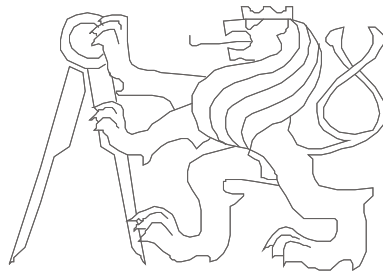


# Architektury počítačů

## I/O podsystém II.

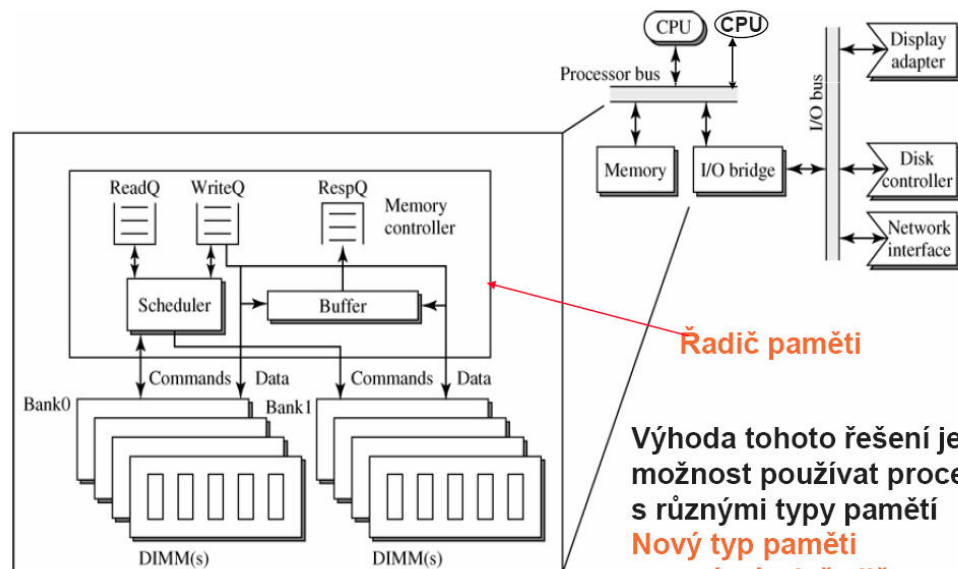


České vysoké učení technické, Fakulta elektrotechnická

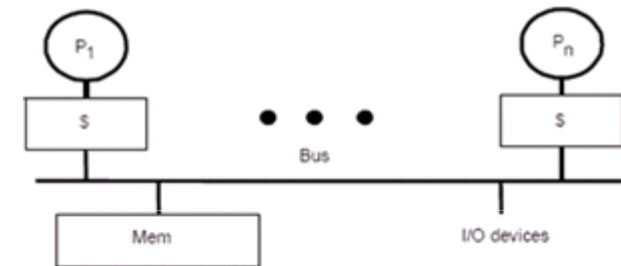
# Opakování - obecně architektura počítače

## 1. Jednoprocesorový stroj

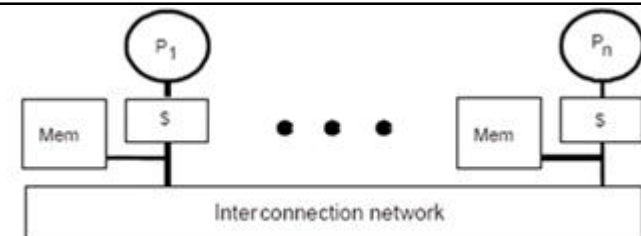
Organizace hlavní paměti – tradiční schema (Intel a další)



