

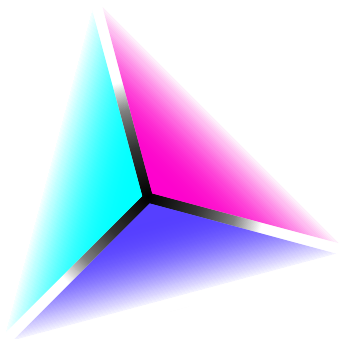
# Struktura a architektura počítačů

Katedra číslicového návrhu  
Fakulta informačních technologií  
České vysoké učení technické

© Hana Kubátová, 2021

## V/V, hazardy v kombinačních obvodech, sběrnice

BI-SAP, duben 2021



# Obsah

- V/V: vstupy a výstupy
- Hardwarová architektura počítače
- Propojování a struktura hradel
- Hazardy
- Sběrnice

# V/V jednotky

- vstupní a výstupní zařízení
  - zařízení mapovaná do paměti
  - zařízení řízená instrukcemi
  - DMA
  - připojení periferních zařízení pomocí sběrnic

Jak dostat data do procesoru z konkrétního vstupu a obráceně?

# V/V a přerušení

- původně obsluhu V/V zařízení řídil procesor
- z důvodů rychlosti snaha o nezávislost V/V operací na procesoru (*historicky první paralelismus*)
- nutnost synchronizace, upozornění V/V zařízení na své požadavky → **přerušení**
- příčina se obvykle zjistí až při „**obsluze přerušení**“ (= kus programu, podprogram pro zpracování přerušení)
- kde najdeme tento podprogram pro zpracování konkrétního vstupu?
- zajištění návratu k původně běžícímu programu

záleží na konkrétním typu procesoru, vždy viz dokumentace  
AVR: <https://courses.fit.cvut.cz/BI-SAP/labs/common-avr/interrupt.html>

## ... přerušení opakování z minula

- **Vnější** (periférie, uživatel)
  - Nemaskovatelné (**NMI**)
  - Maskovatelné (typicky z řadiče přerušení) ... v AVR vše řeší programátor
- **Vnitřní**
  - Instrukce **INT n**, chyby operandů, výsledků, krokování .... AVR nemá

### Co se typicky děje při obsluze přerušení

1. Úklid: uloží se na zásobník informace o přerušeném programu: **SREG, PC, ...**
2. Zakáže se další přerušení (nastavení příznaku I=0 ve stavovém registru CLI)
3. Nastaví se **PC** na adresu začátku podprogramu, který provádí „obsahu“ daného typu přerušení ... podle HW procesoru, pomocí vektorů přerušení
4. Provedení přerušení (= obsluha, realizace podprogramu pro toto přerušení)
5. Návrat z programu obsluhy přerušení a obnova informací patřící původnímu programu, ve kterém bylo přerušení vyvoláno ..., **PC, SREG**

## ... přerušení ...

**Maskování přerušení** pomocí příznaku **I** – ve stavovém registru SREG

- Povolení např. pomocí instrukce **SEI**, která nastaví **I**, tj.  $I \leftarrow 1$
- Zákaz např. pomocí instrukce **CLI**, která nuluje **I**, tj.  $I \leftarrow 0$

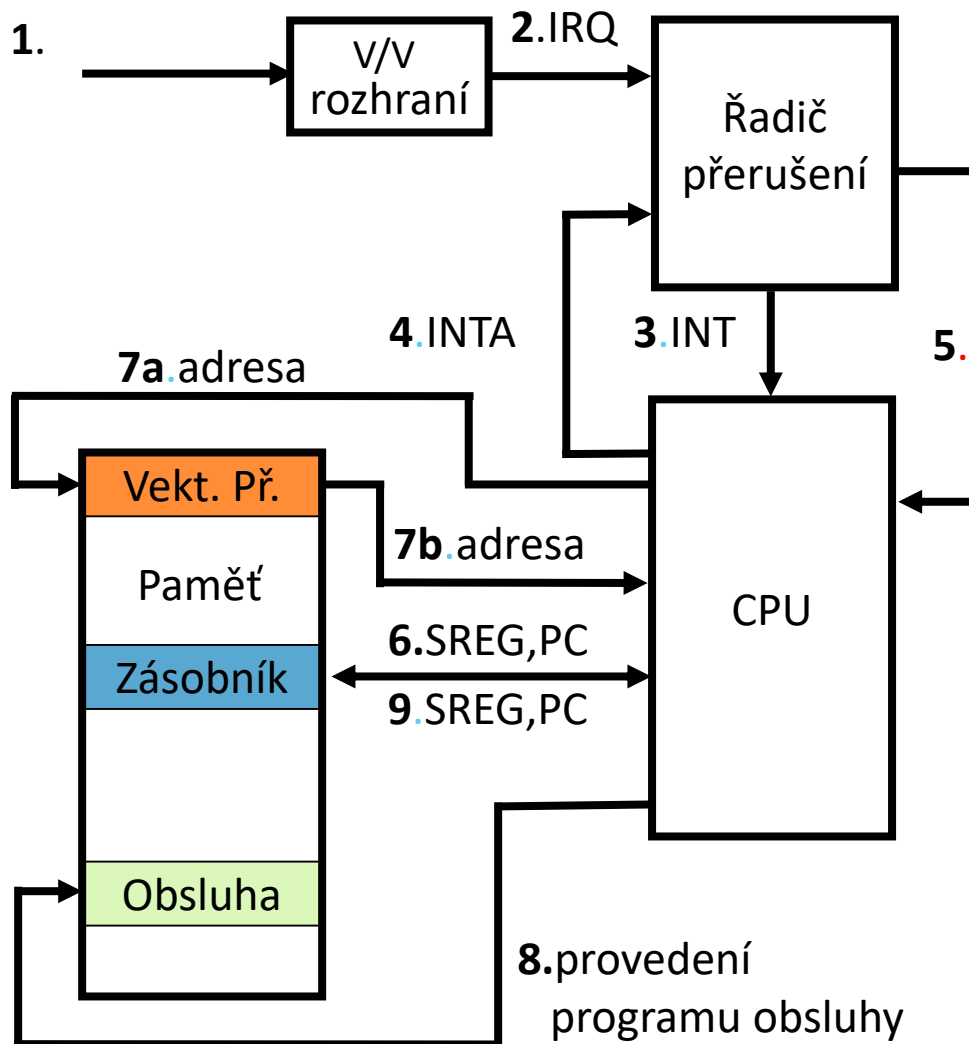
**Vektory přerušení** .... (např. v x86)

