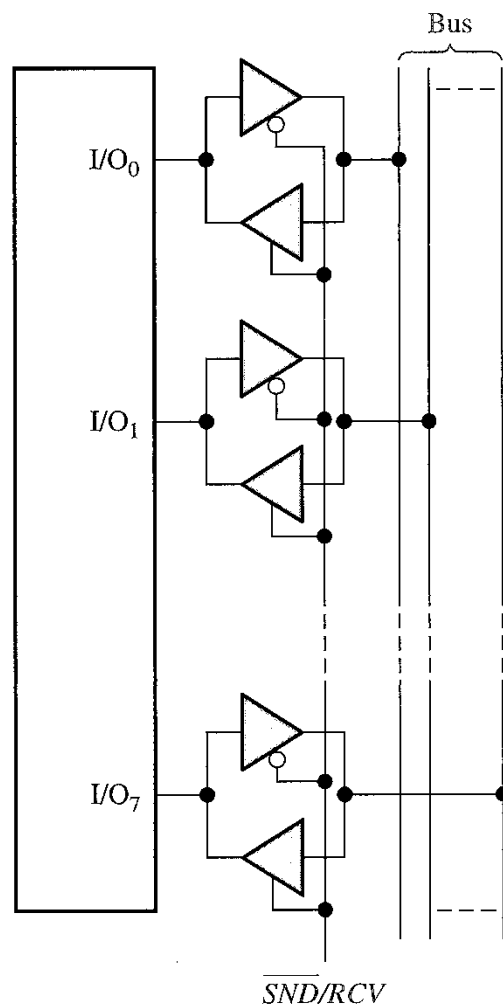
- Sběrnice
  - Jednotlivé subsystémy typického počítače jsou propojeny sběrnicí
  - V jednom okamžiku se přenáší data z jednoho zdroje do jednoho cíle - datové přenosy jsou multiplexovány v čase (pomalé)
  - Výsledkem je zjednodušení propojovací infrastruktury



- Třístavový budič slouží jako spínač
  - Může být ovládán signálem Enable aktivním v nule či v jedničce
- Příklad sběrnice se zdroji dat A a B
  - Signálem S se vybírá jeden z nich

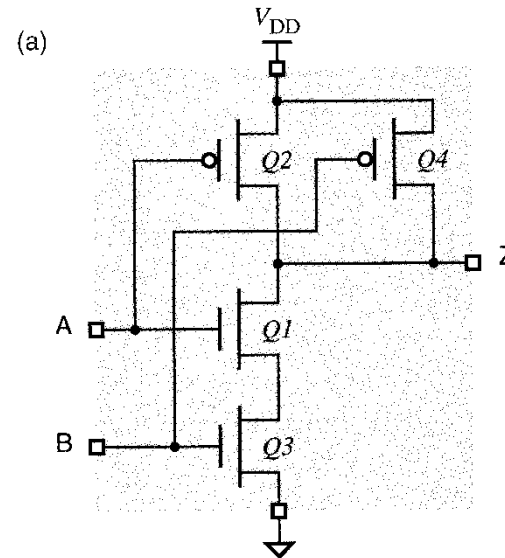


- Často je třeba přenášet data oběma směry
  - Třístavové budiče je třeba zapojit anti-paralelně
  - Vždy je aktivní jeden, nebo druhý



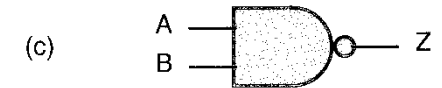
- Standardní CMOS NAND

- Tranzistory Q2 a Q4 představují řízený rezistor (pull-up)
- Schéma (a)
- Pravdivostní tabulka (b)
- Symbol (c)



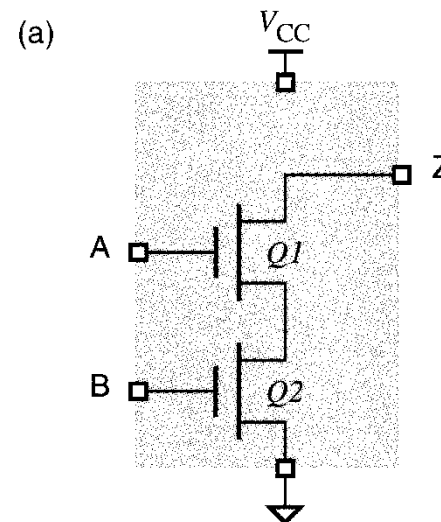
(b)

A	B	Q1	Q2	Q3	Q4	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	H
H	L	on	off	off	on	H
H	H	on	off	on	off	L



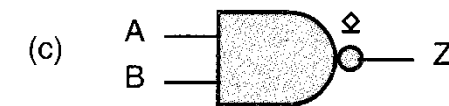
- Open-drain CMOS NAND

- Obdoba otevřeného kolektoru (open-collector) obvodů TTL
- Značí se podtrženým kosočtvercem
- Výstup je buď L, nebo otevřený
- Schéma (a)
- Pravdivostní tabulka (b)
- Symbol (c)

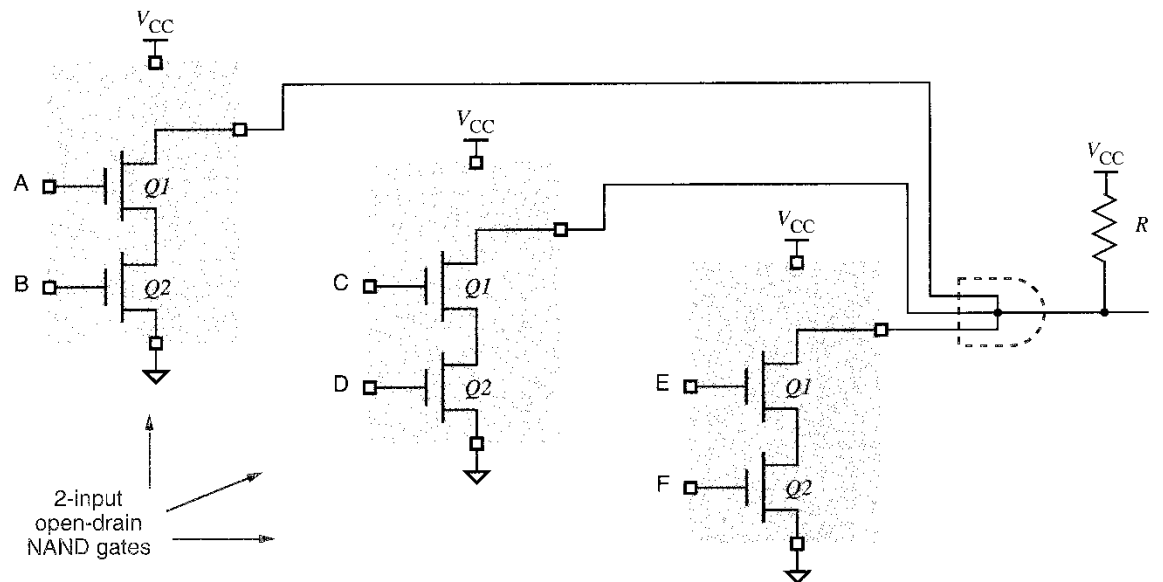


(b)

A	B	Q1	Q2	Z
L	L	off	off	open
L	H	off	on	open
H	L	on	off	open
H	H	on	on	L

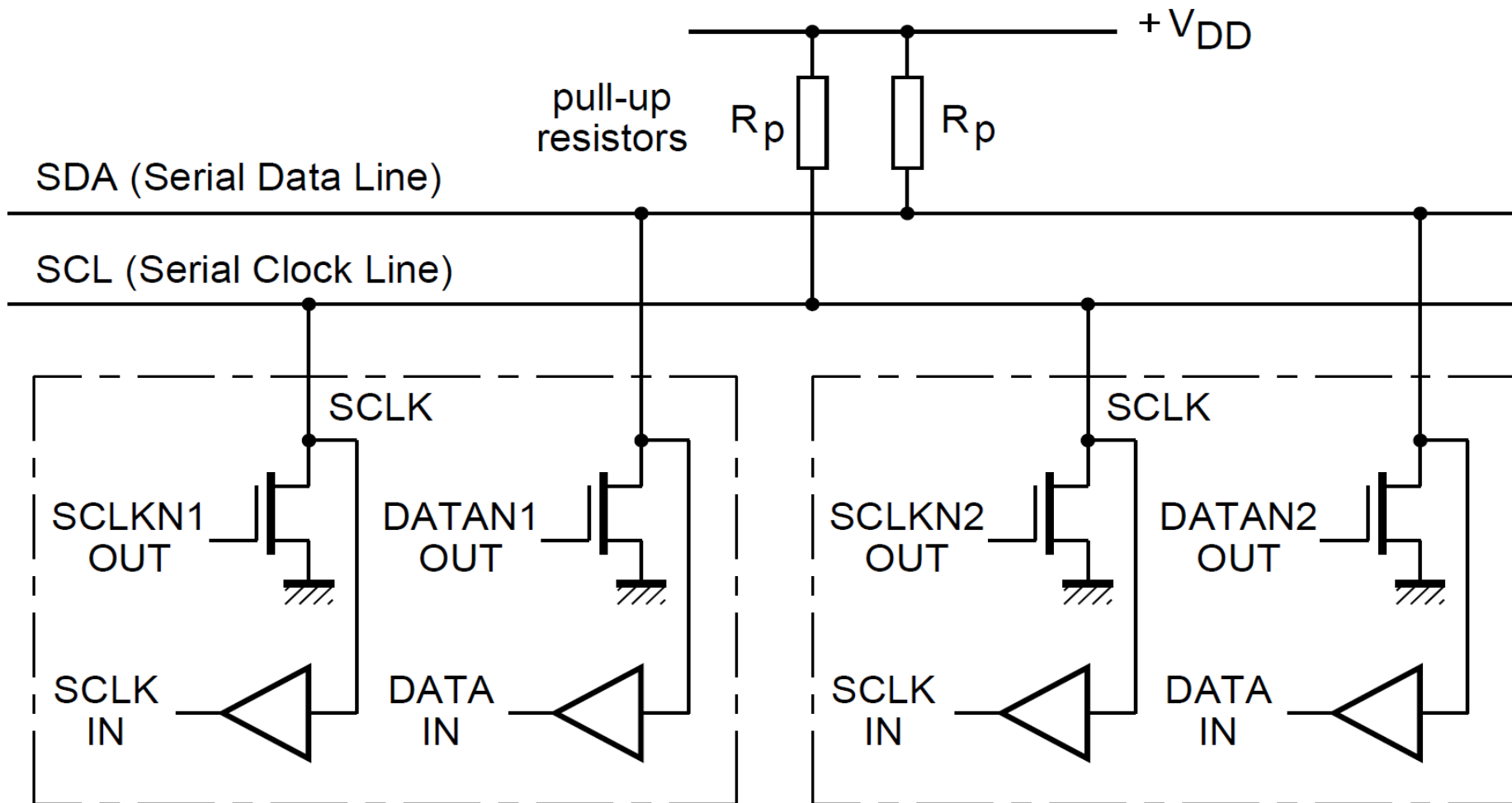


- Použití pro realizaci tzv. montážní logiky („wired logic“)
  - Spojením více výstupů členu open-drain dostáváme tzv. montážní člen AND
  - Pokud bude kterýkoliv z výstupů jednotlivých členů v L, bude i celkový výstup v L = log. funkce AND (wired-and)
  - Montážní říkáme proto, že danou funkci realizujeme např. na desce s plošnými spoji (že ji „zadrátujeme“)
  - Výsledný obvod realizuje funkci AND-OR (disjunktivní formu)



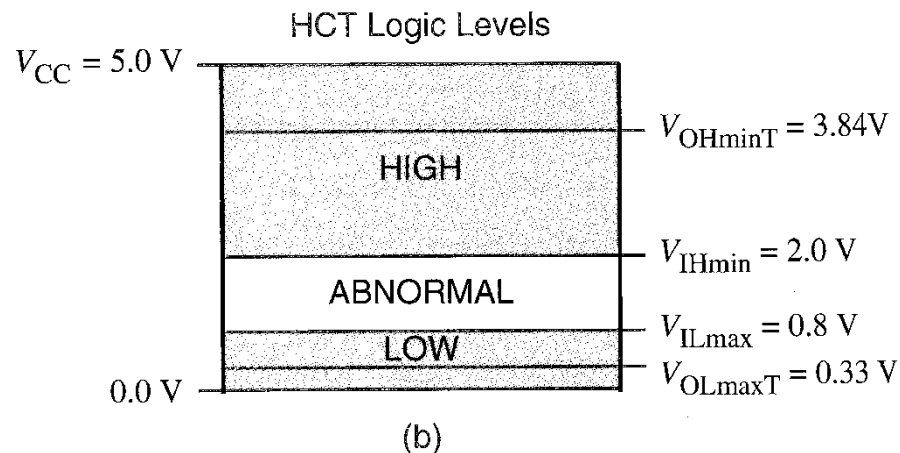
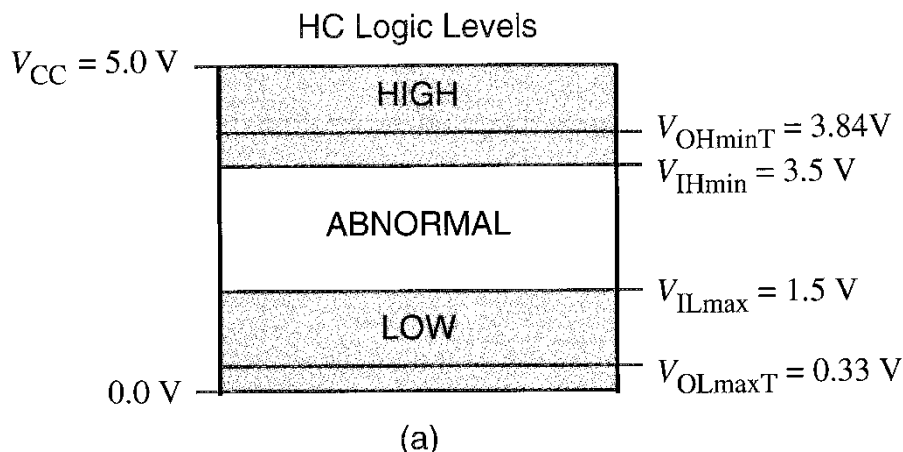
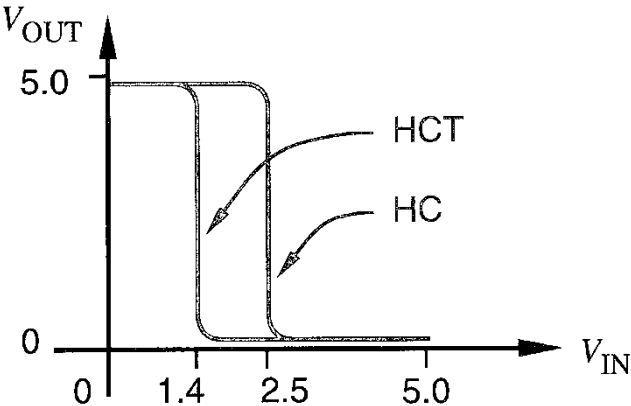


- Sběrnice I<sup>2</sup>C (Inter Integrated Circuits)

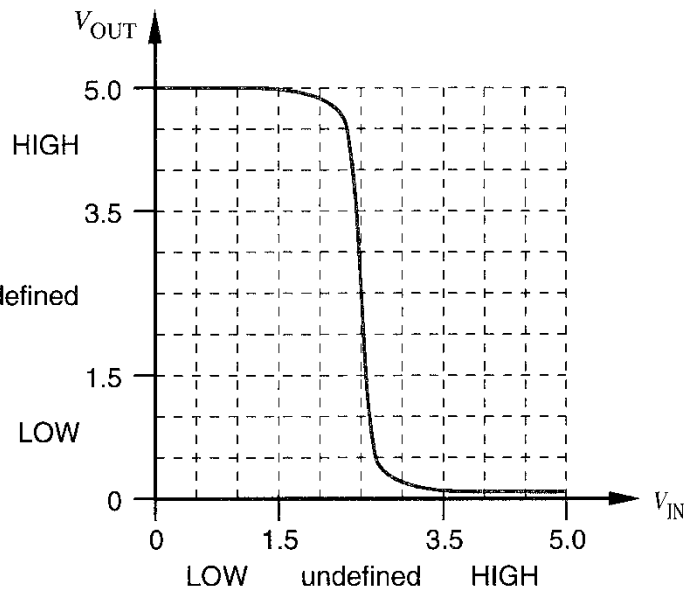


Zdroj: Philips Semiconductors: The I<sup>2</sup>C-bus specification, version 2.1 [online], 2002 [cit. 5.1.2012]. Dostupné z WWW: [http://www.classic.nxp.com/acrobat\\_download2/literature/9398/39340011.pdf](http://www.classic.nxp.com/acrobat_download2/literature/9398/39340011.pdf).