## 2. Symetrický multiprocesor - SMP



## 3. Multiprocesor s distribuovanou pamětí - NUMA

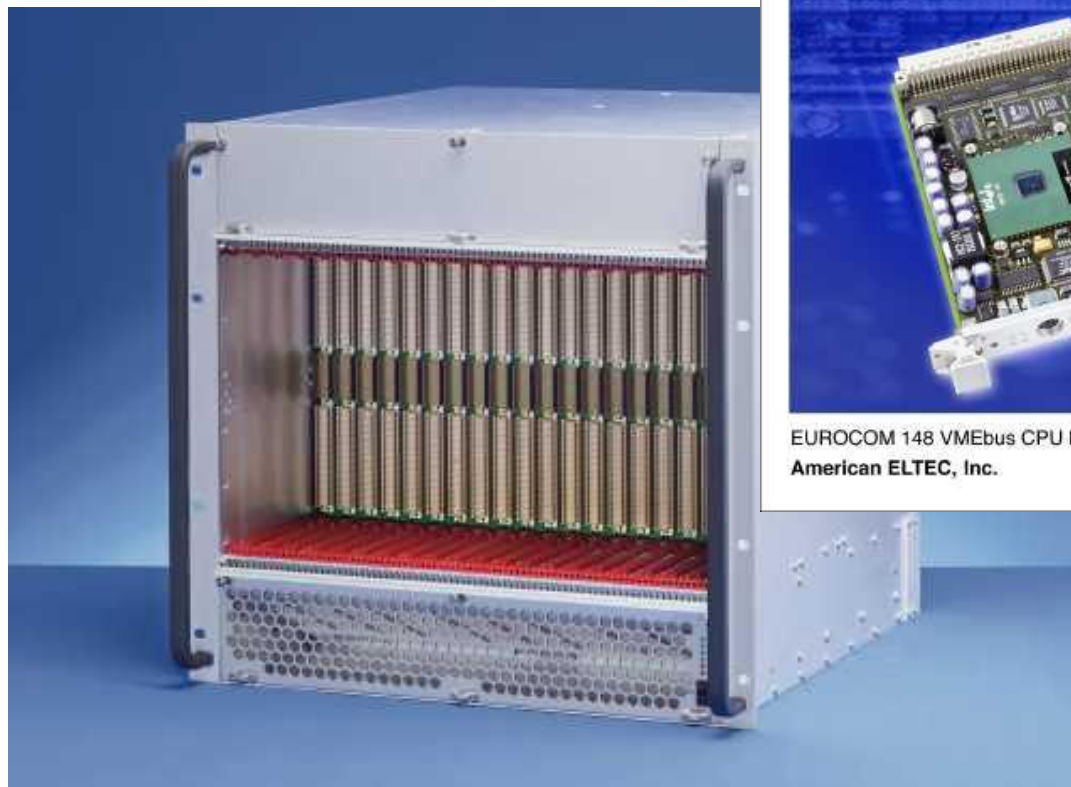


*Mikropočítačové stavebnice,  
protože PCI, resp. PCIe, nejsou jedinou možností propojení*

## Pozor: nejde o LEGO součástek

- ale o skládačku z **desek**.
- Jedná se o obrovský trh (ročně 3B US\$),
- iniciovaly ho v 80-tých letech obě vedoucí procesorové firmy,
  - **Intel, Motorola** (dnes Freescale Semiconductor).
- Některé standardy se udržely až dodnes VME (Motorola),
- jiné zmodifikovaly
  - SBC → xxxPCI (Intel).

# Motivační obrázek



EUROCOM 148 VMEbus CPU board  
American ELTEC, Inc.