- Obsahuje adresu podprogramu obsluhy přerušení
- V paměti *někdy bývá* na nejnižších hodnotách vyhrazen prostor, kde je umístěno **n** (INT n) vektorů přerušení, tj. adres podprogramů pro zpracování daného typu přerušení ... **v AVR si přerušovací rutiny musí napsat programátor**

**Řadič přerušení**

- Hardwarové zařízení přijímající signály z vstupně-výstupných zařízení
- Identifikuje požadavky na přerušení podle jejích priorit **IRQ** (Interrupt request)
- Generuje přerušovací signál **INT**

# Přerušení x86



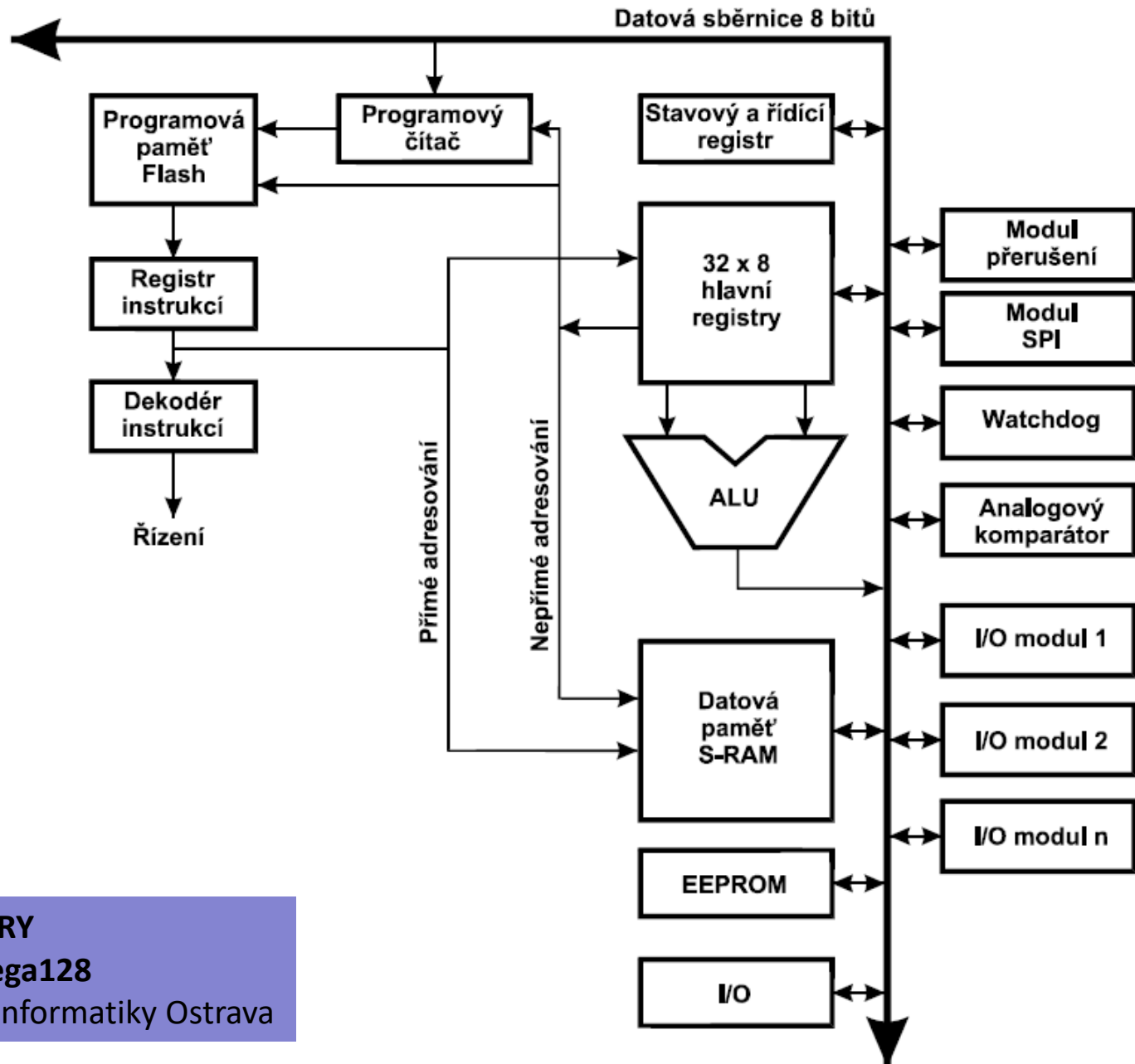
## HW

- 1.- 3. Vznik žádostí o přerušení
4. Rozhodnutí o obsluze (**IF=1** a **INTA**)
5. Identifikace příčiny přerušení (číslo typu)
6. Uložení stavové informace **SREG** a **PC** na zásobník
7. Nalezení začátku podprogramu pro obsluhu daného typu přerušení pomocí vektoru přerušení (nové **PC**)

8. Provedení podprogramu obsluhy přerušení
9. Návrat do přerušeného programu a obnovení **PC** a **SREG**

## SW

## AVR



## MIKROKONTROLÉRY

Atmel AVR – Atmega128

V. Pivoňka, SŠ teleinformatiky Ostrava



# Vstup a výstup: instrukce IN, OUT

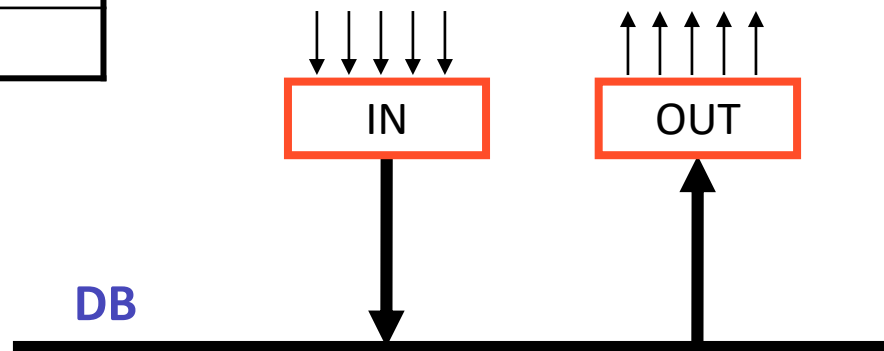
Pro jednoduchou komunikaci s prostředím je možné připojit tzv. brány (porty).