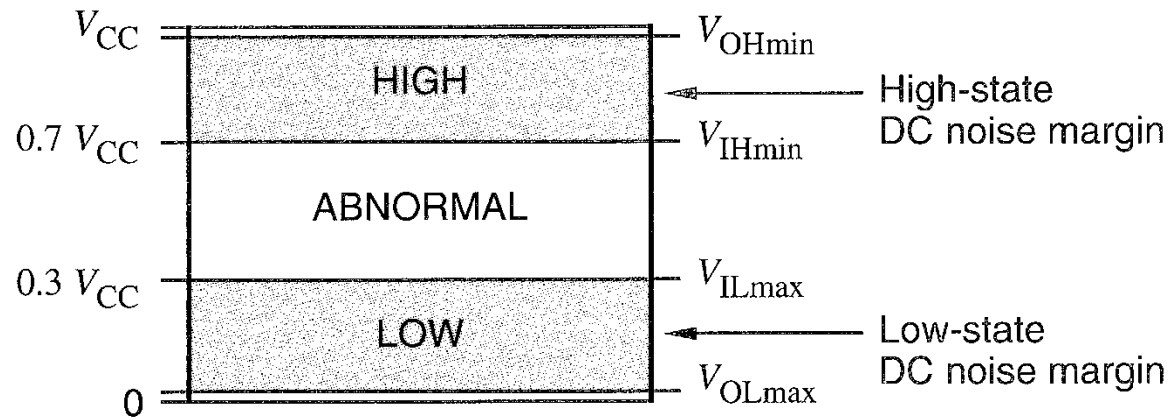
- Existuje řada stavebnic či tzv. rodin (anglicky „families“) CMOS obvodů
  - Rodina – skupina obvodů, která má stejné  $V_{OUT}$  elektrické vlastnosti
- Příklad log. úrovní
  - Převodní charakteristiky (vpravo)
  - HC – High-speed CMOS (a)
  - HCT – HC TTL Compatible (b)
    - Log. úrovně jsou kompatibilní s technologií TTL



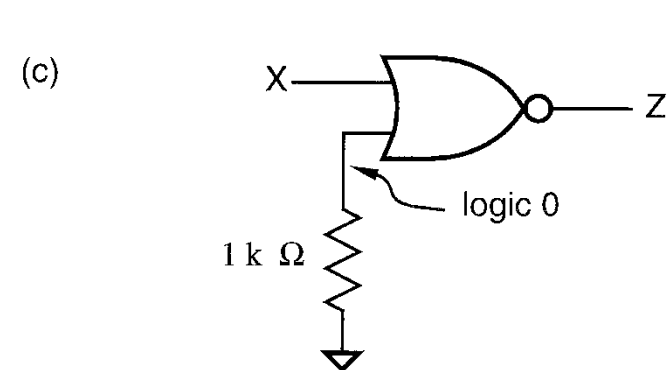
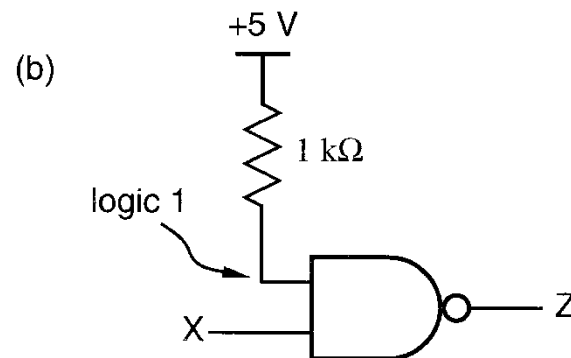
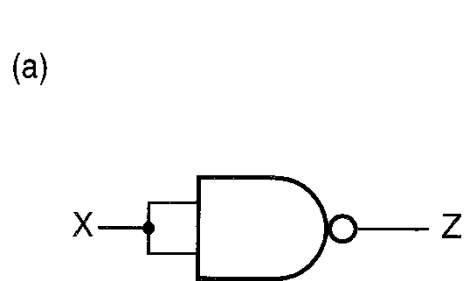
- Převodní charakteristika CMOS invertoru
  - 5V napájení
  - $V_{in}$  - vstupní napětí
  - $V_{out}$  - výstupní napětí



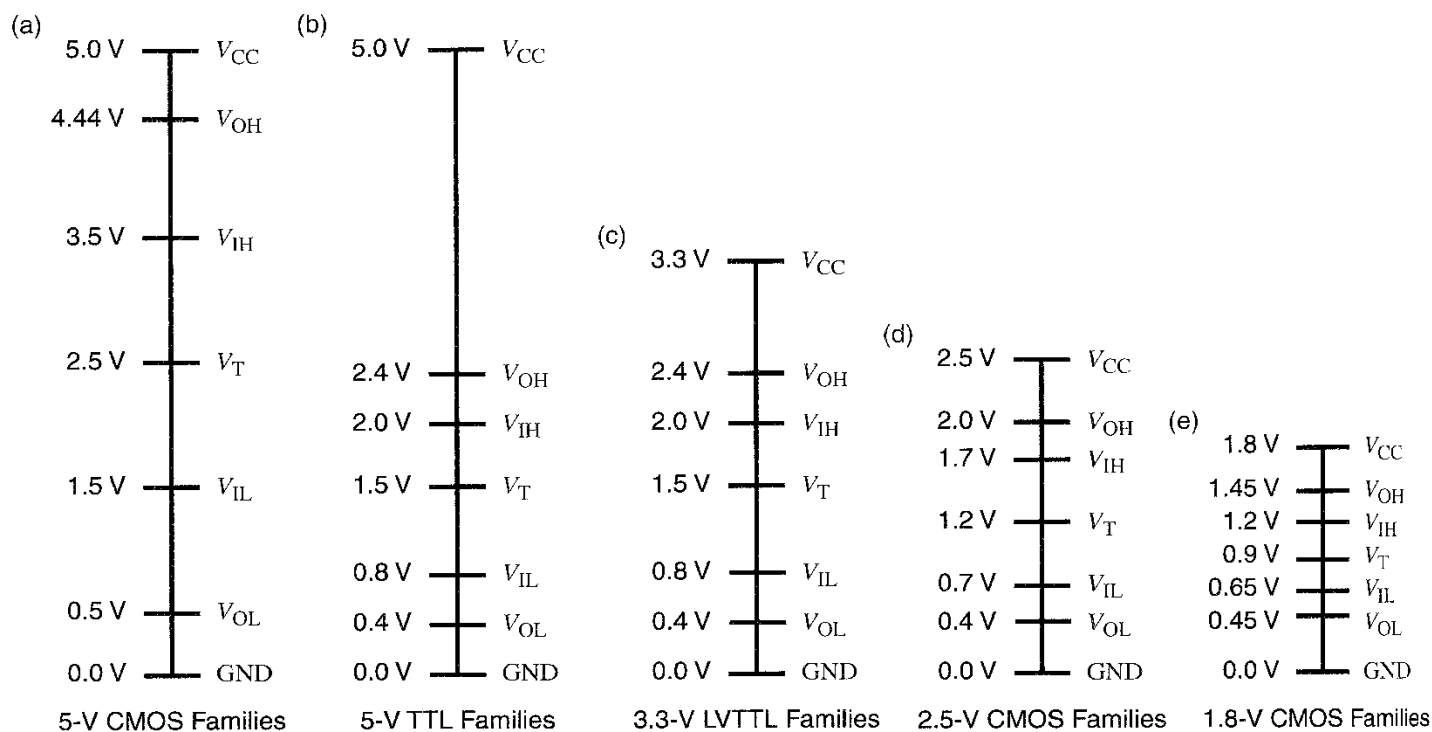
- Příklad CMOS HC - logické úrovně a rozsah odolnosti proti rušení (noise margins)
  - $V_{CC}$  napájecí napětí
  - $V_{OHmin}$  min. výstupní napětí H =  $V_{CC} - 0,1 V$
  - $V_{IHmin}$  min. vstupní napětí H = 70 %  $V_{CC}$
  - $V_{ILmax}$  max. vstupní napětí L = 30 %  $V_{CC}$
  - $V_{OLmax}$  max. výstupní napětí L = zem + 0,1 V



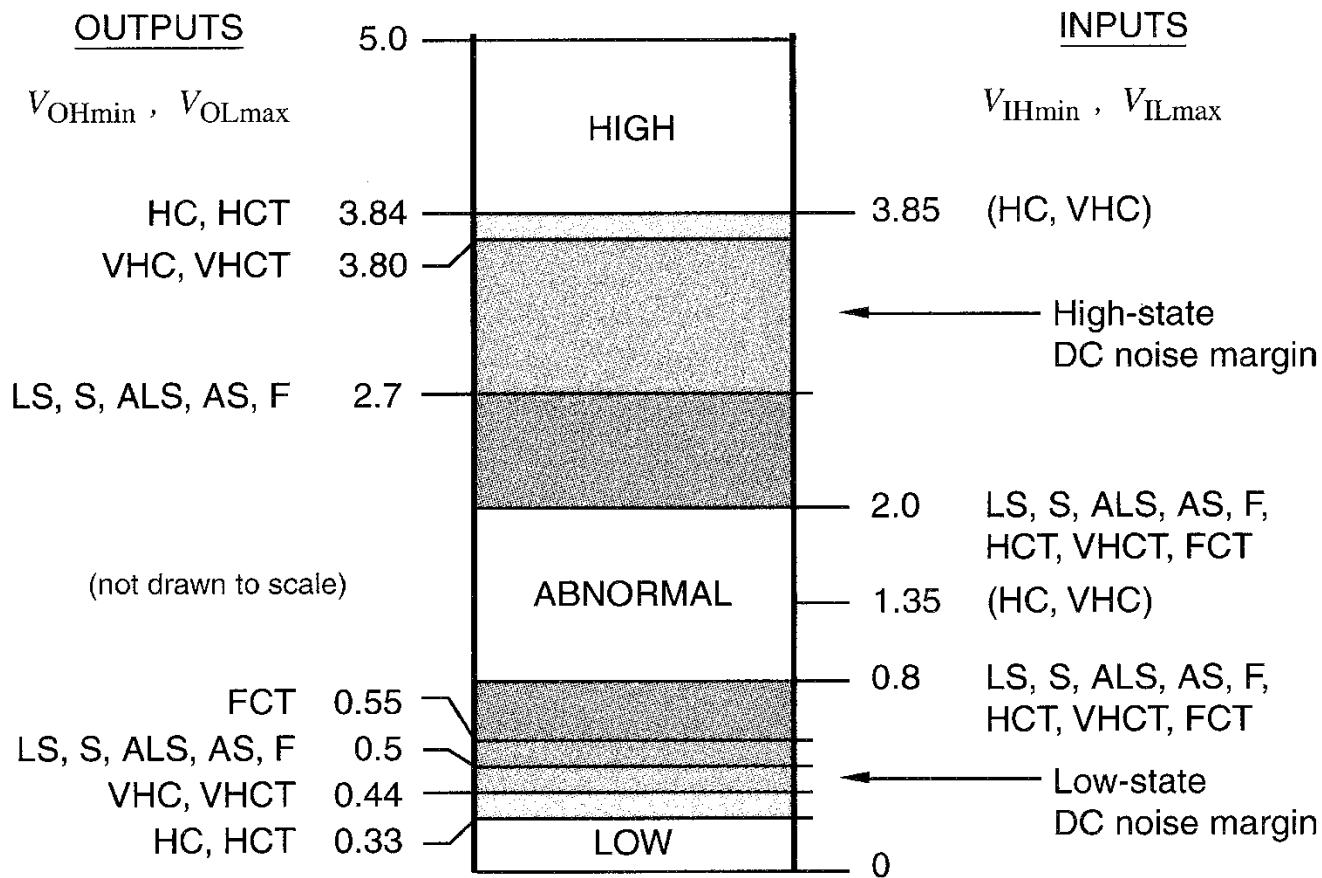
- Nevyužitě vstupy
  - Je třeba „ošetřit“ - připojit na takové log. úrovně, které neovlivní chování obvodu a zajistí, že vstup nebude náchylný k rušení
- Způsob ošetření
  - Připojení k jiným vstupům (a)
  - Vstup hradla NAND na zdroj log. 1 – napájecí napětí přes ochranný rezistor (b)
  - Vstup hradla NOR na zdroj log. 0 – zem napájecího napětí přes rezistor, případně přímo (c)



- Dnešním trendem je snižování příkonu a zvyšování rychlosti
- Napájecí napětí určuje řadu parametrů obvodů
  - Vyšší napájecí napětí = větší příkon a rychlost
  - Nižší napájení = menší odolnost proti rušení („noise margin“)
- Příklad technologií s různým napájecím napětím



- Příklad propojení TTL a CMOS obvodů
  - Je třeba dodržet logické úrovně

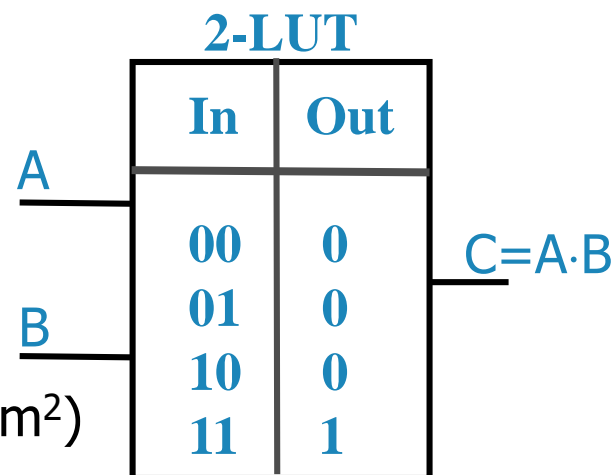


- Pouze pro čtení (Read-Only Memory - ROM)
  - Programované maskou ve výrobě (Mask-Programmable ROM)
  - Jednorázově programovatelné uživatelem (Programmable ROM)
- Pro omezený počet zápisů a neomezené čtení
  - Elektricky mazatelné a programovatelné
  - Existuje řada verzí - EPROM, EEPROM, FLASH (NAND, NOR)...
- Pro neomezený zápis a čtení (Read-Write Memory RWM)
  - Se sekvenčním přístupem
    - Zásobník (Last-In First-Out - LIFO)
    - Fronta (First-In First-Out - FIFO)
  - S náhodným přístupem (Random Access Memory – RAM)
    - Statická RAM (Static RAM - SRAM) s asynchronním či synchronním přístupem
    - Dynamická RAM (Dynamic RAM -DRAM) s asynchronním či synchronním (SDR, DDR...) přístupem

- Paměti lze použít jako generátory logických funkcí
  - Tzv. vyhledávací tabulka (Look-Up Table - LUT) implementuje libovolnou log. funkci  $N$  proměnných, kde  $N$  je počet adresovacích vstupů paměti

- Příklad

- Funkce AND

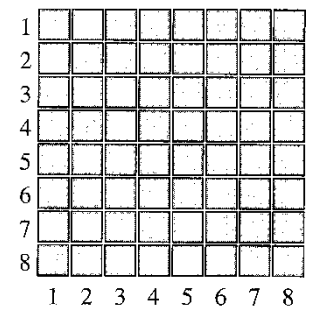


- Vlastnosti

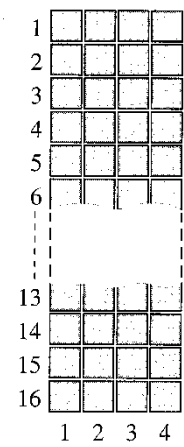
- Plocha - hustota integrace (počet bitů na  $\mu\text{m}^2$ )
  - Kapacita (počet bitů, organizace)
  - Doba přístupu (access time) - doba čtení a zápisu
  - Doba cyklu – (cycle time) – doba mezi jednotlivými přístupy do paměti
  - Propustnost – počet čtení/zápisů za sekundu (závisí na době cyklu)
  - Příkon, atd.