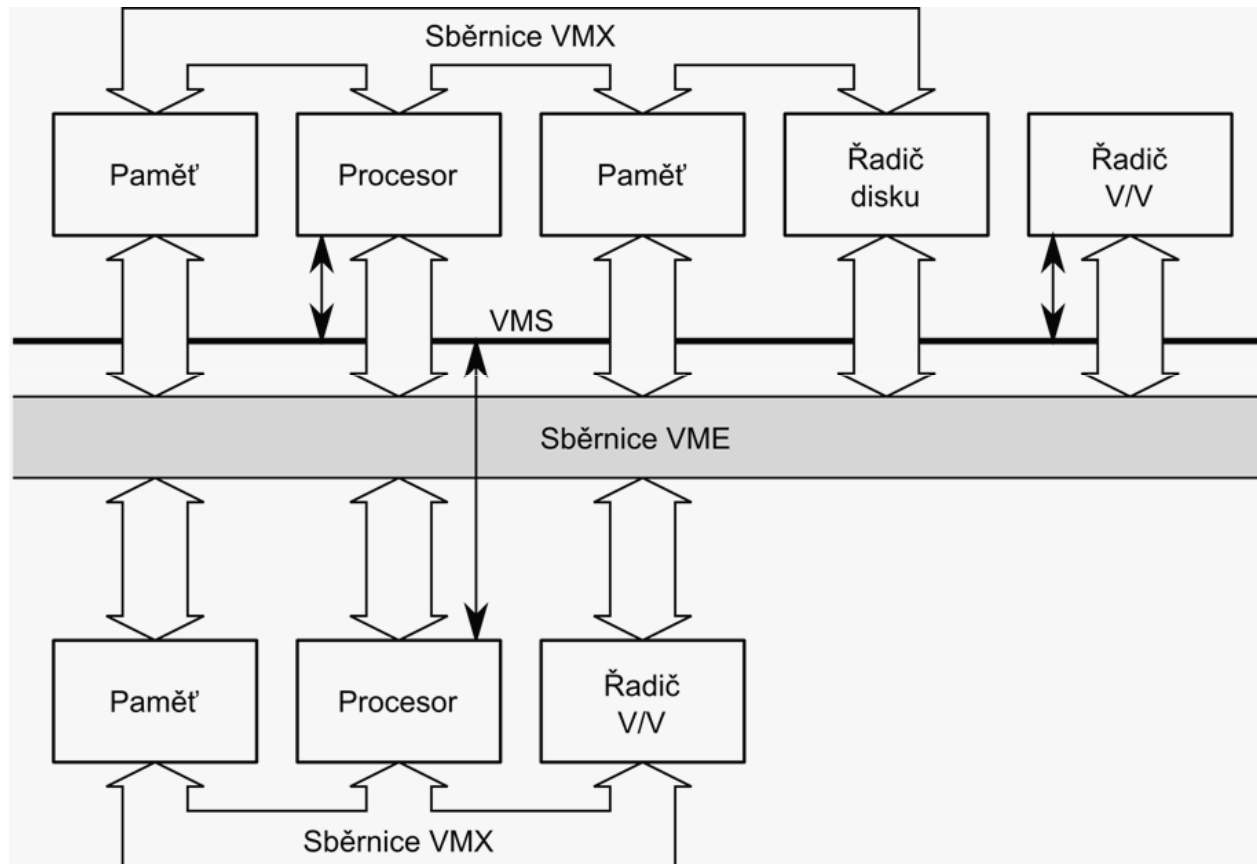
Radstone's PMC GA4, P10 graphics board

01/19/2001

## VME

- Versa Module Eurocard,
  - jsou i jiné výklady zkratky.
- Zavedena Motorolou, Phillipsem, Thompsonem a Mostekem v roce 1981.
- TTL kompatibilní,
- asynchronní přenos rychlostí cca 40 MB/s.
- Standardizována jako IEEE 1014-1987.

# VME - architektura I.



## VME - architektura II.

- Ve skutečnosti jde též o hierarchii sběrnic
  - VME (paralelní, globální),
  - VMX (paralelní, lokální),
  - VMS (sériová, globální).
- Nepřímý, třířadý konektor, Eurokarta se dvěma základními velikostmi,
- původně orientována jen na procesory Motorola.
- (Dnes tu najdete „všechny“).