- Ovládá je procesor pomocí instrukcí vstupu a výstupu IN a OUT
- Jsou připojené na sběrnici
- Jejich výstupy ovládají nebo monitorují vnější prostředí

Příklad možných instrukcí pro ovládaní bran:

Instrukce	Význam
IN reg, P	$\text{reg} \leftarrow \text{brána P}$
OUT P, reg	$\text{brána P} \leftarrow \text{reg}$

Adresy těchto bran (registrů) tvoří nezávislý datový prostor oddělený od pamětí pro programy a data



# IN, OUT v AVR

**IN: ulož obsah registru do I/O**

$$R_d \leftarrow I/O(A)$$

**Syntax:**

IN  $R_d, A$

**Operands:**

$$0 \leq d \leq 31, 0 \leq A \leq 63$$

**Program Counter:**

$$PC \leftarrow PC + 1$$

**OUT: ulož I/O lokaci do registru**

$$I/O(A) \leftarrow R_r$$

**Syntax:**

OUT  $A, R_d$

**Operands:**

$$0 \leq d \leq 31, 0 \leq A \leq 63$$

**Program Counter:**

$$PC \leftarrow PC + 1$$

V AVR: na adr. 20h-5Fh V/V registry  
dosažitelné instrukcemi IN a OUT

# V/V zařízení mapovaná do paměti

- Část adresového prostoru není využívána jako paměť.
- Tyto adresy se používají jako čísla zařízení V/ V.
- Tomu musí odpovídat dekodéry adres.
- Pro výstup se používají instrukce „zápis do paměti“.
- Pro vstup se používají instrukce „čtení z paměti“.

AVR:

00h-1Fh registry R0-R31,

20h-5Fh V/V registry dosažitelné instr.  
IN a OUT,

60h-FFh další V/V reg.,

100h-4FF paměť RAM pro volné použití  
a pro zásobník

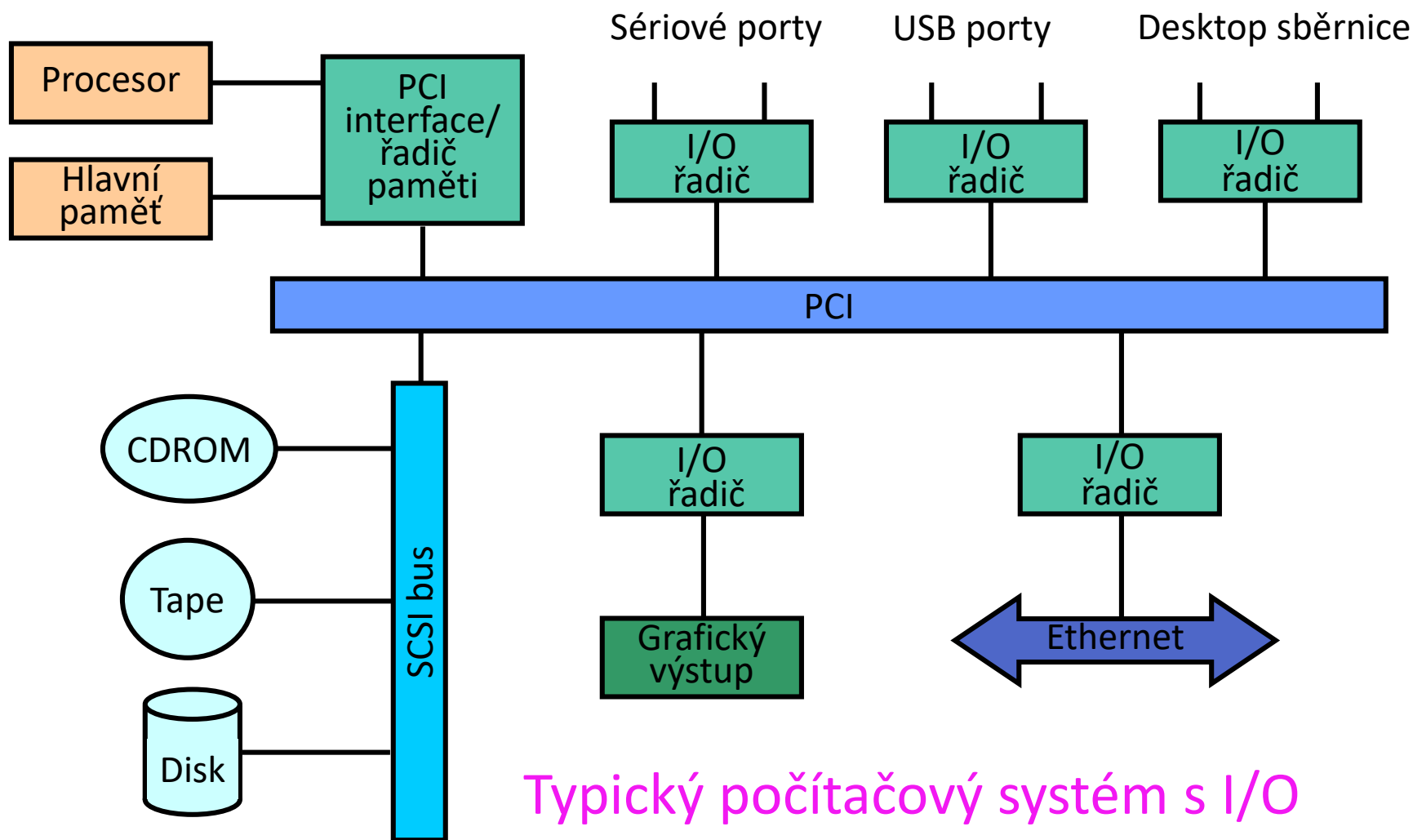
# V/V zařízení řízená instrukcemi

- Zařízení V/V mají vlastní adresový prostor.
- Existují speciální instrukce pro vstup a výstup.

## Možnosti:

1. Instrukce trvá, dokud není provedena příslušná operace (popř. její nezbytná část).
2. Existuje instrukce, kterou lze zjistit stav zařízení.  
Programově (s využitím dotazů na stav zařízení) se zajišťuje potřebné čekání.
3. Po provedení instrukce pro vstup nebo výstup pracuje zařízení autonomně.
  - Procesor může v té době provádět jiné instrukce.
  - Ukončení činnosti periferního zařízení je signalizováno přerušením.