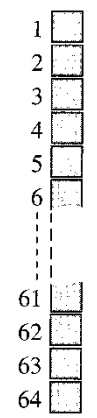
- Paměť o kapacitě 64 bitů
  - Matice (pole) 8x8 (a)
  - Matice (pole) 6x4 (b)
  - Matice (pole) 64x1 (c)
- Adresa
  - Umístění jednotky dat v paměťové matici



(a) 8 × 8 array



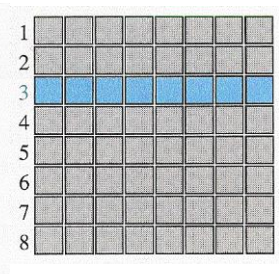
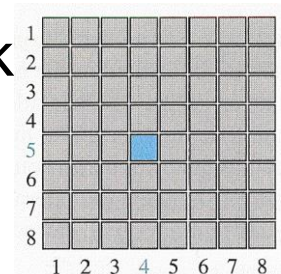
(b) 16 × 4 array



(c) 64 × 1 array

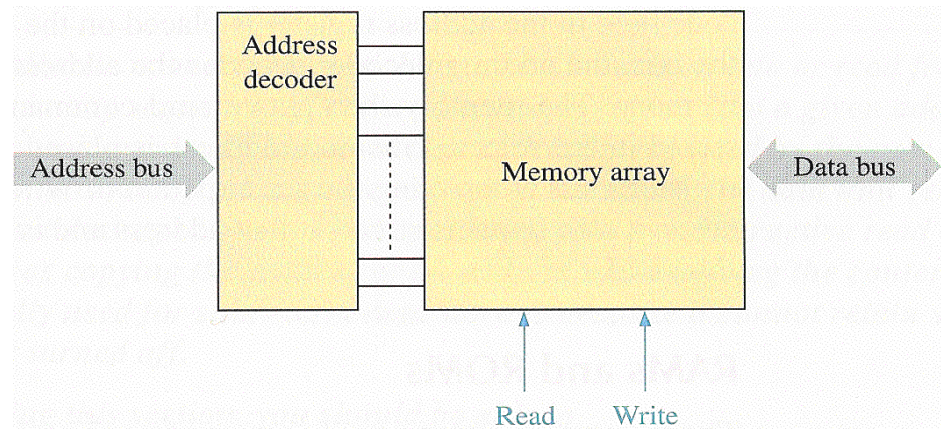
- Příklad adresování paměťových buněk

- Adresa bitu - řádek 5, sloupec 4
- Adresa řádku 5

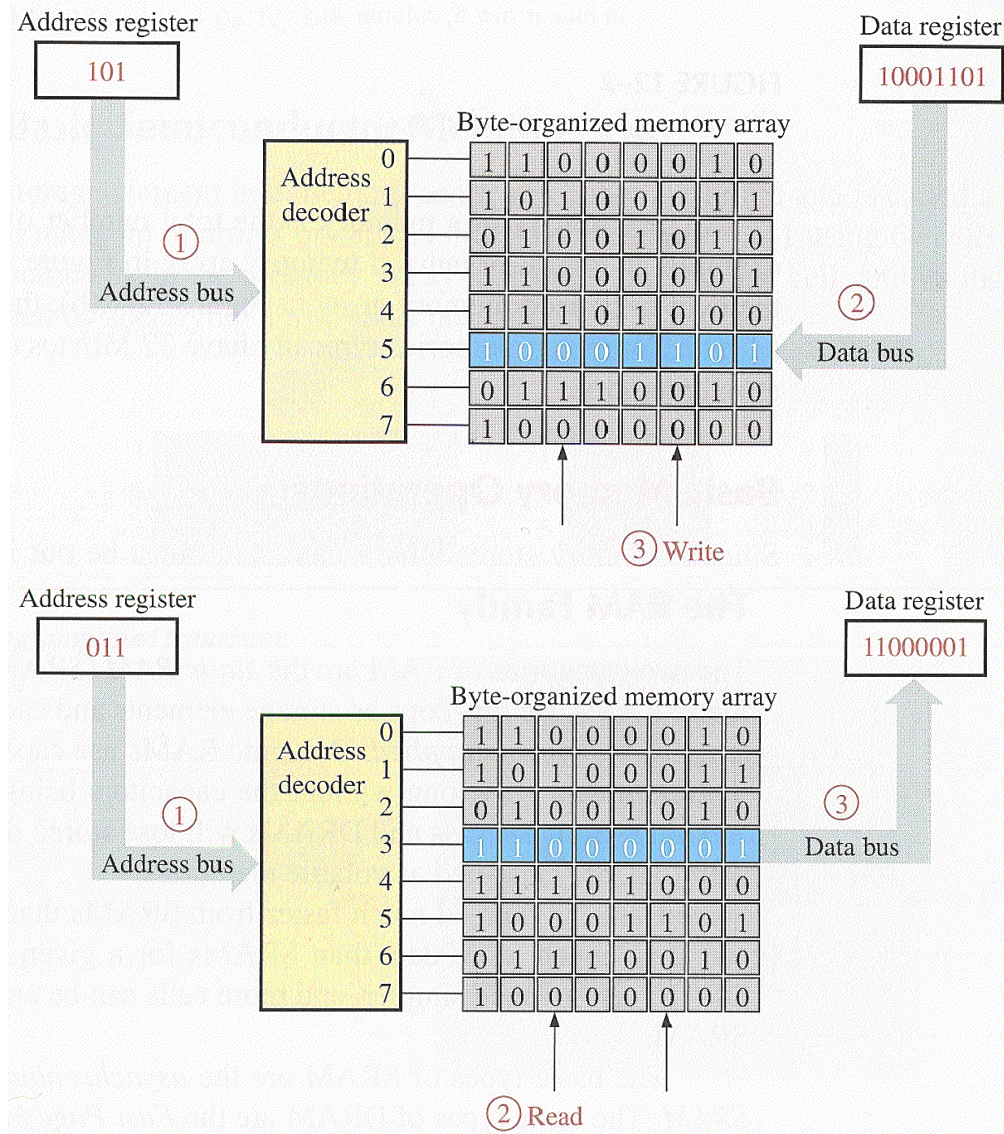


- Organizace paměti

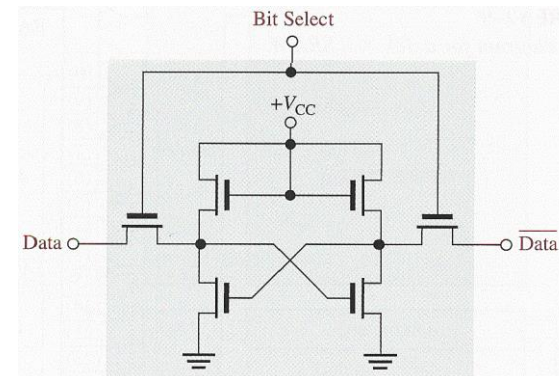
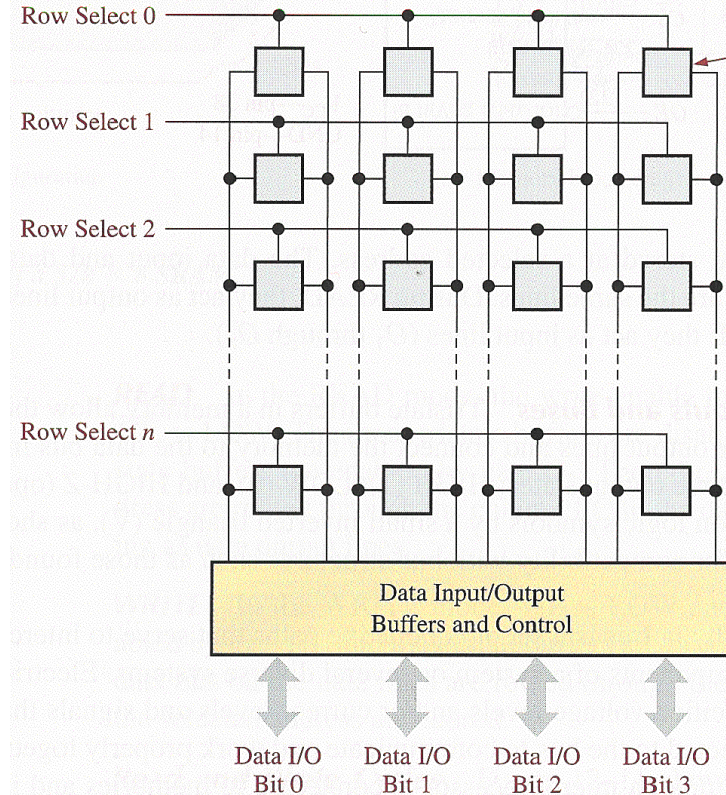
- Adresová a datová sběrnice
- Řídící signály pro zápis a čtení
- Paměťová matice



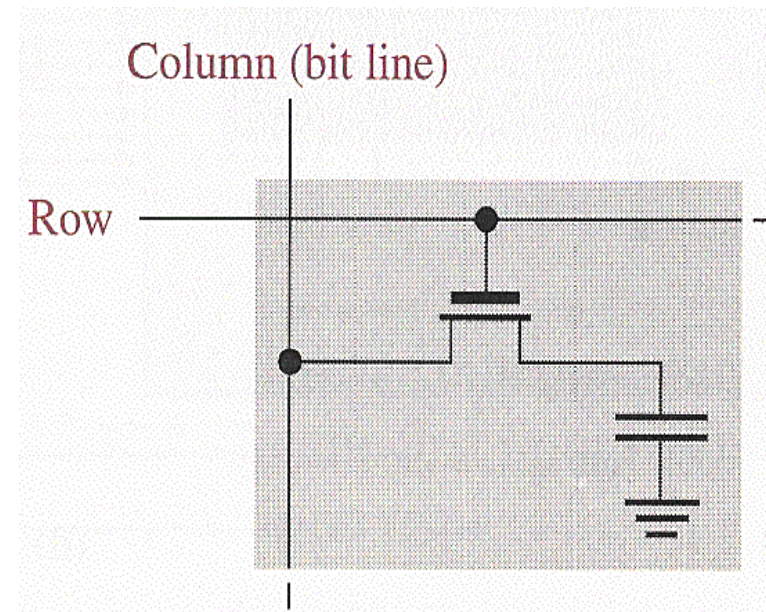
- Zápis dat do paměti
  - Adresový registr si pamatuje adresu, na kterou se má psát
  - V datovém registru jsou uložena data, která se mají zapsat do paměti
  - Signálem WRITE (zapiš) se data uloží do paměti
- Čtení dat z paměti
  - Adresový registr si pamatuje adresu, na kterou se má psát
  - Signálem READ (čti) se data přepíší z paměti do datového registru



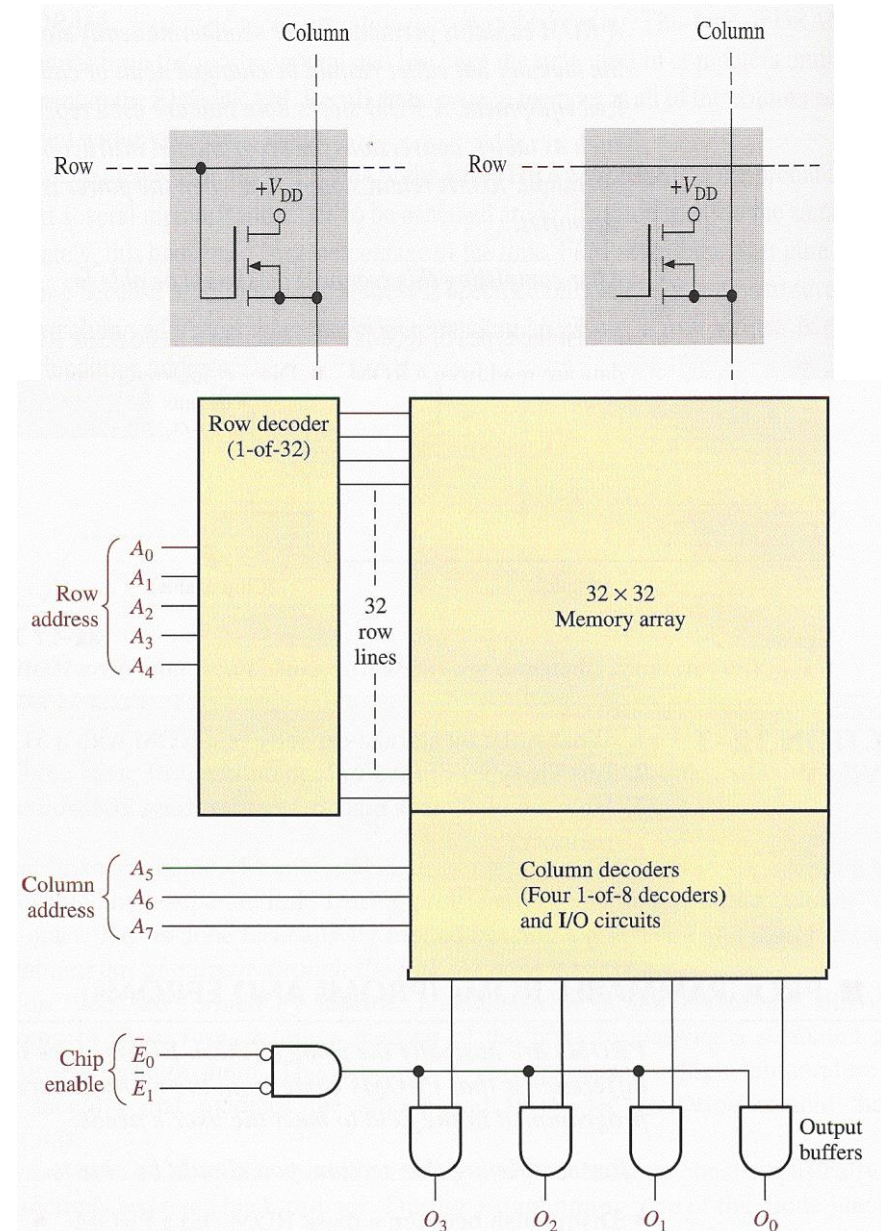
- Statická paměť pro zápis i čtení s náhodným přístupem
- Implementace paměťových buněk
  - V podstatě se jedná o hladinový RS klopný obvod sestavený ze dvou invertorů, který se překlápí na základě komplementárních signálů DATA při aktivním výběrovém signálu BIT SELECT
- Rychlé, složité, drahé, malé kapacity, velký příkon



- Dynamická paměť pro zápis i čtení s náhodným přístupem
- Implementace paměťových buněk
  - Hodnota se uchovává na parazitní kapacitě jako náboj
  - Kondenzátor se časem vybíjí vlivem parazitních svodů
  - Hodnota náboje se musí tzv. osvěžovat (anglicky „refreshing“)
- Velké kapacity, jednoduché, levné, pomalé

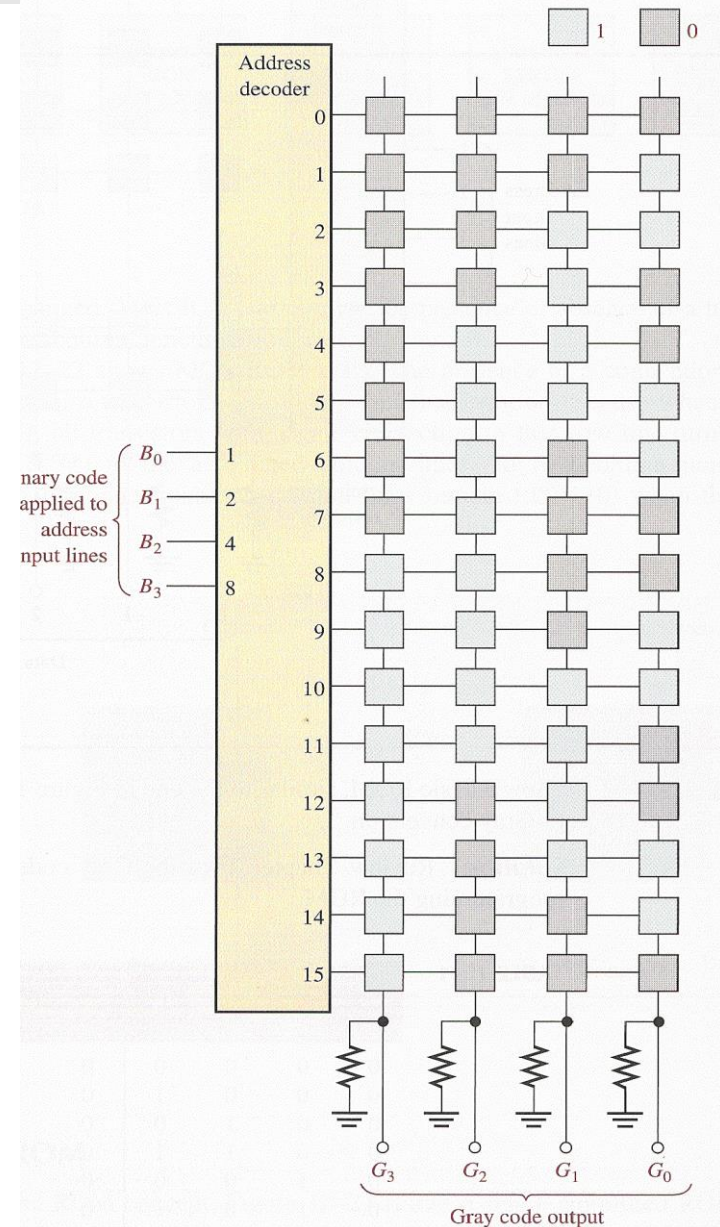


- Příklad implementace paměťových buněk
  - Tranzistorem MOS se zapojenou (log.1)/nezapojenou (log.0) řídicí elektrodou
  - Spoj k elektrodě se např. „přepálí“ při programování ve výrobě
- Příklad organizace paměti ROM
  - 32x32 buněk
  - Dekodér adresy sloupce (Column)
  - Dekodér adresy řádku (Row)
  - Chip Select – povolení funkce