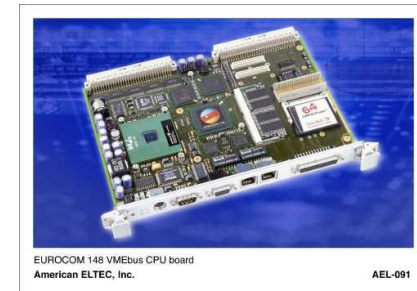
## VME - stručná charakteristika:

- Multiprocessorová,
- paralelní,
- nemultiplexovaná,
- asynchronní,
- obousměrná,
- arbitr centralizovaný, umístěn v 0. pozici skříně (šasí).

# Signály sběrnice VME

## PIN ASSIGNMENTS

Pin Number	-----P1 Connector-----			P2 Connector
	Row A	Row B	Row C	Row B
1	D00	BBSY	D08	+5 V
2	D01	BCLR	D09	GND
3	D02	ACFAIL	D10	RESERVED
4	D03	BGDIN	D11	A24
5	D04	BGDOUT	D12	A25
6	D05	BG1IN	D13	A26
7	D06	BG1OUT	D14	A27
8	D07	BG2IN	D15	A28
9	GND	BG2OUT	GND	A29
10	SYSCLK	BG3IN	SYSFAIL	A30
11	GND	BG3OUT	BERR	A31
12	DS1	BR0	SYSRESET	GND
13	DS0	BR1	LWORD	+5 V
14	WRITE	BR2	AM5	D16
15	GND	BR3	A23	D17
16	DTACK	AM0	A22	D18
17	GND	AM1	A21	D19
18	AS	AM2	A20	D20
19	GND	AM3	A19	D21
20	IACK	GND	A18	D22
21	IACKIN	SERCLK	A17	D23
22	IACKOUT	SERDAT	A16	GND
23	AM4	GND	A15	D24
24	A07	IQR7	A14	D25
25	A06	IQR6	A13	D26
26	A05	IQR5	A12	D27
27	A04	IQR4	A11	D28
28	A03	IQR3	A10	D29
29	A02	IQR2	A09	D30
30	A01	IQR1	A08	D31
31	-12 V	+5STDBY	+12 V	GND
32	+5 V	+5 V	+5 V	+5 V



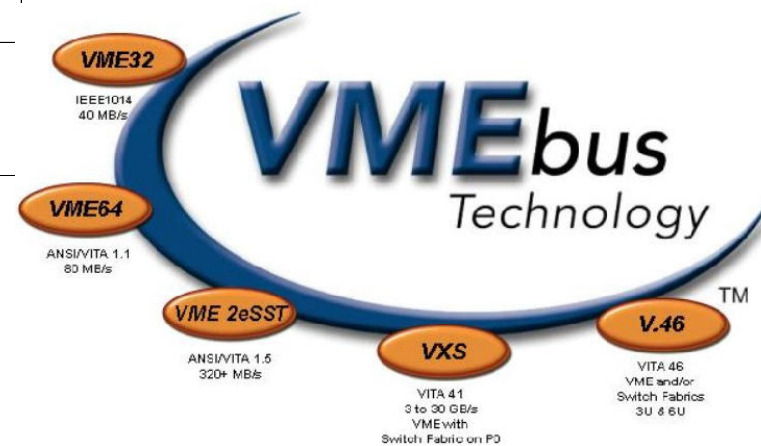
Rows A and C of the P2 connector are user defined.