# Hardwarová architektura počítače



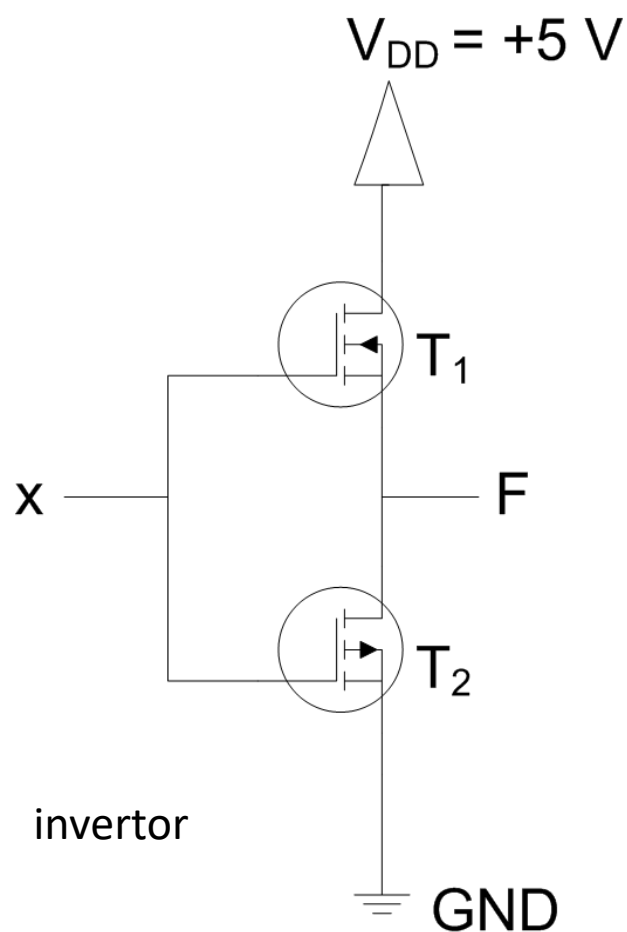
Typický počítačový systém s I/O propojení sběrnice.

# Připomenutí z BI-CAO

- výstupy standardních hradel nelze spojovat
- třístavová logika: další vstup, který na výstup(y) obvodu „pustí“ data nebo nastaví výstupy do stavu Z (vysoká impedance)
- otevřený kolektor: montážní funkce

# CMOS logika

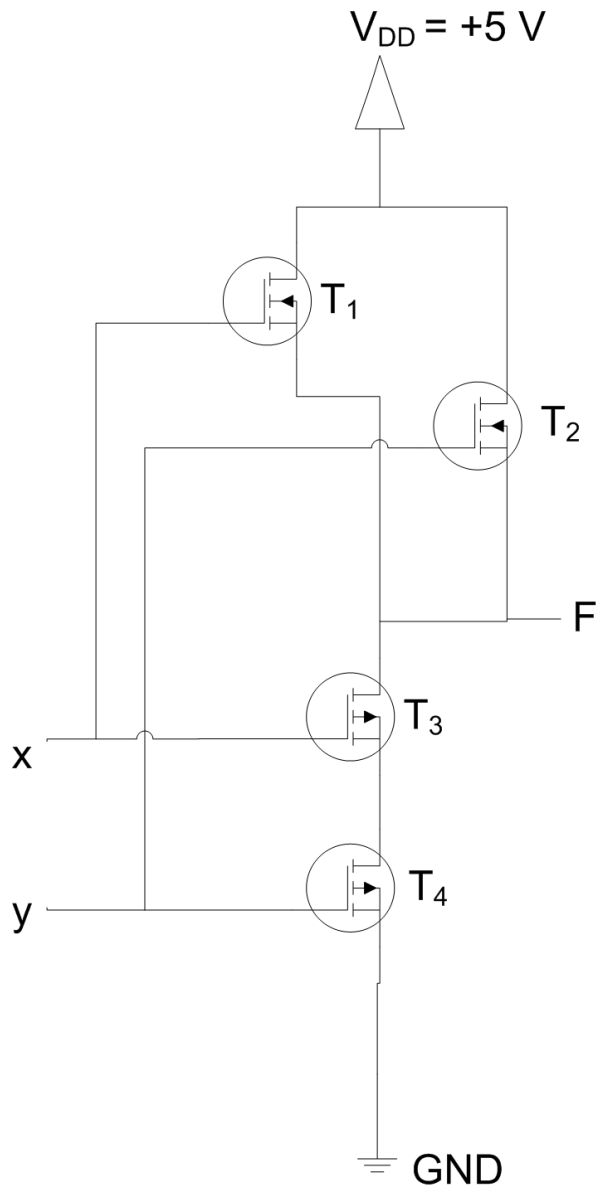
- complementary metal-oxid-semiconductor logic