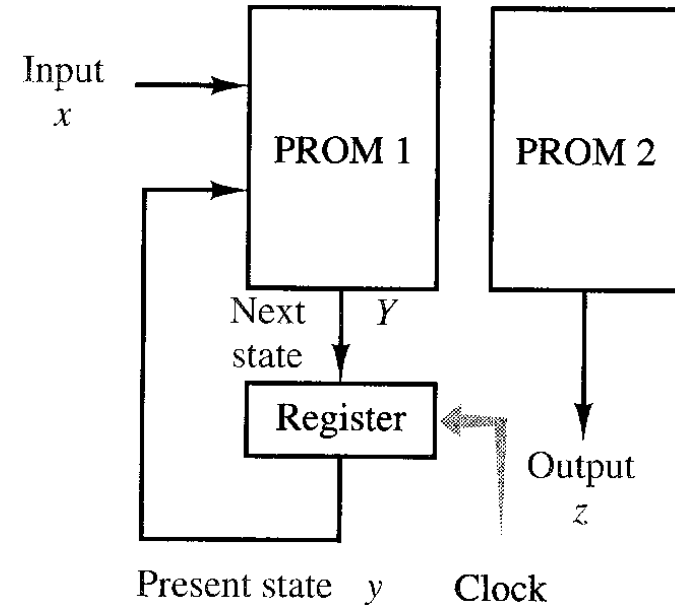


- Příklad použití ROM paměti
  - Převodník binárního kódu na Grayův kód
  - Pravdivostní tabulka je implementována v paměťové matici
  - Adresa – vstupní proměnné
  - Data - výstupy

Binary				Gray			
$B_3$	$B_2$	$B_1$	$B_0$	$G_3$	$G_2$	$G_1$	$G_0$
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0



- Implementace kombinační logické sítě
  - PROM1 – přechodová funkce
  - PROM2 – výstupní funkce
    - Obsah obou pamětí lze programovat, a tím též měnit činnost automatu
    - Zapojení obvodu zůstává stejné (výhoda)
    - Obě paměti lze sloučit do jedné
  - Registr – paměť stavu



## • Popis

- Vstupy  $X$  jsou přivedeny na část adresových vodičů paměti PROM1
- Na datových vodičích paměti PROM1, které jsou přivedeny na registr stavu, se čte kód následujícího stavu (next state)
- Výstup registru stavu (present state - současný stav) je přiveden na zbytek adresových vodičů paměti PROM1
- Adresové vodiče PROM2 jsou přivedeny
  - Pro Mooreův automat na výstup registru
  - Pro Mealyho automat na výstup registru a vstupy  $X$
- Na datových výstupech PROM2 se čtou výstupy  $Z$

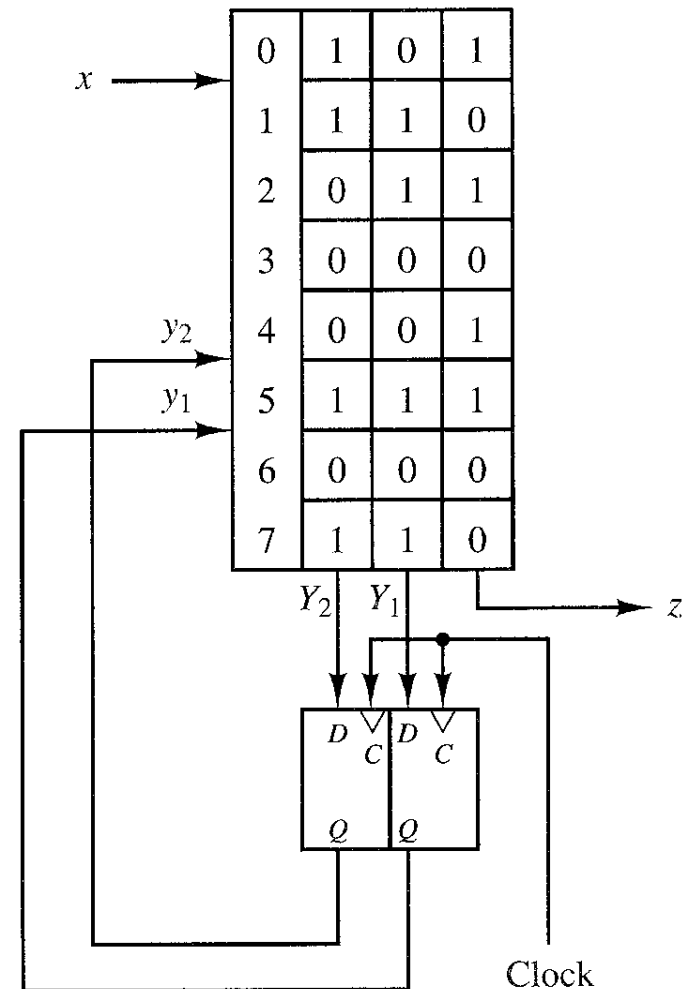
- Popis

- Tabulka přechodů

	$x$	
	0	1
$y_2y_1$		
00	10/1	00/1
01	11/0	11/1
10	01/1	00/0
11	00/0	11/0
	$Y_2Y_1/z$	

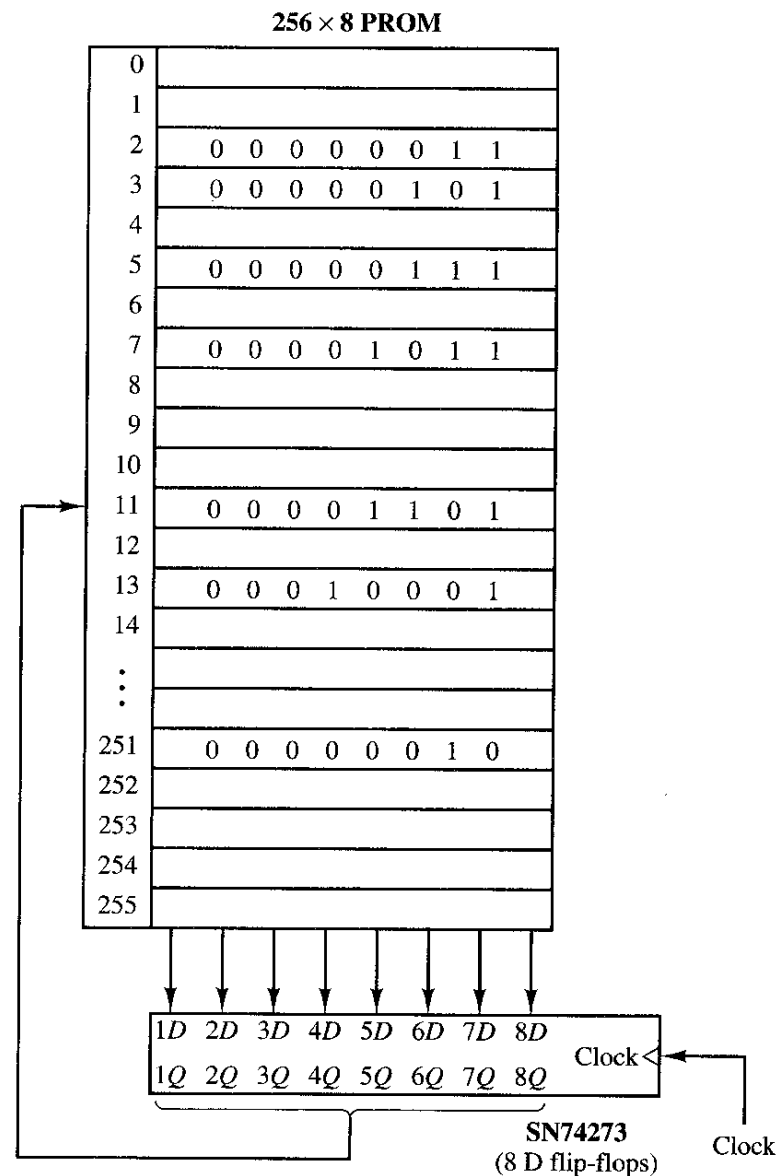
- Přechodová a výstupní funkce = obsah paměti PROM

$x$	$y_2$	$y_1$	$Y_2$	$Y_1$	$z$
0	0	0	1	0	1
0	0	1	1	1	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	0

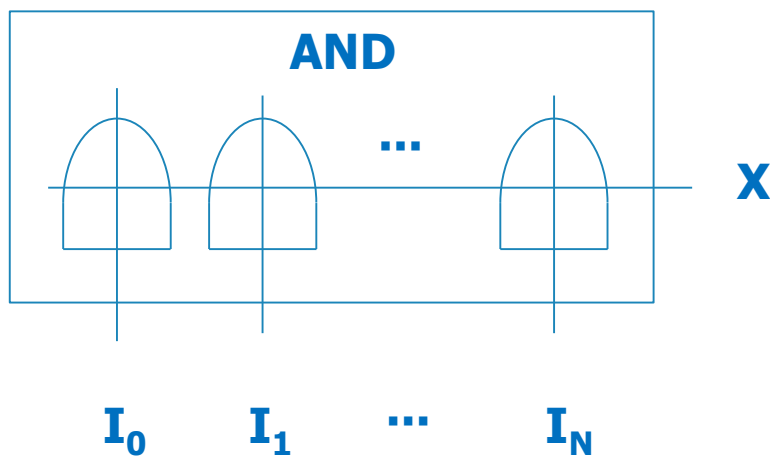




- Generátor posloupnosti
  - Na adresách odpovídajícím jednotlivým prvočísly jsou uloženy adresy následujících prvočísel
  - Automat startuje od adresy 2
  - S každým taktem hodin se na výstupu registru generuje rostoucí posloupnost prvočísel 2,3,5,7,...251,2,... (8 bitů)
  - Pouhou změnou obsahu paměti lze vytvořit (přeprogramovat) automat pro generování libovolné posloupnosti s max. počtem  $2^n$  prvků, kde  $n$  je počet bitů paměti a registru



- Programovatelný logický obvod
  - Anglicky Programmable Logic Device (PLD)
  - ROM, PLA, PAL, GAL, CPLD, FPGA
  - Mají předdefinovanou strukturu, kterou lze různým způsobem programovat pro realizaci log. obvodů
  - PLD mohou též obsahovat klopné obvody, třístavové budiče, paměti atd.
- Programování lze provést např.
  - Přepálením „pojistky“ – PAL, PLA
  - Naprogramováním paměťových buněk PROM – GAL, CPLD
  - Naprogramováním paměti SRAM – FPGA
  - Provádí se pomocí programátoru či přímo v zařízení pomocí specializovaného rozhraní
- Výhody
  - Uživatel si může realizovat složité log. obvody sám
  - Obvody lze modifikovat
  - Dnešní kapacity PLD obvodů jsou velmi velké (miliony ekvivalentních členů NAND)

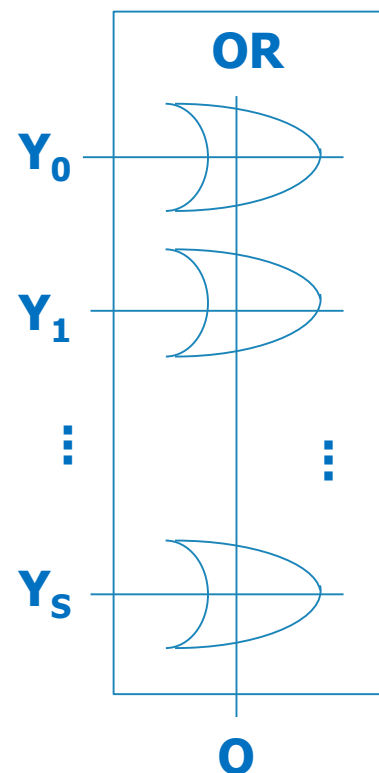


## AND pole

Log. 0 na kterémkoliv ze vstupů  $I_0..I_N$ , generuje log. 0 na výstupu (montážní součin)

- Programování

- Se provádí přepálením (odpojením) spoje (pojistky) mezi voliči v rámci pole



## OR pole

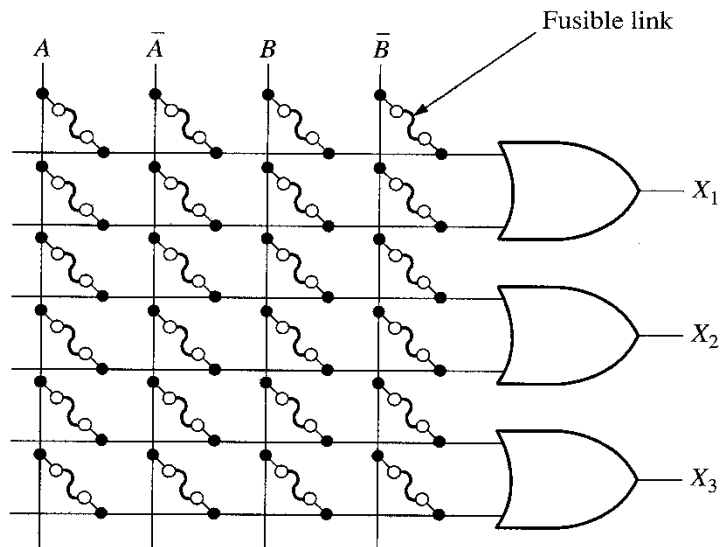
Log. 1 na kterémkoliv ze vstupů  $Y_0..Y_S$ , generuje log. 1 na výstupu O (montážní součet)

- Popis

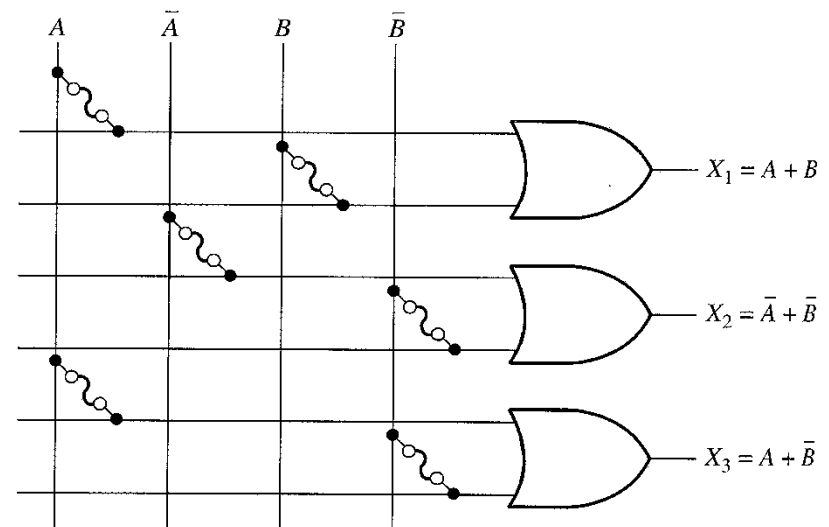
- Pole hradel OR s programovatelným propojením na vstupní proměnné v přímé i negované podobě (invertor není zakreslen)
- Realizuje konjunktvní formu

- Příklad

- Nenaprogramované pole OR (a)
- Naprogramované pole OR (b)