## Signály sběrnice VME

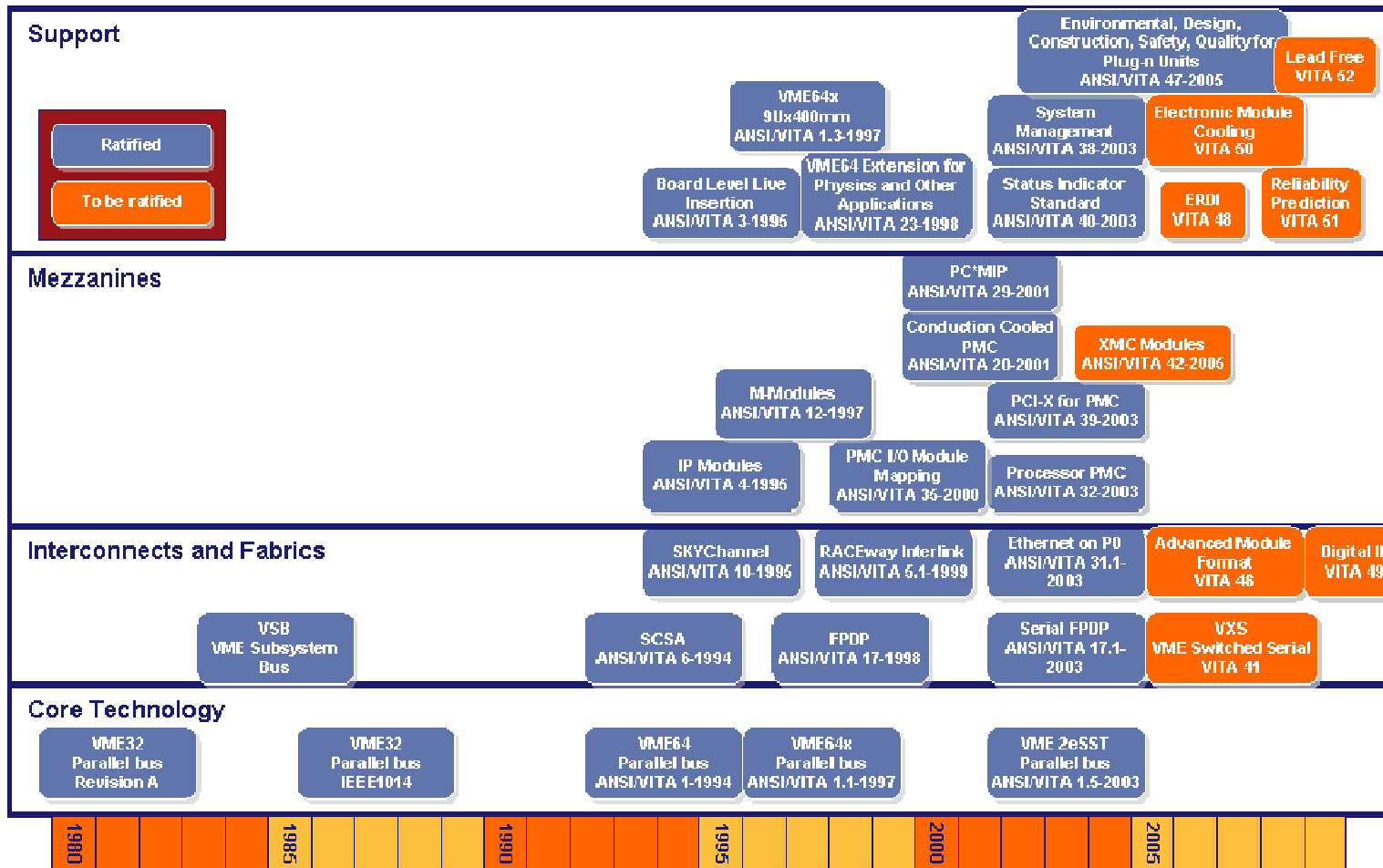
- Čtyři podmnožiny signálů
  - arbitrační,
  - přerušovací,
  - pro datový přenos,
  - utility.
- K těm posledním patří
  - **SYSCLK, SYSRESET#, SYSFAIL#, ACFAIL#.**