**Pravdivostní tabulka:**

$x$	$T_1$	$T_2$	$F$
$L$	on	off	$H$
$H$	off	on	$L$

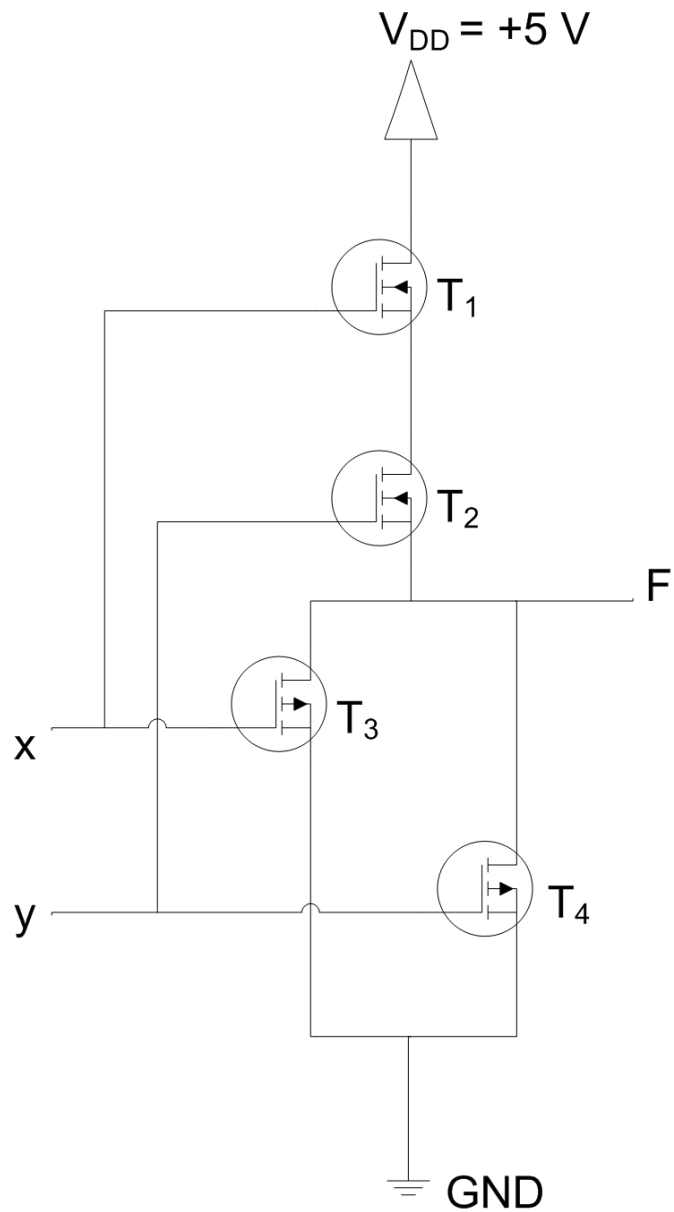
## NAND



$x$	$y$	$T_1$	$T_2$	$T_3$	$T_4$	$F$
$L$	$L$	on	on	off	off	$H$
$L$	$H$	on	off	off	on	$H$
$H$	$L$	off	on	on	off	$H$
$H$	$H$	off	off	on	on	$L$

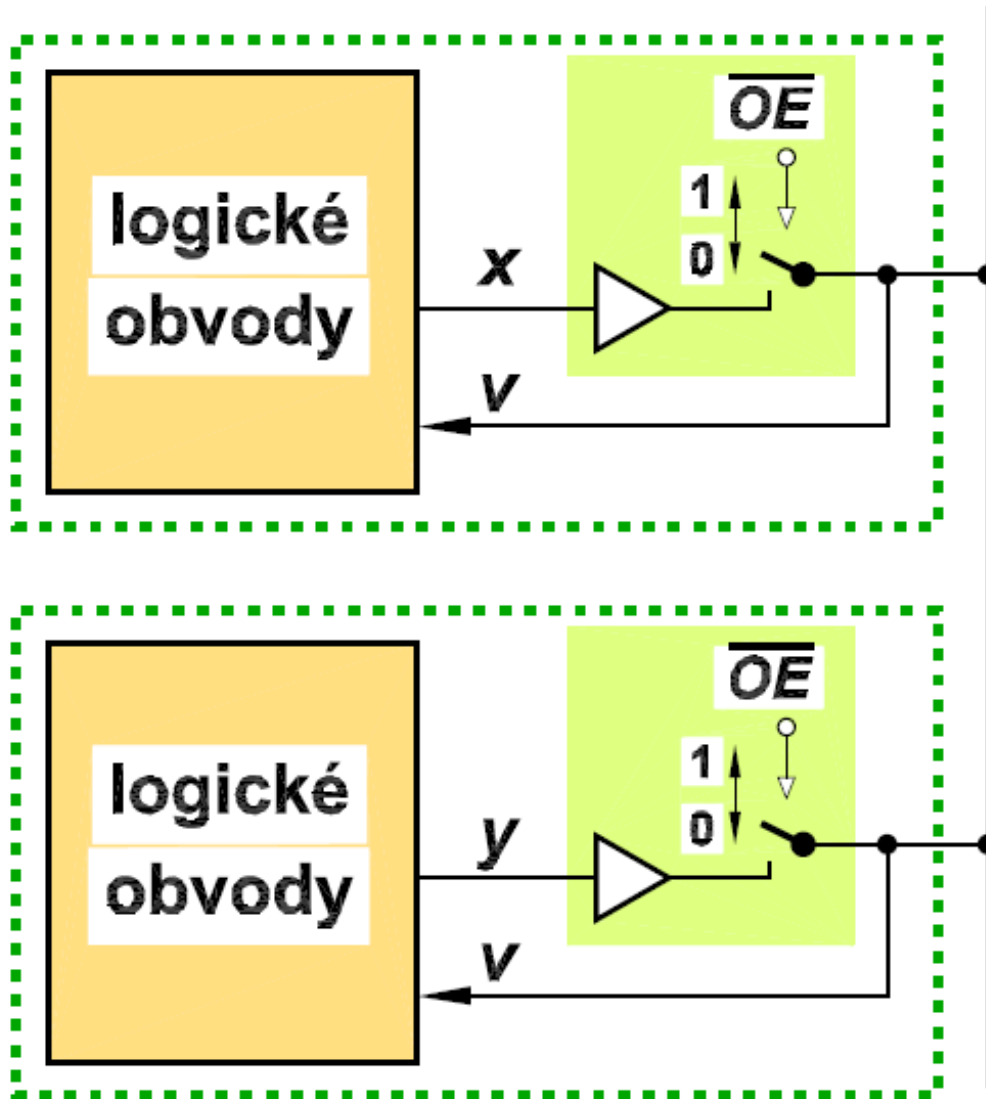


# NOR



$x$	$y$	$T_1$	$T_2$	$T_3$	$T_4$	$F$
$L$	$L$	on	on	off	off	$H$
$L$	$H$	on	off	off	on	$L$
$H$	$L$	off	on	on	off	$L$
$H$	$H$	off	off	on	on	$L$