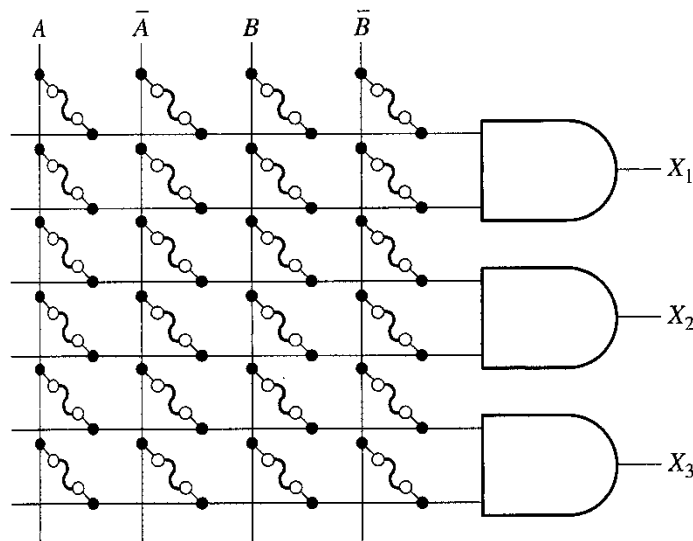


(a) Unprogrammed

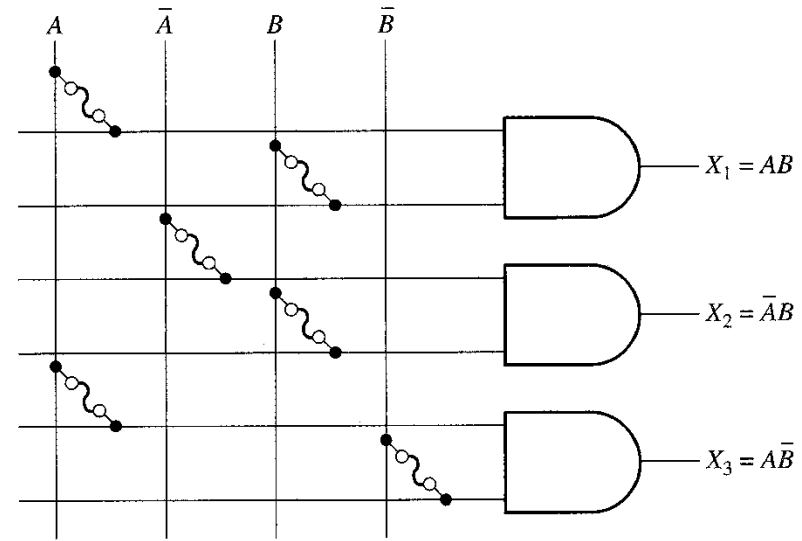


(b) Programmed

- Popis
  - Pole hradel AND s programovatelným propojením na vstupní proměnné v přímé i negované podobě (invertor není zakreslen)
  - Realizuje disjunktční formu
- Příklad
  - Nenaprogramované pole AND (a)
  - Naprogramované pole AND (b)

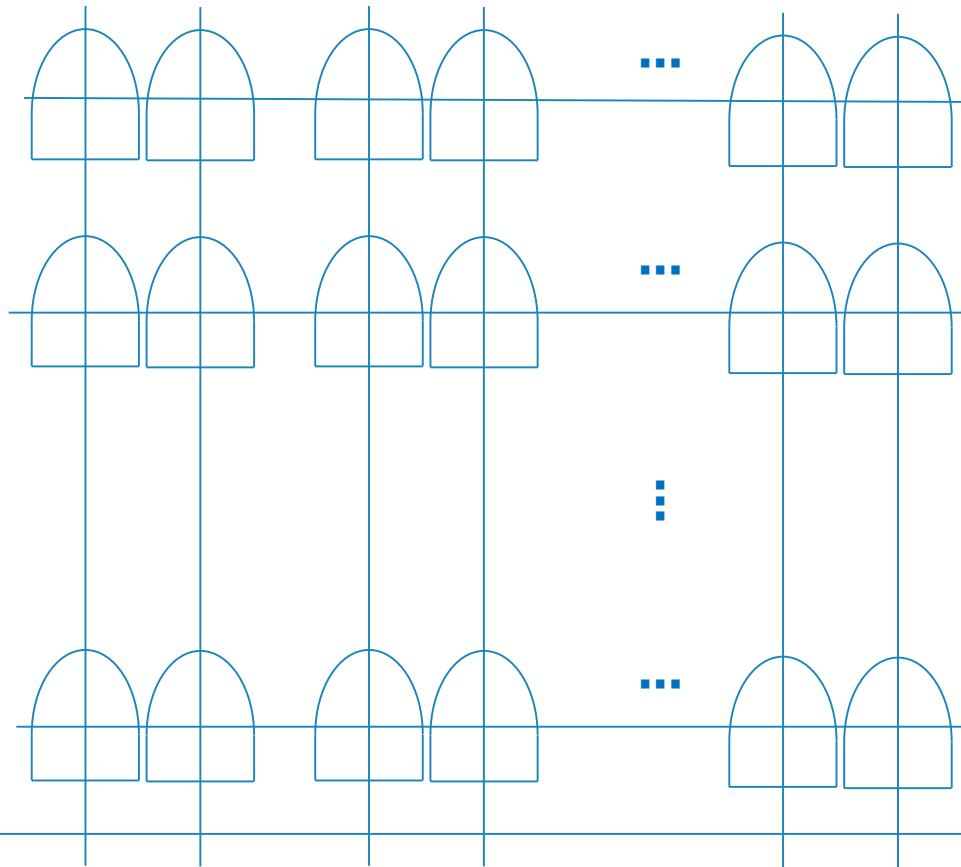


(a) Unprogrammed



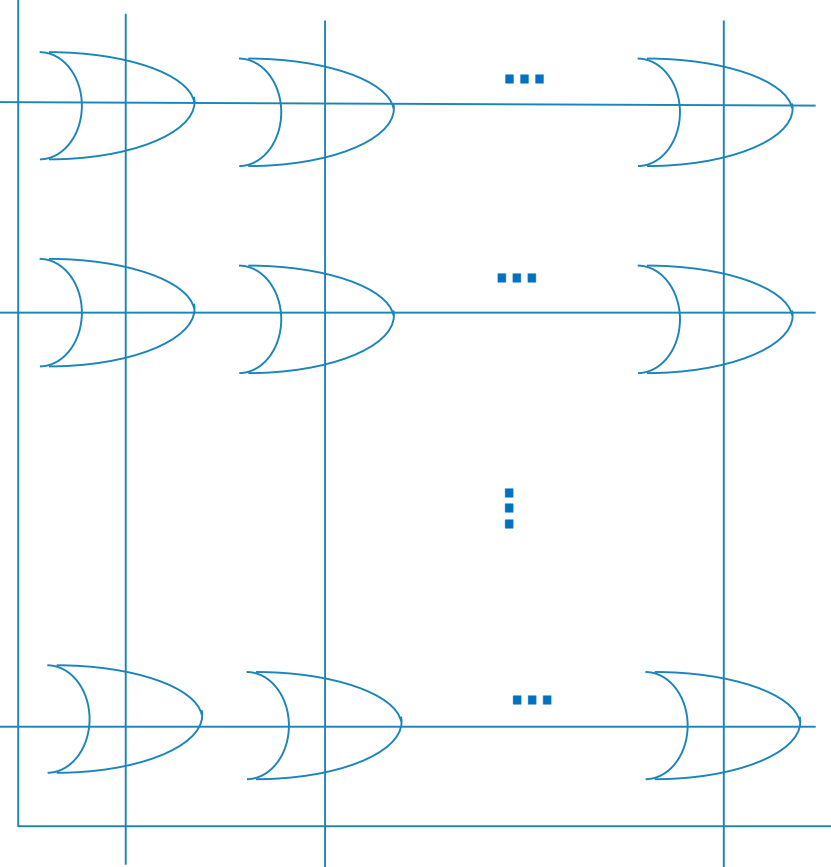
(b) Programmed

## Pole AND



$I_0$   $\text{not}(I_0)$   $I_1$   $\text{not}(I_1)$  ...  $I_N$   $\text{not}(I_N)$

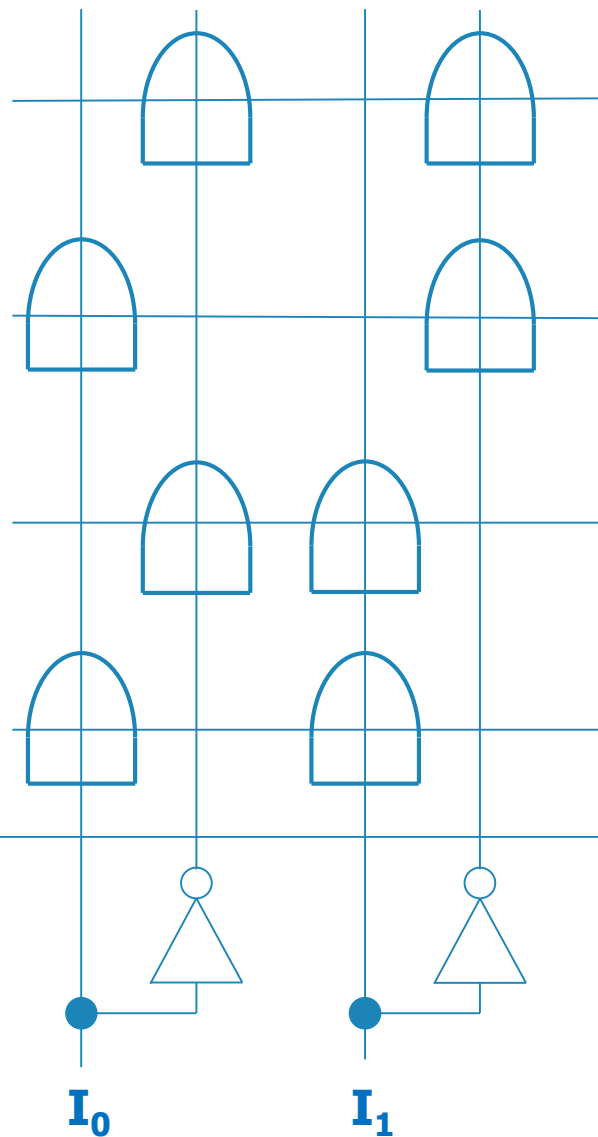
## Pole OR



$O_0$   $O_1$  ...  $O_M$

Duálně platí, že pole OR-AND implementuje součin sum (Úplná normální konjunktivní forma, POS)

## Fixní pole AND - dekodér



$$\text{not}(I_0) \cdot \text{not}(I_1)$$

$$I_0 \cdot \text{not}(I_1)$$

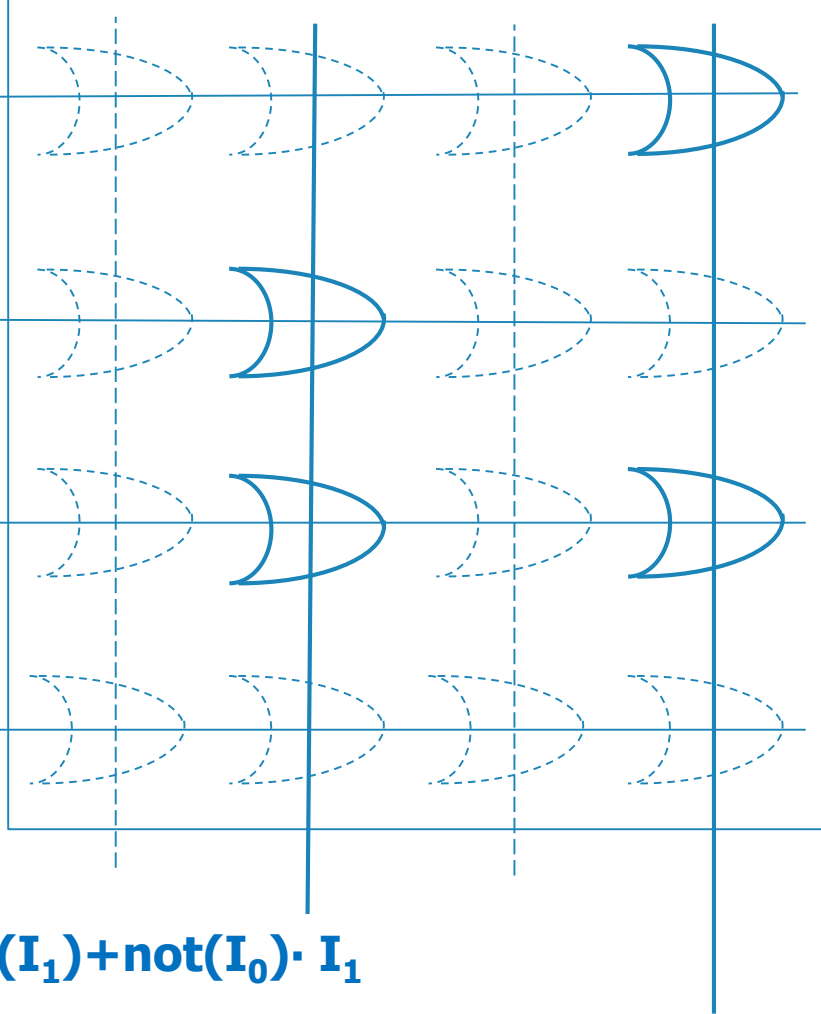
$$\text{not}(I_0) \cdot I_1$$

$$I_0 \cdot I_1$$

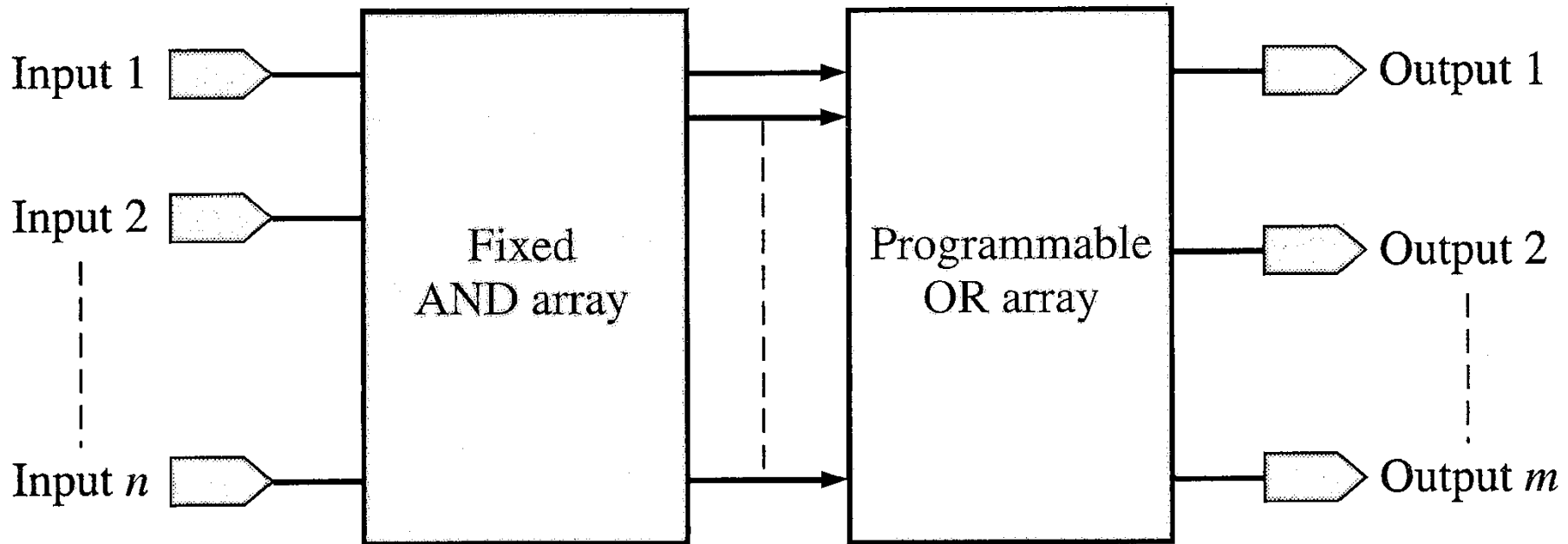
$$O_1 = I_0 \cdot \text{not}(I_1) + \text{not}(I_0) \cdot I_1$$

$$O_3 = \text{not}(I_0) \cdot \text{not}(I_1) + \text{not}(I_0) \cdot I_1$$

## Programovatelné pole OR

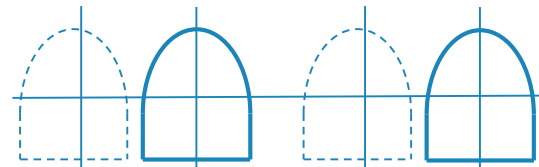


- Použití paměti ROM (PROM) pro realizaci log. funkcí
  - AND pole – dekodér adres
  - OR pole – data uložena v paměti
  - Díky fixnímu AND poli je využití omezeno, neboť pro funkce s více vstupy je třeba velkých kapacit paměti, což je drahé