## Vývoj VME

Topologie	Max. výkon
VME IEEE-1014	40 MB/s
VME64	80 MB/s
VME2eSST	320+ MB/s



# Vývoj VME jinak

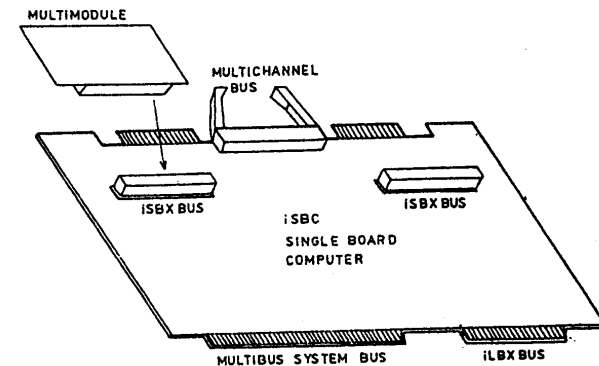
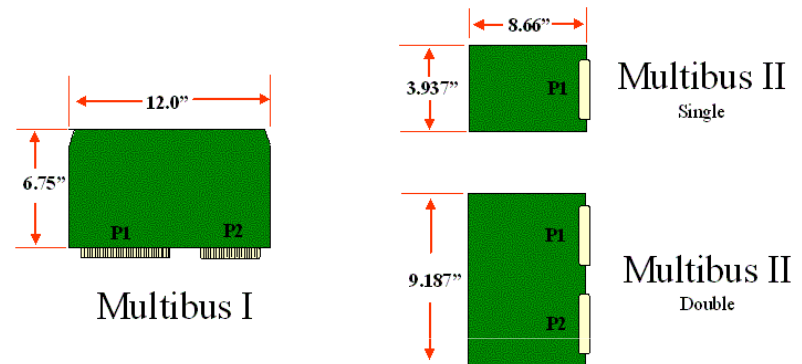


## Zajímají-li vás podrobnosti, tak

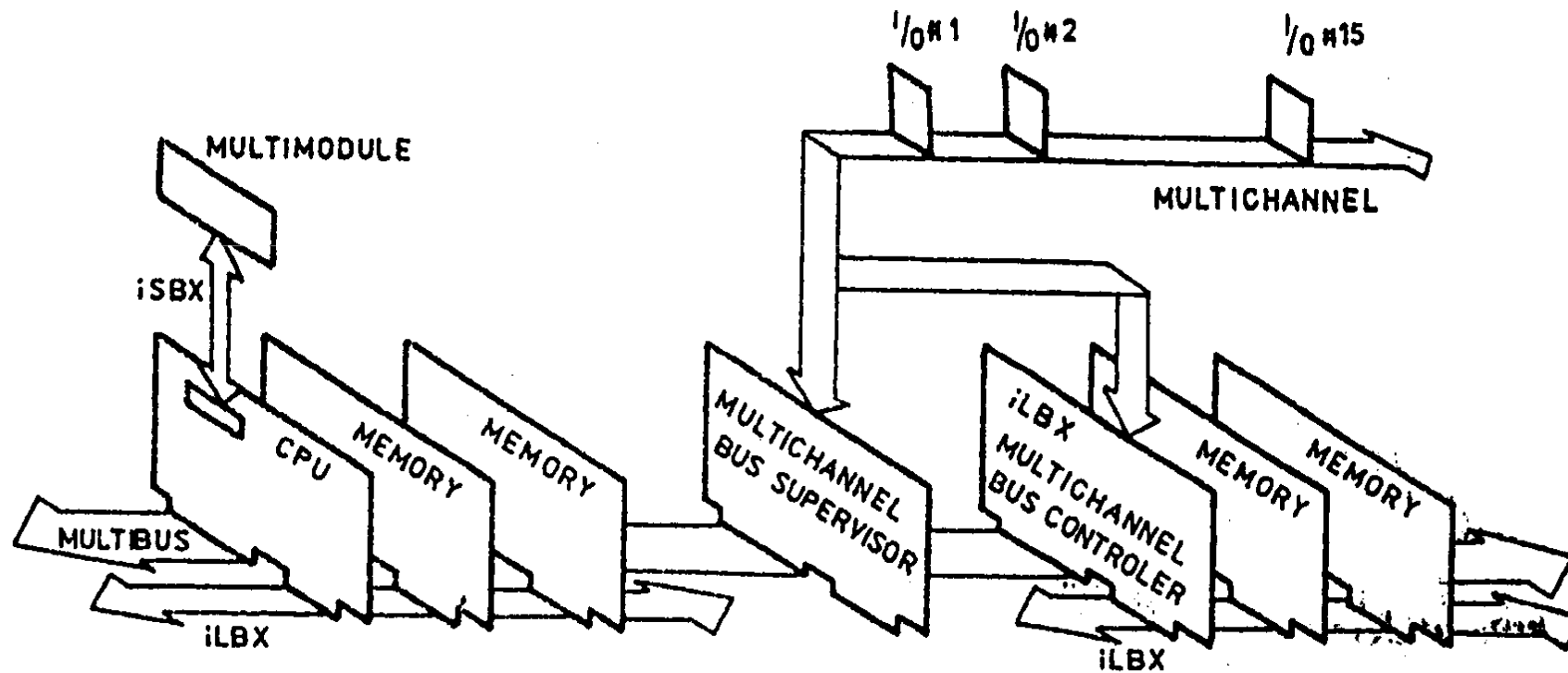
- <http://vita.com>
  - což je server neziskové organizace výrobců a uživatelů VME sběrnice.
  - Založena byla v roce 1984,
  - má i evropskou pobočku.
- Na této adrese najdete (kromě jiného)
  - VMEbus FAQ,
  - přehled výrobců i výrobků.
- Poznámka: i SUN 4 se vyráběl s touto stavebnicí ...