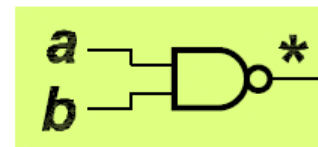
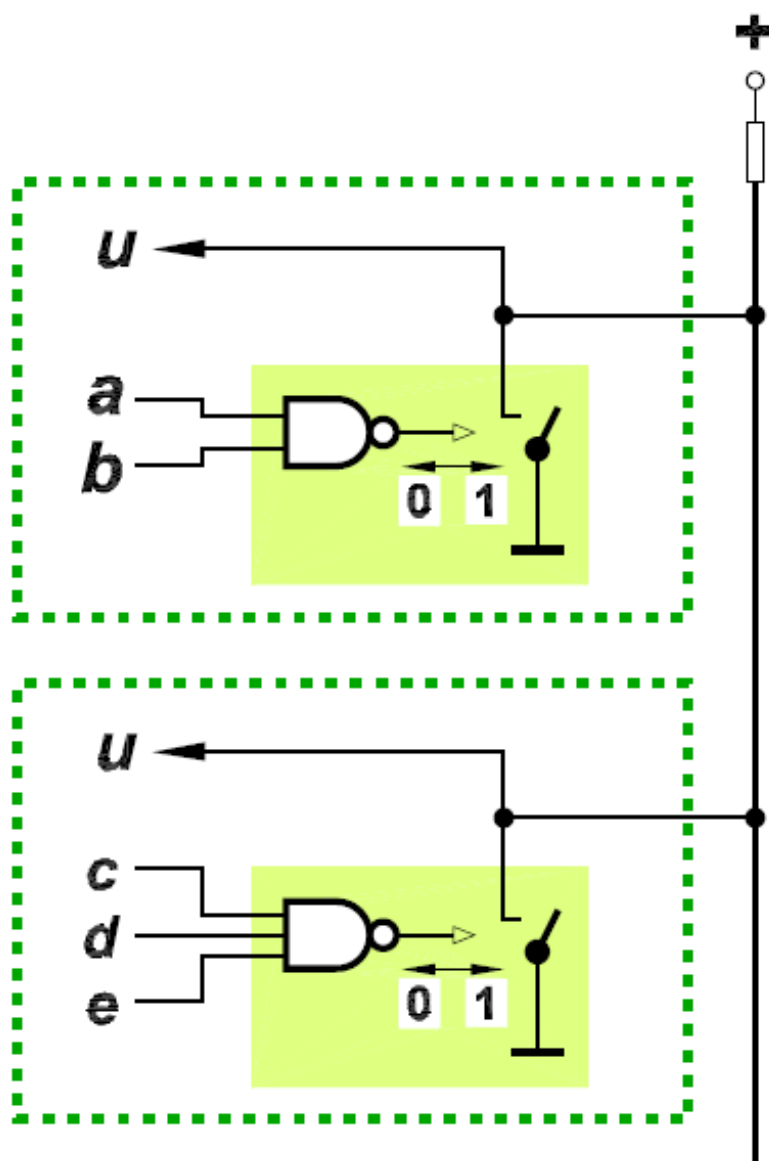
# Třístavový výstup



$\overline{OE}$	výstup
0	$\left. \begin{matrix} 0 \\ 1 \end{matrix} \right\} = x$
1	Z

zajištění, aby jen jedna  
jednotka měla  
povolený výstup  
(output enable)

# Otevřený kolektor



$a \cdot b$	$c \cdot d \cdot e$	$u$
0	0	1
0	1	0
1	0	0
1	1	0

0  $\Rightarrow$  „někdo vysílá“

1  $\Rightarrow$  „nikdo nevysílá“

zde agresivní nula, montážní součin (negovaný)

# Hazardy

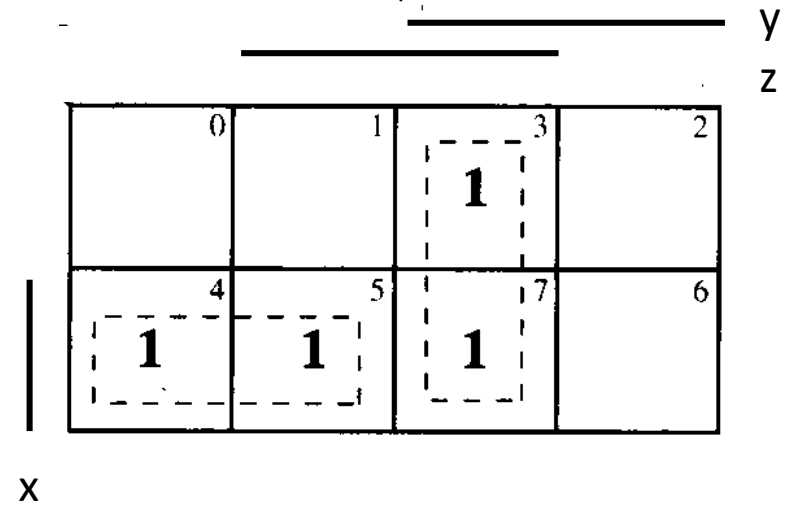
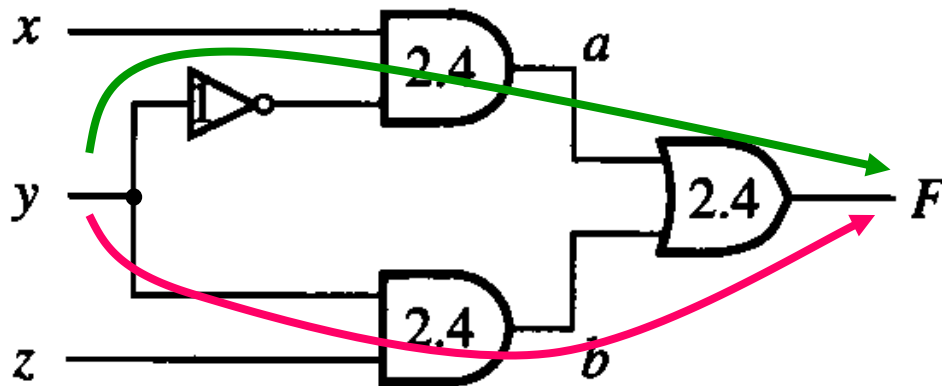
- Co je hazard
- Důvody vzniku hazardů
- Nalezení
- Kdy vadí a kdy ne

**Poznámka:** zde se budeme zabývat jen statickým hazardem.  
Existuje ještě dynamický hazard a hazardy v  
sekvenčních obvodech asynchronních.