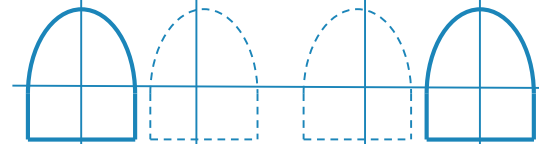




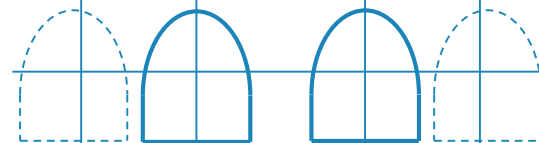
## Programovatelné pole AND



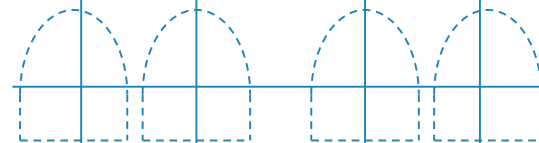
$$\text{not}(I_0) \cdot \text{not}(I_1)$$



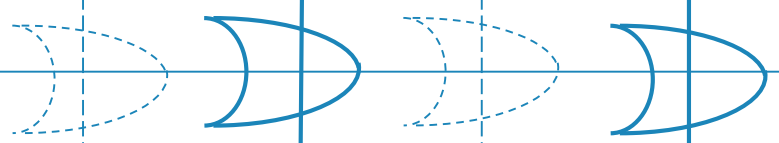
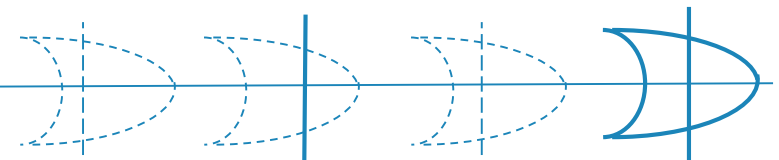
$$I_0 \cdot \text{not}(I_1)$$



$$\text{not}(I_0) \cdot I_1$$



## Programovatelné pole OR



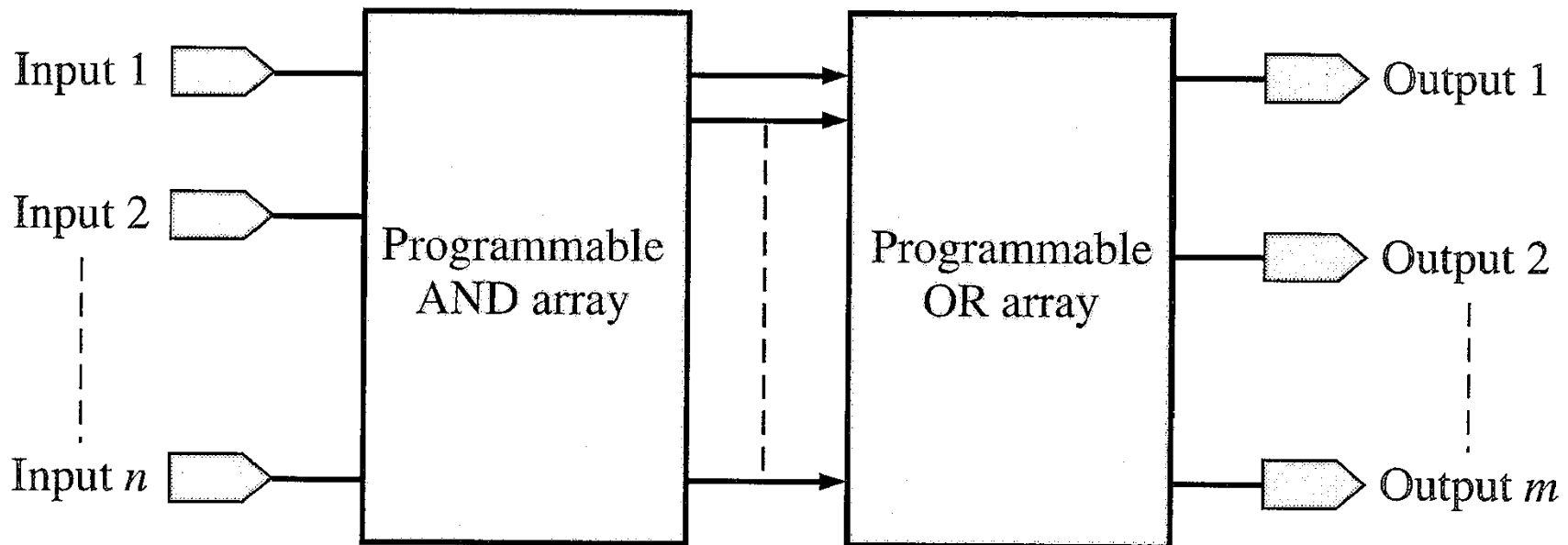
$$O_1 = I_0 \cdot \text{not}(I_1) + \text{not}(I_0) \cdot I_1$$

$$O_3 = \text{not}(I_0) \cdot \text{not}(I_1) + \text{not}(I_0) \cdot I_1$$

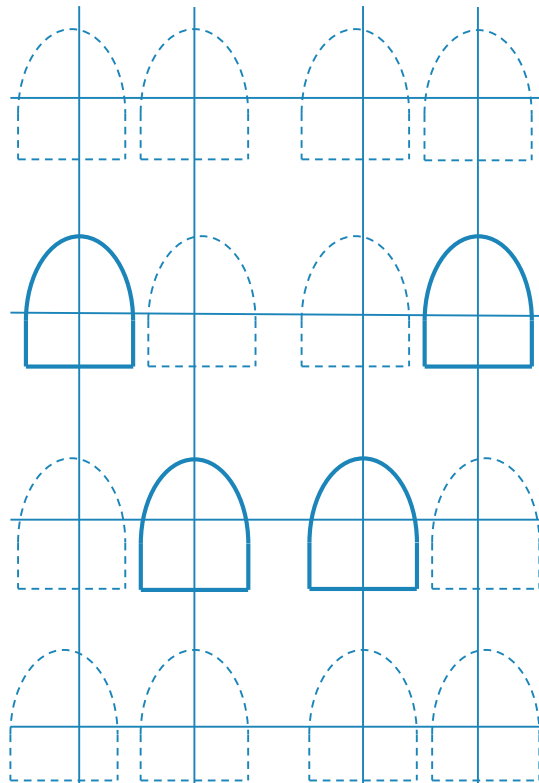
$I_0$

$I_1$

- Programmable Logic Array (PLA)
  - Obě pole jsou programovatelná
  - Výhodou je skutečnost, že jsou plně programovatelná
  - Nevýhodou je větší zpoždění kvůli programovacím propojkám („pojistkám“) v obou polích, které mají větší zpoždění než vodiče a log. členy díky přechodovému odporu a parazitním kapacitám
  - Pomalé, drahé



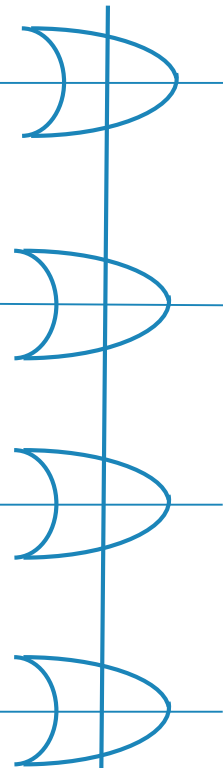
## Programovatelné pole AND



$$I_0 \cdot \text{not}(I_1)$$

$$\text{not}(I_0) \cdot I_1$$

## Fixní pole OR

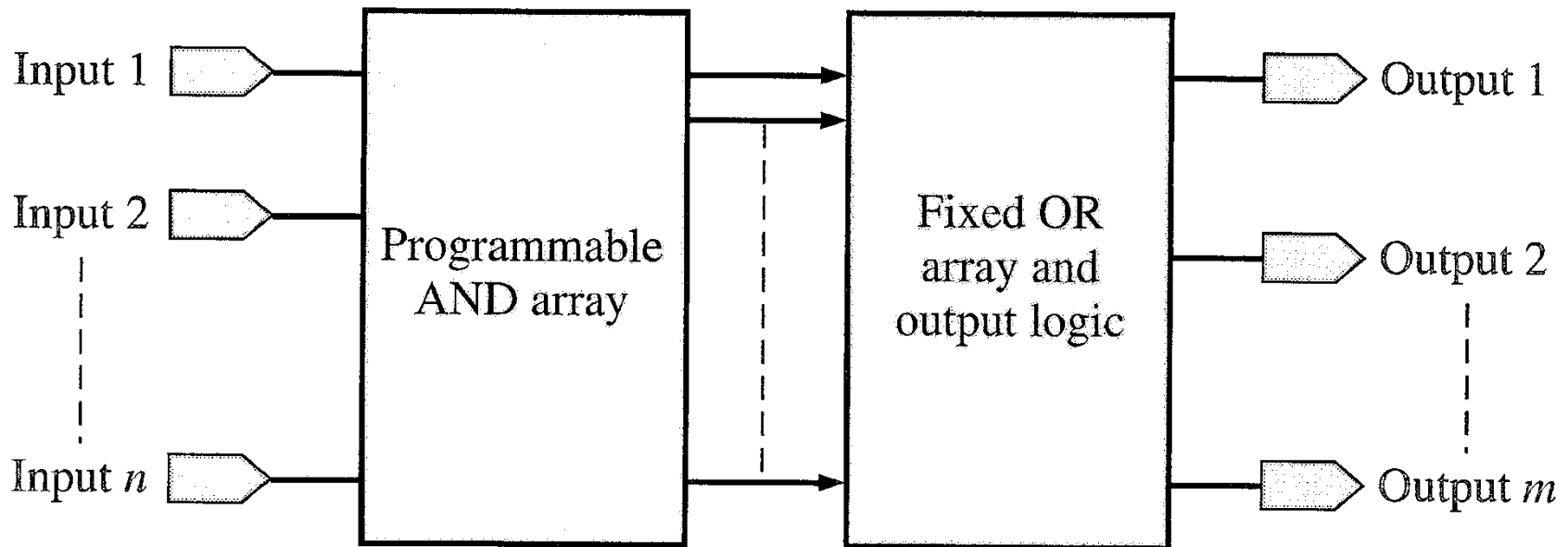


$$O = I_0 \cdot \text{not}(I_1) + \text{not}(I_0) \cdot I_1$$

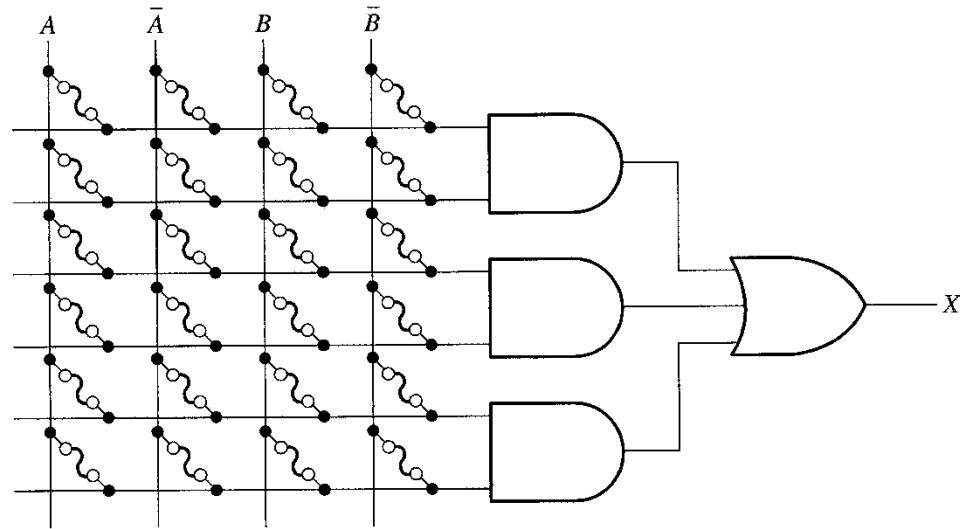
$I_0$

$I_1$

- Programmable Array Logic (PAL)
  - Pouze pole AND je programovatelné
  - Nemají omezení paměti PROM (díky programovatelnému poli AND)
  - Jsou rychlejší díky fixnímu poli OR bez propojek
  - Jsou levnější než PLA
  - Realizuje disjunktvní formu

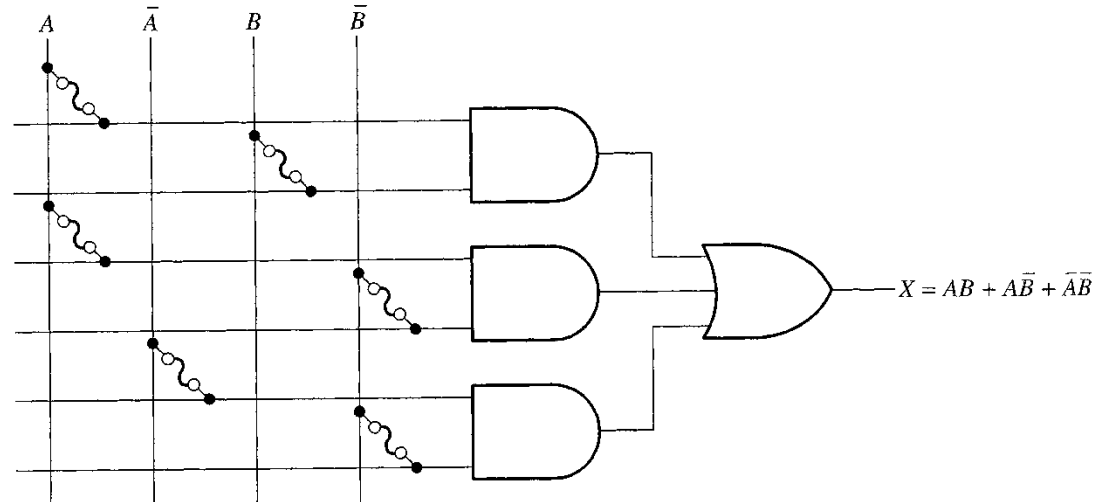


- Struktura obvodu PAL před naprogramováním



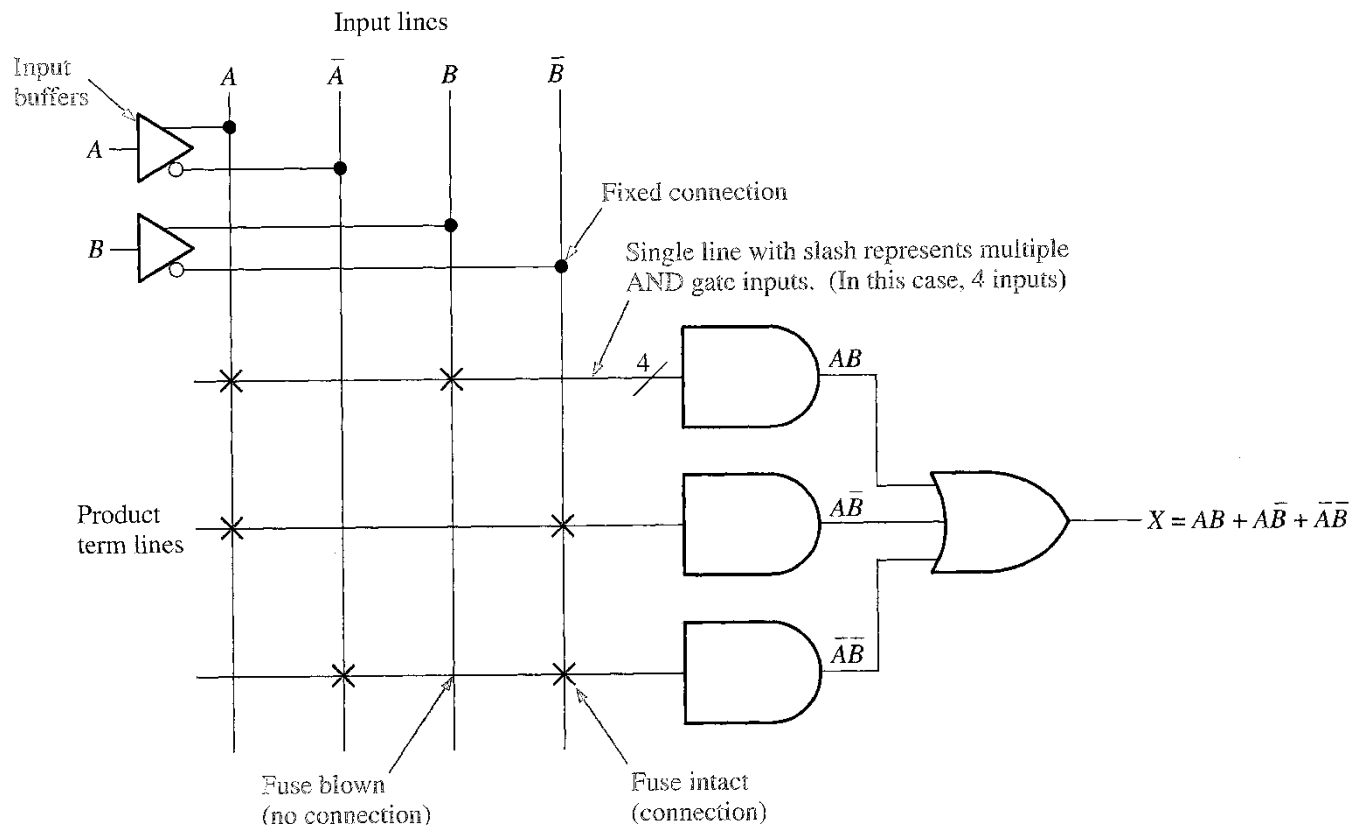
- Příklad implementace log. funkce

- Propojky zůstanou v místech, která odpovídají proměnným ve výrazu

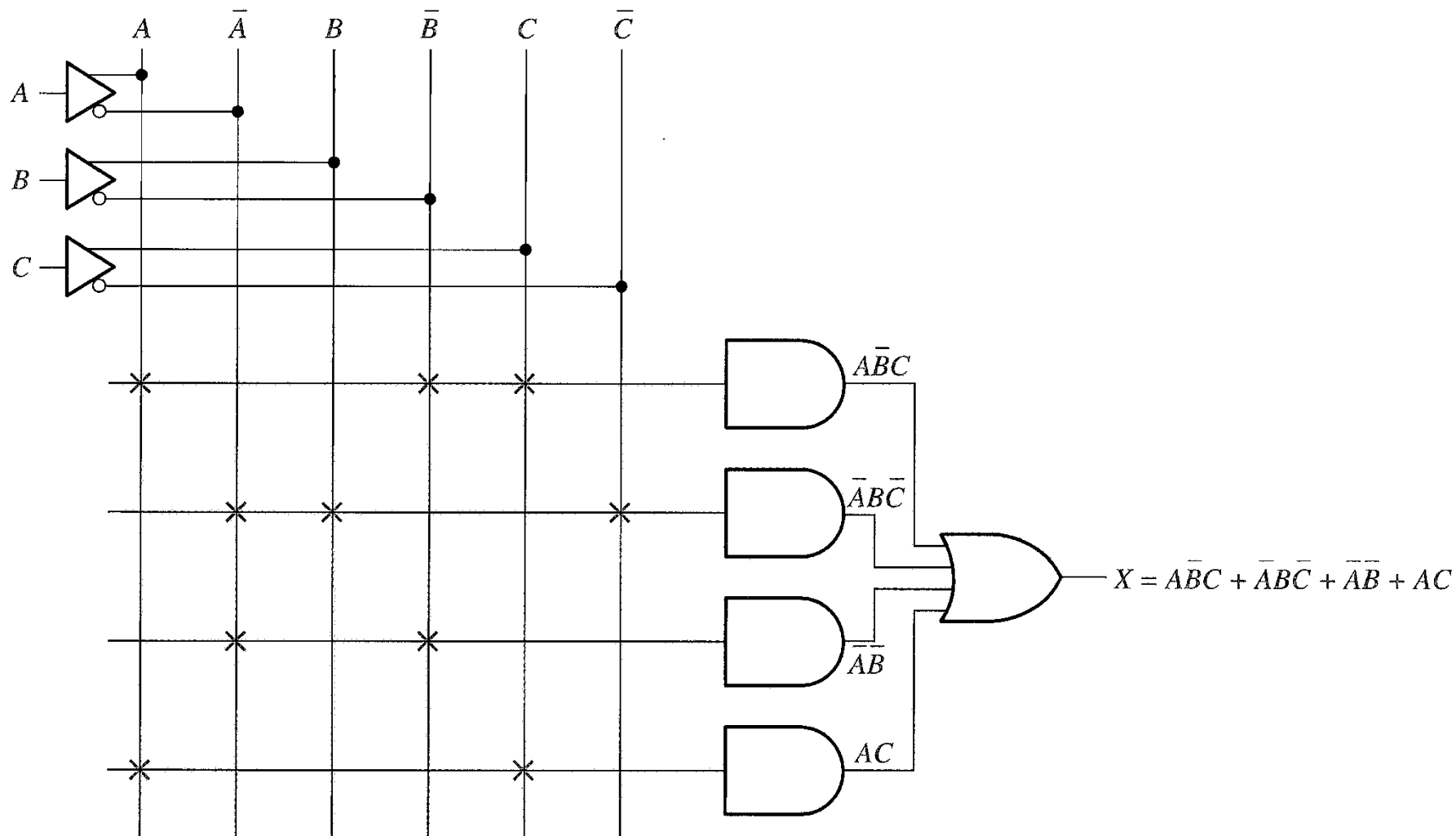


- Vnitřní struktura

- Obsahuje též invertory pro každou vstupní proměnnou
- Pro naše účely budeme místa, ve kterých propojky zůstanou, značit křížkem

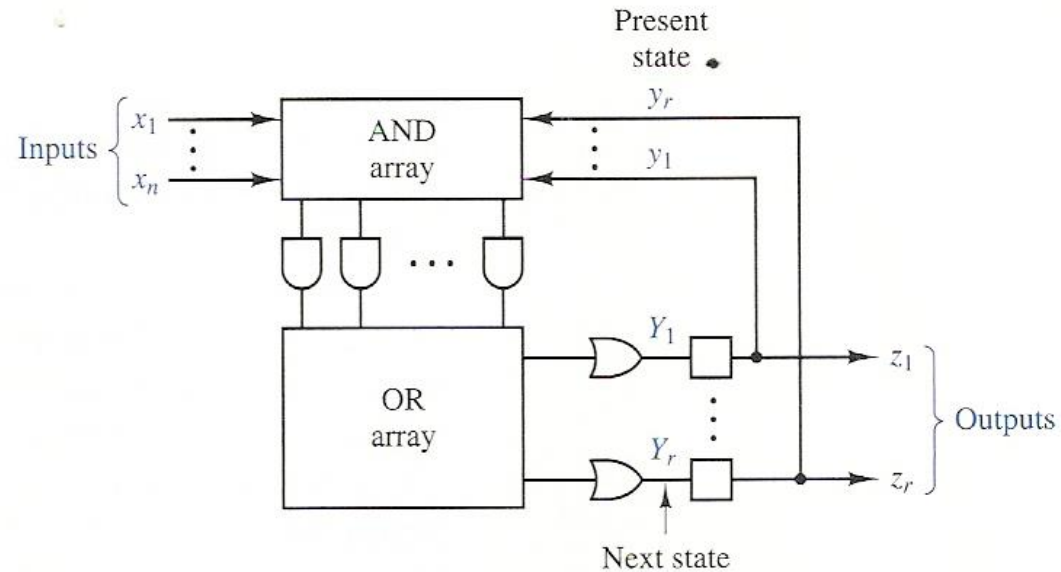
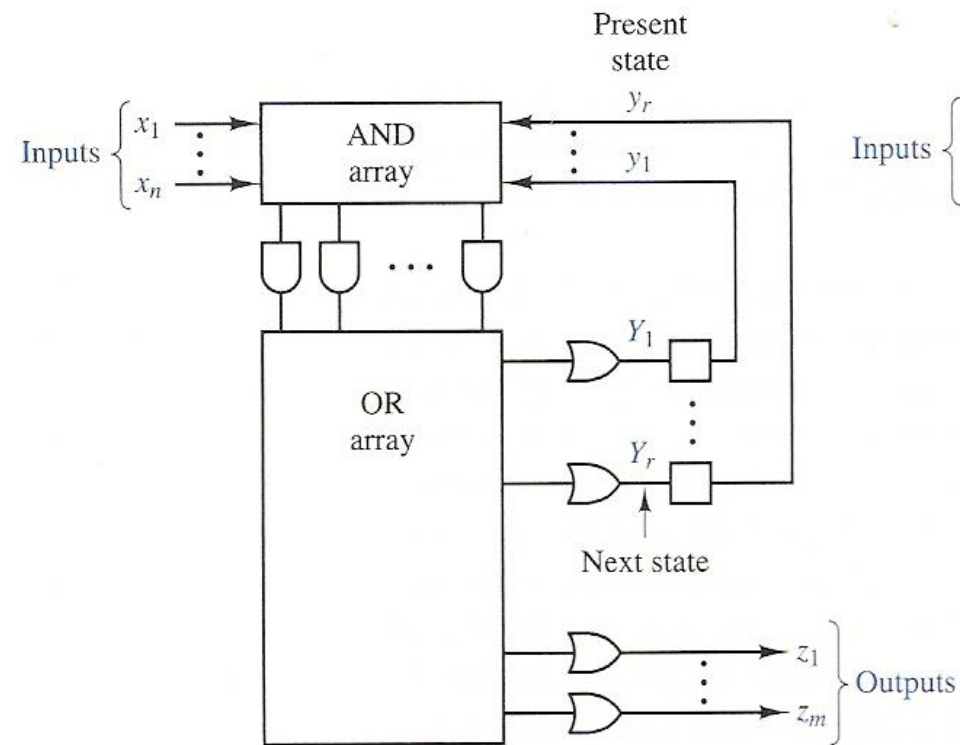


- Příklad implementace funkce tří proměnných pomocí PAL



- PLD struktury bývají doplněny o registry
- Příklad Mealyho automatu

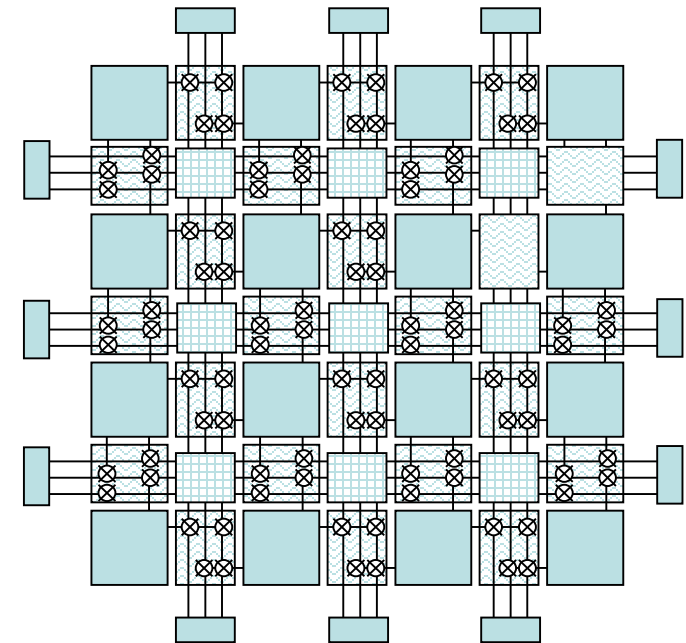
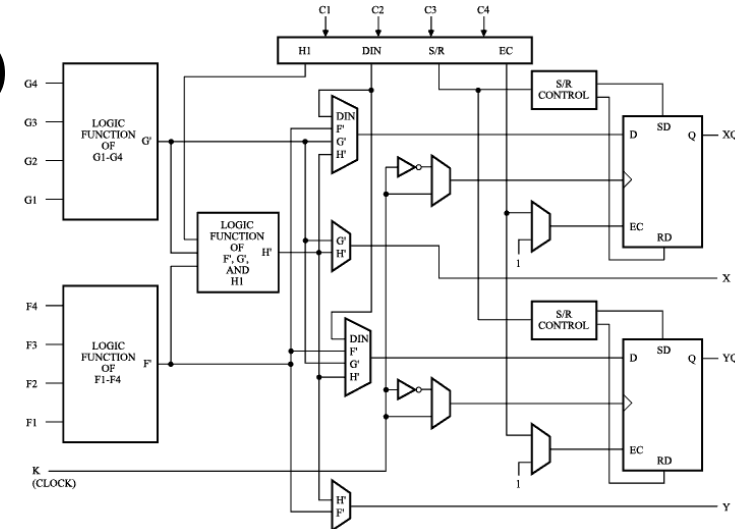
- Příklad Mooreova automatu





- PAL (Programmable Array Logic)
  - Programovatelné AND pole (volba mintermu normální disjunktvní formy se provádí pomocí destrukce propojek)
  - Pevné OR pole (logický součet vybraných mintermů), které jsou často doplněny o klopné obvody
- GAL (Generic Array Logic)
  - Obdobná architektura jako PAL
  - Propojky jsou však elektricky programovatelné (řízené pomocí paměťových buněk např. typu EEPROM)
- CPLD (Complex Programmable Logic Devices)
  - Skládají se z AND-OR polí a registrů s elektricky (typicky pomocí paměťových buněk typu FLASH) programovatelnými křížovými přepínači

- Field Programmable Gate Array (FPGA)
  - Vynález 1969, výroba Altera (1983) a Xilinx (1984)
  - Matice konfigurovatelných logických bloků (malá SRAM pro implementaci log. funkcí, multiplexory, registry)
  - Konfigurovatelná propojovací síť
- Konfigurace je uložena v paměti SRAM
  - Po každém připojení napájecího napětí je třeba konfigurační informaci zkopírovat do FPGA z externí paměti
  - Možnost neomezeného „programování“
- Klíčové výhody
  - Výkonnost blízka HW
  - Flexibilita blízka SW



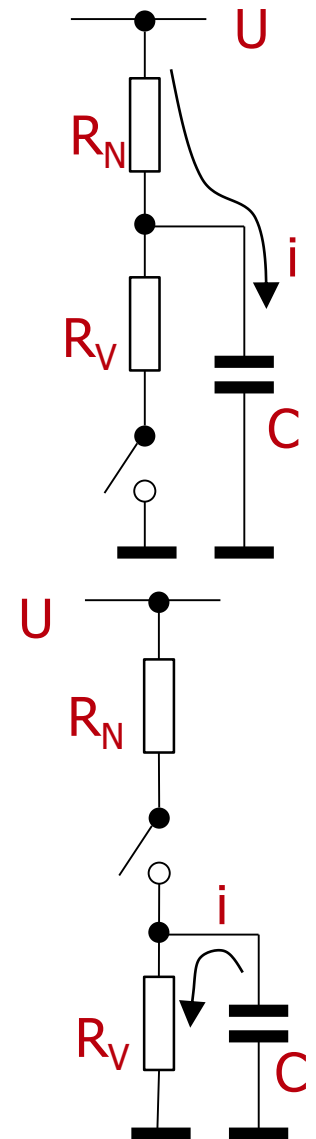
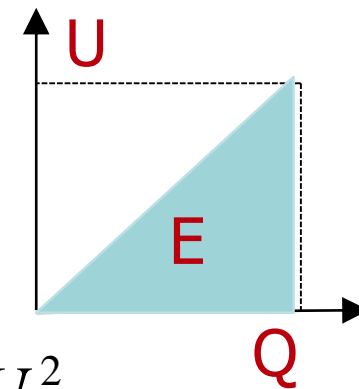
- Náboj  $Q = C \cdot U [C] = I \cdot t [As]$
- Energie  $E = 1/2 \cdot Q \cdot U = 1/2 \cdot C \cdot U^2 [J = Ws]$
- Nabíjení  $E_C = E_{RN} = 1/2 \cdot C \cdot U^2$
- Vybíjení  $E_C = E_{RV} = 1/2 \cdot C \cdot U^2$
- Energie nabití/vybití

$$E_{0 \rightarrow 1 \rightarrow 0} = E_{RN} + E_{RV} = C \cdot U^2$$

- Příkon při periodickém nabíjení/vybíjení

$$P[W] = \frac{E[Ws]}{t[s]} = E[Ws] \cdot f[Hz] = C \cdot U^2 \cdot f$$

- f... frekvence změn 0-1-0 (frekvence hodinového signálu)