# Co je a jak vzniká hazard

- Krátká neočekávaná změna výstupního signálu – **glitch**
- Signál se ze vstupu na výstup šíří různými, ale konvergujícími cestami, z důvodů zpoždění na hradlech i vodičích cestami s různým zpožděním
- Statický hazard – výstup má být trvale v 0 nebo 1 (má mít stejnou úroveň), místo toho se objeví krátká změna do opačné úrovně
- 0-1-0 ..... statický hazard v 0
- 1-0-1 .....statický hazard v 1

# Statický hazard v jedničce - příklad



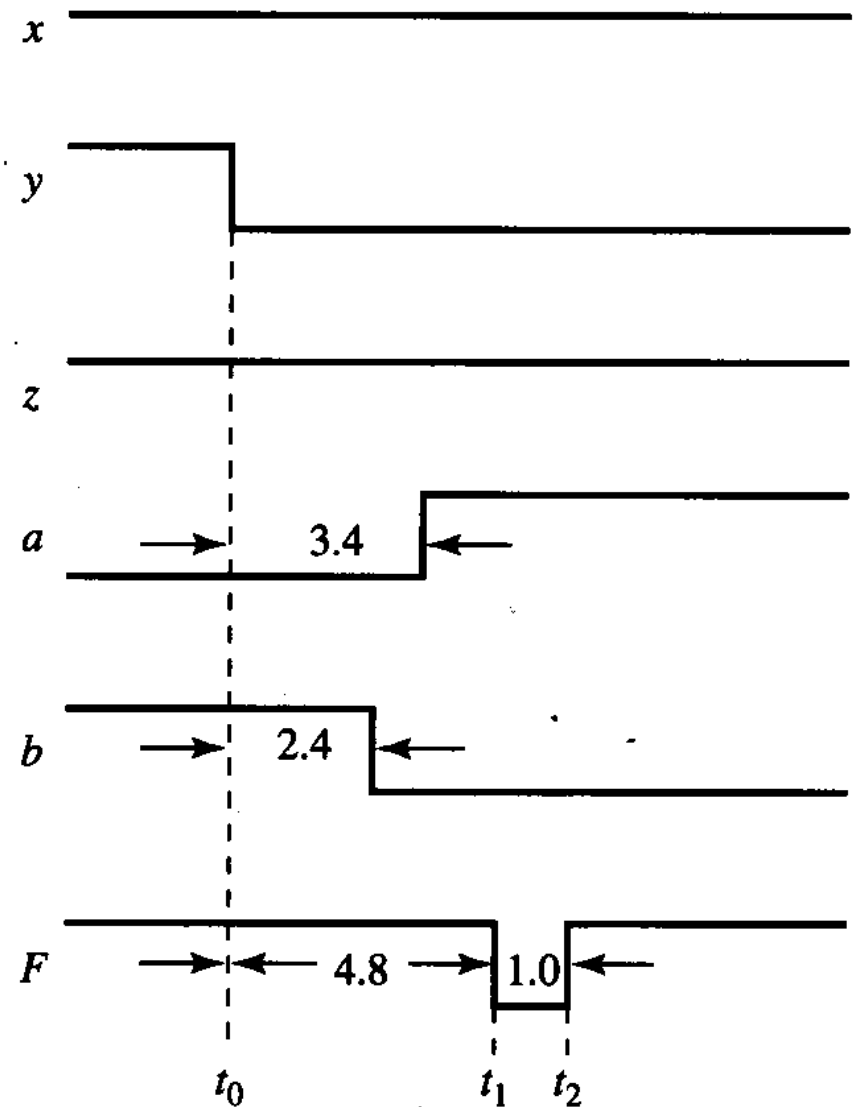
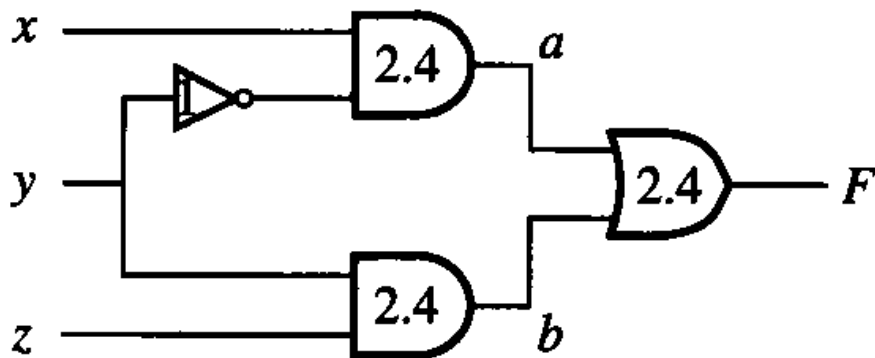
$$F = x\bar{y} + yz$$

Ze vstupu  $y$  na výstup  $F$  vedou dvě cesty, které se nejdříve rozpojí a pak zase spojí

# Hledání hazardu v časovém diagramu

Hledáme podmínky  
pro proměnnou  $y$   
(dvě cesty), tzn.

Pro  $x = z = 1$  (hradla AND)  
 $x\bar{y} + yz = 1$ , ale uvidíme:

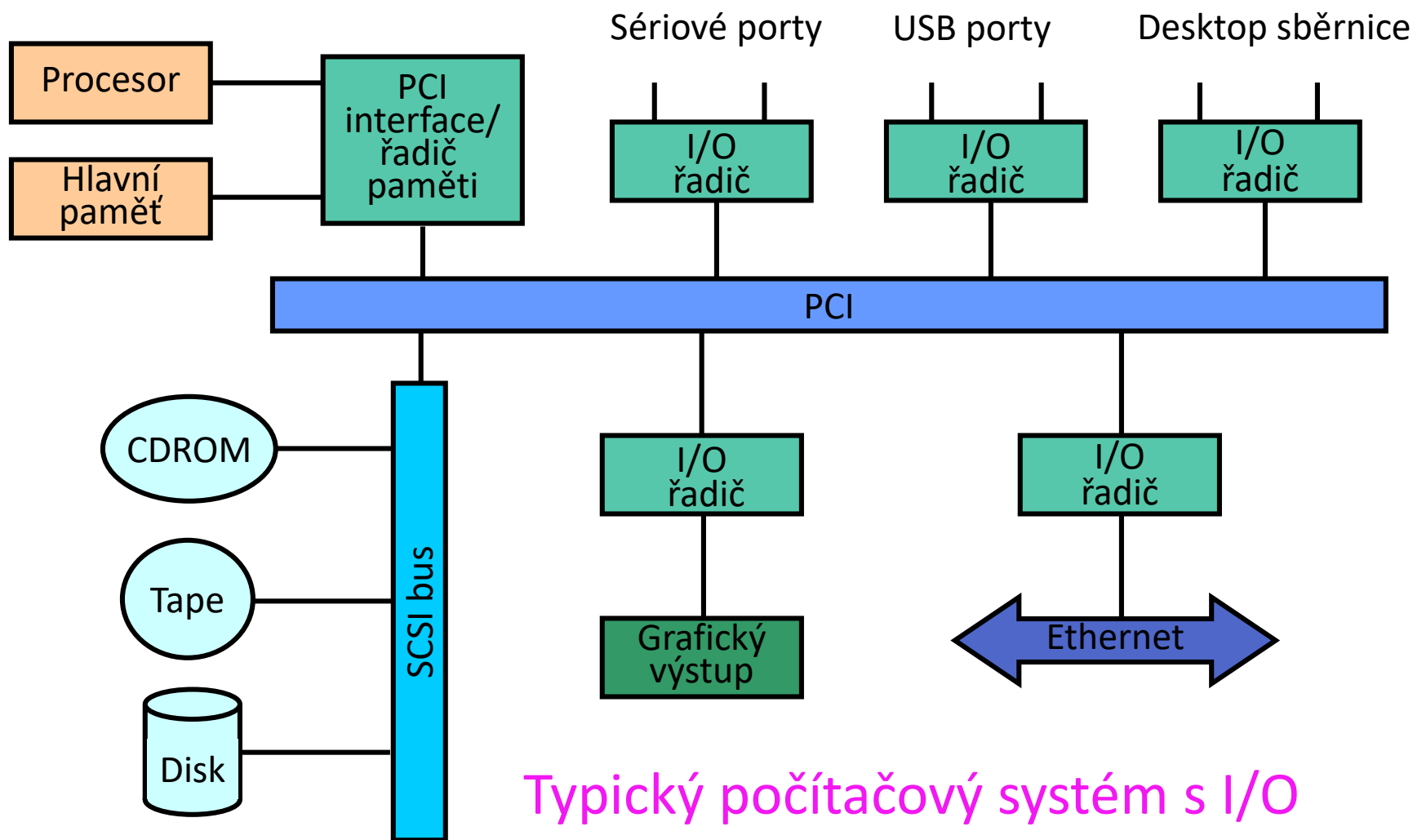


# Kdy hazardy vadí?

- Co je hazard-free design?
- Hazardy v kombinačních obvodech nejsou kritické, protože se výstupy po „nějaké“ (krátké) době vždycky ustálí ve správných hodnotách
- Naopak v sekvenčních obvodech mohou přivést klopný obvod do nesprávného stavu, a tím celý obvod i do nevratného „špatného“ vnitřního stavu
- Řešení – synchronní návrh a správný výpočet maximální hodinové frekvence



# Hardwarová architektura počítače



Typický počítačový systém s I/O

# Sběrnice ... BUS

Soubor vodičů a pravidel určený k propojení jednotek počítače a  
přenos signálů mezi nimi

styk = propojení = rozhraní [interface] ... často **systemová sběrnice**  
(CPU bus)

- **podsběrnice** — obvykle též nazývané sběrnice:
  - adresová [Address Bus]
  - datová [Data Bus]
  - řídicí . . . řídicí a stavové signály [Control Bus]
- adresy a data mohou být multiplexovány (časově):
  - stačí jediná sběrnice pro adresy i data
  - je třeba řídicí signál „adresa / data“

## ... připojení ke sběrnici

- Standardní výstup ... „*totem-pole, push–pull output*“
  - Výstupy nelze spojovat !!!
- Otevřený kolektor ... open collector (OC)
- Třístavový výstup ... tri-state output
  - Připojení ke sběrnici, 3 stavy: H, L, Z – vysoká impedance

*K obvodům se standardními výstupy musíme přidat další logiku a další řízení, např. multiplexor, který na základě adresy vybírá, který člen připojit (a tedy např. přenášet po sběrnici data v něm uložená) a který ne.*

# Typické operace na sběrnících

- operace (typické):

- čtení [read]
- zápis [write]
- čtení – změna – zápis [read – modify – write]
- zápis – čtení [read after write]
- přenos bloku [block data transfer]
- broadcast (oběžník): příkaz všem jednotkám
- broadcast: vyžádání informace od všech jednotek (popř. od určených jednotek)

# Typy sběrníc

Sběrnice:

- paralelní
- sériové

používají se zpravidla:

- » pro systémové účely
- » jako záložní (např. při poruše paralelní sběrnice)

# Typy sběrníc

- jednosměrně řízené

řídící jednotka nečeká na potvrzení příkazu k provedení operace

- obousměrně řízené

.....

- synchronní

řízená hodinami (vysílanými po příslušném vodiči) ... (jednosměrné řízení)

- pseudosynchronní ..... (obousměrné řízení)

hodiny + čekání ... speciální signál(y) — např. WAIT

- asynchronní ..... (obousměrné řízení)

# Handshake

- asynchronní sběrnice = korespondenční režim ..... „hand shake“
  - Operace
  - Příkaz (žádost ... request RQ)
  - Potvrzení (acknowledgement ACK)

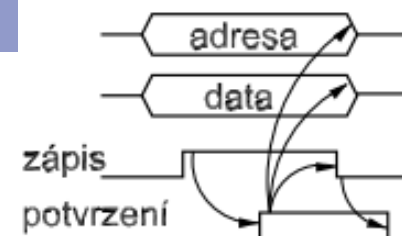
středně vázaný



Jednoduše vázaný



plně vázaný



# Základní parametry sběrníc

Parametr	Význam	Jednotka
Šířka přenosu	Počet bitů, které lze zároveň po sběrnici přenést	bit
Frekvence	Maximální frekvence, se kterou může sběrnice pracovat	Hz
Rychlost (propustnost)	Počet slabik přenesených za jednotku času	B/s



# Přidělování sběrnice

- **centralizované** . . . přidělovač sběrnice (bus arbiter .... funkci přidělovače někdy zastává procesor)
  - cyklické výzvy = přidělovač „nabízí“ postupně a adresně sběrnici jednotlivým jednotkám
  - paralelní = nezávislé .... žádosti a potvrzení
  - sériové (daisy chain)
  - kombinované — paralelní a sériové
- **distribuované** . . . bez přidělovače
  - round-robin — postupné cyklické přidělování
  - prioritní — přidělování na základě priorit