- Celkový = statický + dynamický  $P = P_S + P_D$

- Statický

- Na vstupech hradla se nemění logická úroveň
- Dán svodovými proudy (leakage) ve struktuře hradel

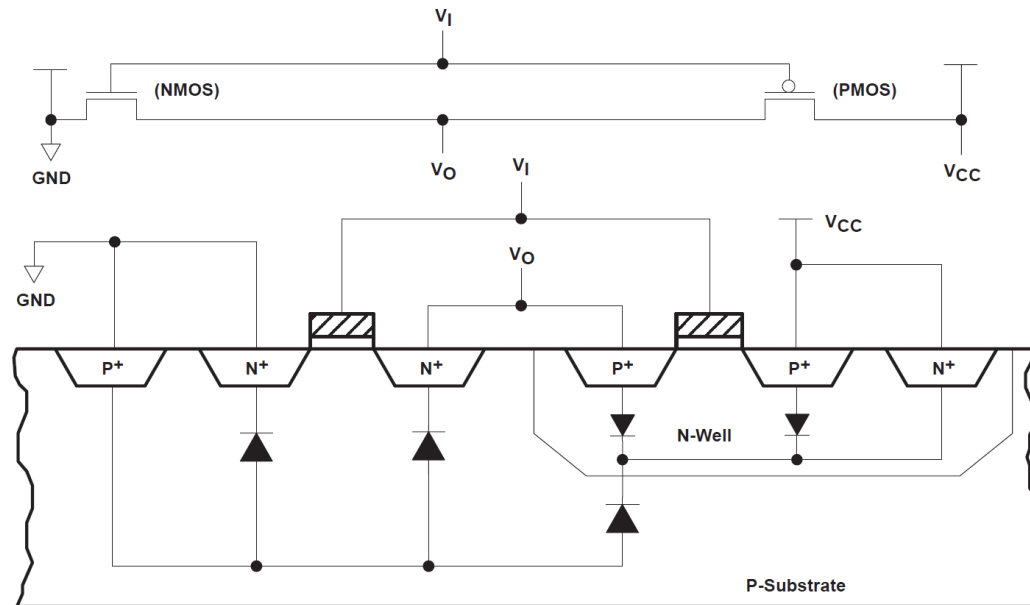
$$P_S = U \cdot \sum_{n=1}^N I_{Ln}$$

- $I_L$  roste s klesajícím  $V_T$
- Napájecí napětí se značí  $U$ ,  $V_{cc}$ ,  $V_{dd}$ ..

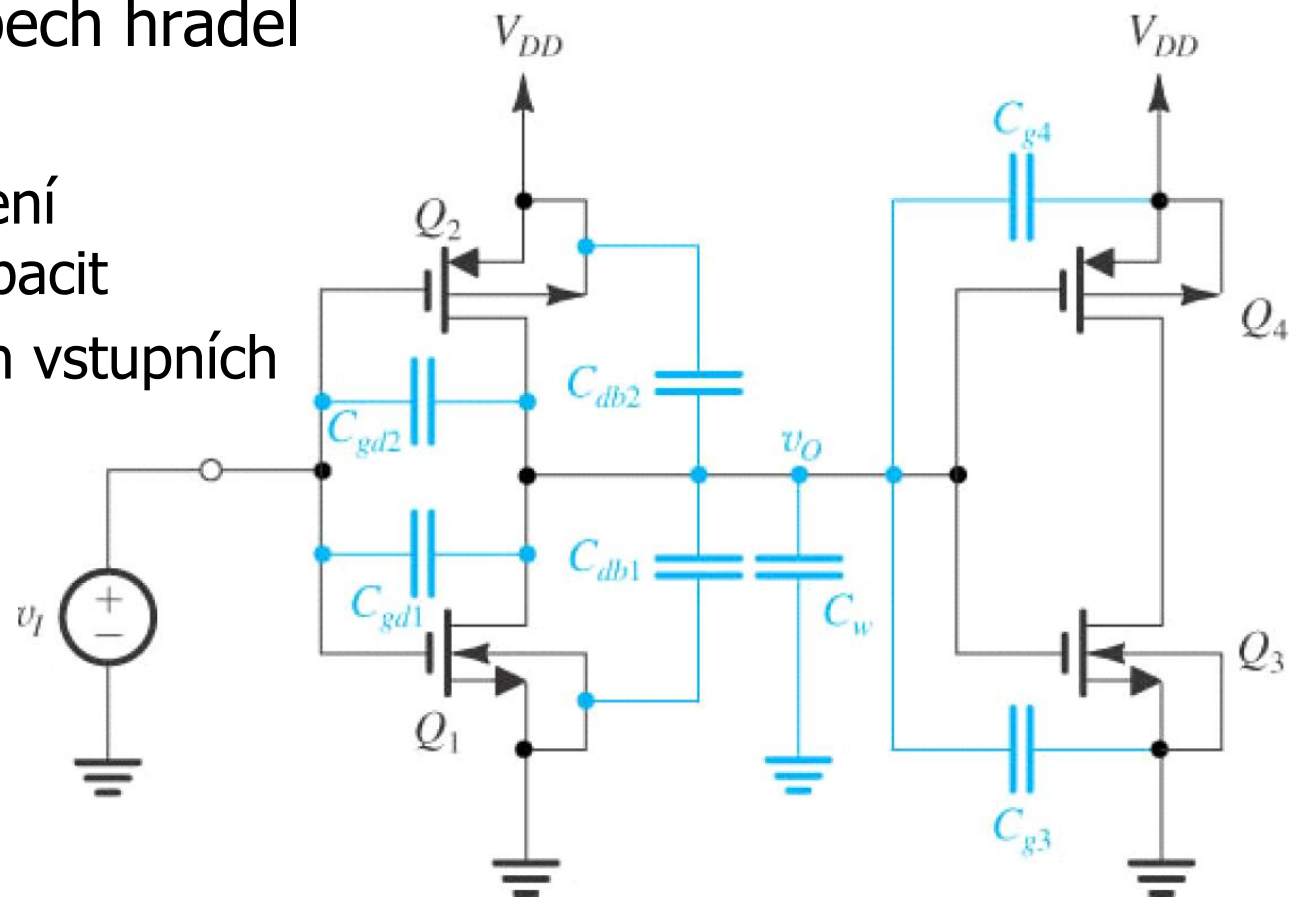
- Příklad – invertor

- Svody na parazitních diodách
- $i_s$  - saturační proud diody
- $V$  - napětí na diodě

$$I_L = i_s \left( e^{qV/kT} - 1 \right)$$



- Dynamický = přechodový + příkon zátěže  $P_D = P_T + P_L$
- Vzniká díky změnám logických úrovní na vstupech/výstupech hradel
- Závisí na
  - Nabíjení/vybíjení parazitních kapacit
  - Rychlosti změn vstupních log. úrovní
- Příklad – parazitní kapacity



- Přechodový (transient) = přepínací + zkratový

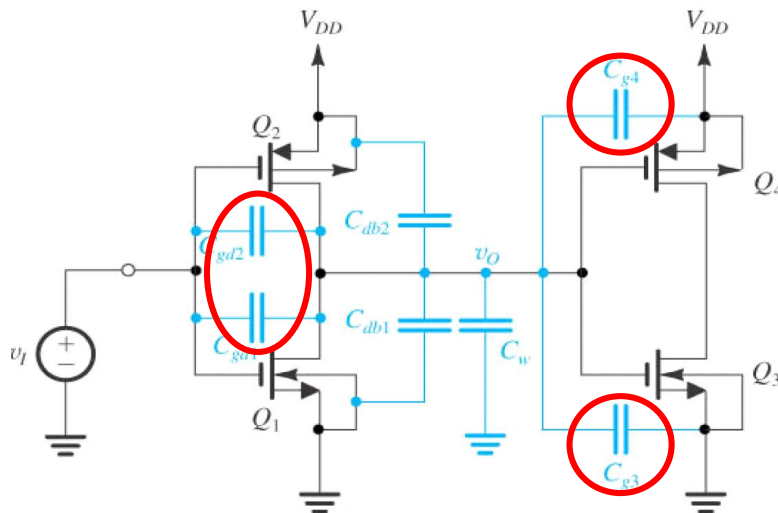
$$P_T = P_{SWI} + P_{THR} = V_{DD}^2 \cdot \sum_{n=1}^N (C_{SWIn} \cdot f_{INn}) + P_{THR}$$

- Přepínací (switching)  $P_{SWI}$

- $C_{SWIn}$  - parazitní kapacity tranzistorů
- $f_{INn}$  – frekvence změn jednotlivých vstupů

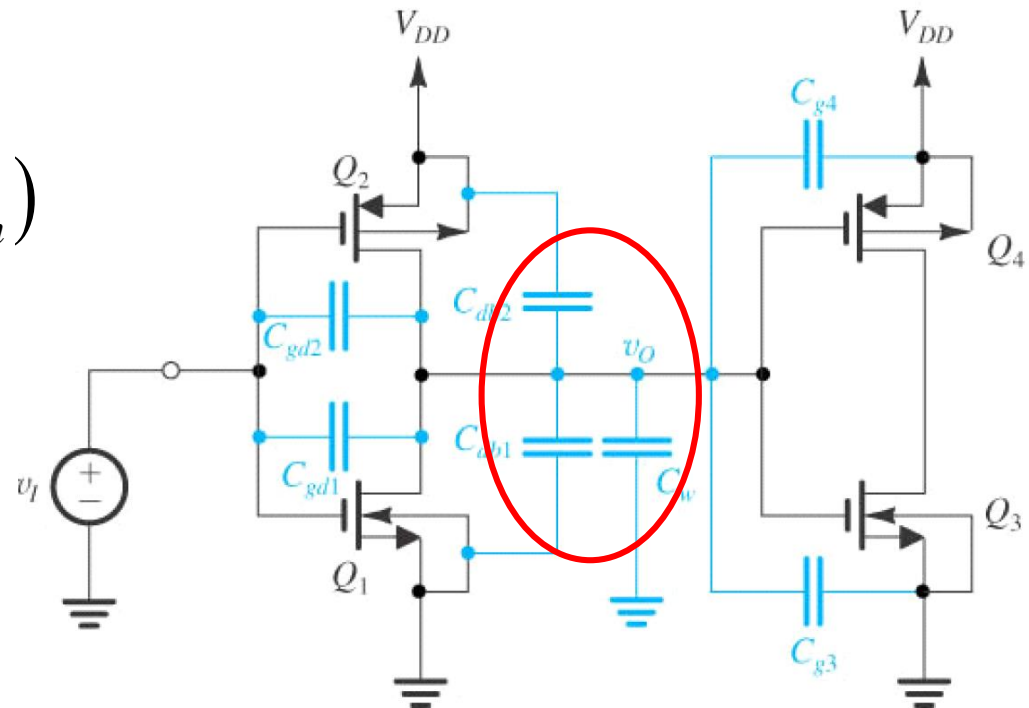
- Zkratový (through)  $P_{THR}$

- Při přechodu mezi log. úrovněmi mohou být horní i dolní tranzistor oba krátce otevřeny = svod mezi VDD a GND
- Závisí na délkách hran signálů - pro krátké hrany signálů je zanedbatelný oproti přepínacímu proudu



- Příkon do zátěže (load)
  - Dán nabíjením a vybíjením parazitních kapacit na výstupu hradla
  - $f_{OUTn}$  – příslušné frekvence, se kterými se mění výstup jednotlivých hradel obvodu
  - $C_{Ln}$  – parazitní kapacity (výstup hradla, spoje mezi hradly)

$$P_L = V_{DD}^2 \cdot \sum_{n=1}^N (C_{Ln} \cdot f_{OUTn})$$



- Energie

- Na změnu log. úrovně 0->1 a 1->0  $E = C \cdot V_{DD}^2$

- Dynamický příkon je dominantní

- $\alpha$  - faktor aktivity – počet přechodů mezi log. úrovněmi za periodu synchronizačních hodin

$$P_D = \alpha \cdot f_{CLK} \cdot C \cdot V_{DD}^2$$

- $f_{CLK}$  - frekvence synchronizačních hodin

- Zpoždění

- $V_T$  - prahové napětí

- $\tau$  - zpoždění

- Ideální invertor  $V_T = V_{DD} / 2$

$$\tau = k \cdot C \cdot \frac{V_{DD}}{(V_{DD} - V_T)^2}$$

$$\tau = k \cdot C \cdot \frac{V_{DD}}{(V_{DD} - V_T)^2} = k \cdot C \cdot \frac{V_{DD}}{(V_{DD} - V_{DD} / 2)^2} = 4 \cdot k \cdot C \cdot \frac{1}{V_{DD}} \approx \frac{1}{V_{DD}}$$



- Faktor aktivity  $\alpha$ 
  - Je dán pravděpodobnostmi přechodů mezi úrovněmi pro uzel obvodu  $i$

$$\alpha_i = p_i^{0 \rightarrow 1} + p_i^{1 \rightarrow 0}$$

- Efektivní parazitní kapacita obvodu

$$C_L = \sum_i (C_i \cdot \alpha_i)$$

- Uplatňuje se v uzlech  $i$  obvodu, ve kterých dochází k přechodům mezi logickými úrovněmi
- Předpokládáme prostorovou a časovou nezávislost uzlů

- Pro hodinově synchronní obvody

- Frekvence hodin:  $f$
- Frekvence přepínání uzlů v obvodu:  $f = \alpha \cdot f_{\text{CLK}}$

- Příklad

- Hodinový signál:  $\alpha = 1$
- Dynamicky řízená hradla – průměrně jedna změna za hodinový signál:  $\alpha = 0,5$
- Staticky řízená hradla - závisí na obvodu, typicky  $\alpha = 0,1$

- Jaký je průměrný ztrátový příkon hradla NAND při přepínání z nuly do jedničky?
  - Předpokládejme, že hodnoty vstupů se mění s frekvencí  $f_{CLK}$  s rovnoměrným rozložením

$$\alpha_{0 \rightarrow 1} = 3/16$$

$$P = 3/16 \cdot f_{CLK} \cdot C_L \cdot V_{DD}^2$$

Current Input	Next Input	Output Transition
00	00	1 → 1
00	01	1 → 1
00	10	1 → 1
00	11	1 → 0
01	00	1 → 1
01	01	1 → 1
01	10	1 → 1
01	11	1 → 0
10	00	1 → 1
10	01	1 → 1
10	10	1 → 1
10	11	1 → 0
11	00	0 → 1
11	01	0 → 1
11	10	0 → 1
11	11	0 → 0

[Zdroj: MIT OpenSourceWare, 2006 ]

- Příklad: snížení napájecího napětí na polovinu
  - Příkon klesne 4x

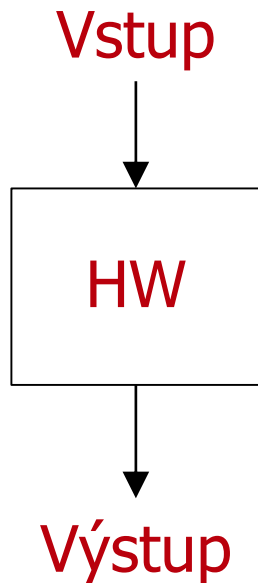
$$P_{(V_{DD}/2)} = \alpha \cdot f \cdot C \cdot (V_{DD} / 2)^2 = 1/4 \cdot \alpha \cdot f \cdot C \cdot V_{DD}^2 = P / 4$$

- Zpoždění vzroste 2x

$$\tau_{(V_{DD}/2)} \approx \frac{1}{\frac{V_{DD}}{2}} = 2 \cdot \frac{1}{V_{DD}} = 2 \cdot \tau$$

- Příklad: Výpočet jednou výpočetní jednotkou

- $f=1 \text{ GHz}$ ,  $V_{DD} = 2 \text{ V}$



$$P_1 = C \cdot 2^2 \cdot f = 4 \cdot C \cdot f$$

$$P_2 = 2 \cdot C \cdot 1^2 \cdot f / 2 = C \cdot f = P_1 / 4$$

- Příklad: Výpočet dvěma identickými jednotkami

- $f=500 \text{ MHz}$ ,  $V_{DD} = 1 \text{ V}$