## SBC, resp. MULTIBUS, dnes už jen pro info

- Stručná charakteristika:
- **Single Board Computer**,
- ucelený soubor modulárního hardwaru
  - procesorové desky 8/16/32/64, paměti, I/O rozhraní, periférie, atd.,
- vývojových prostředků a
- backplane sběrnice včetně šasí (bedny se zdroji).
- Vyvinut INTELeM, nasazován od roku 1974.



# MULTIBUS - architektura

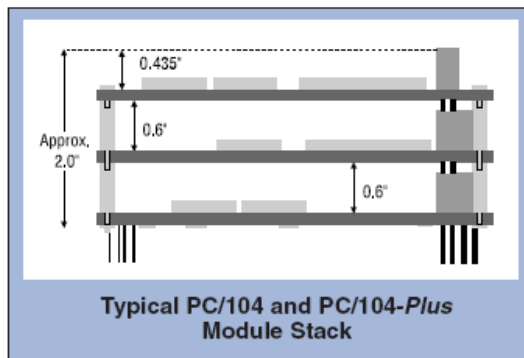
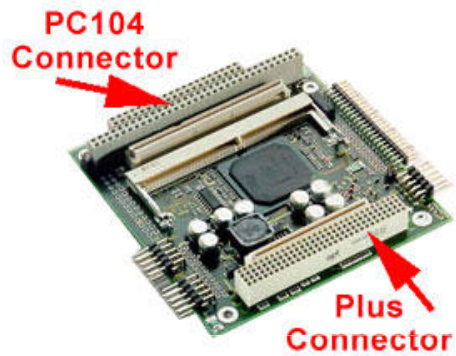




## MULTIBUS shrnutí

- Ani inovace na MULTIBUS II (1987, nepřímý konektor, synchronní protokol, ale tím také nekompatibilita s MULTIBUSEM) neodvrátila jeho **propad**.
- Tím spíše, že se objevil nový favorit, **PCI**,
- a jeho průmyslové verze
  - **CompactPCI, SmallPCI, PC/104-Plus a PC-104.**
- To je ale jiná historie.

# CompactPCI, SmallPCI, PC/104-Plus a PC-104

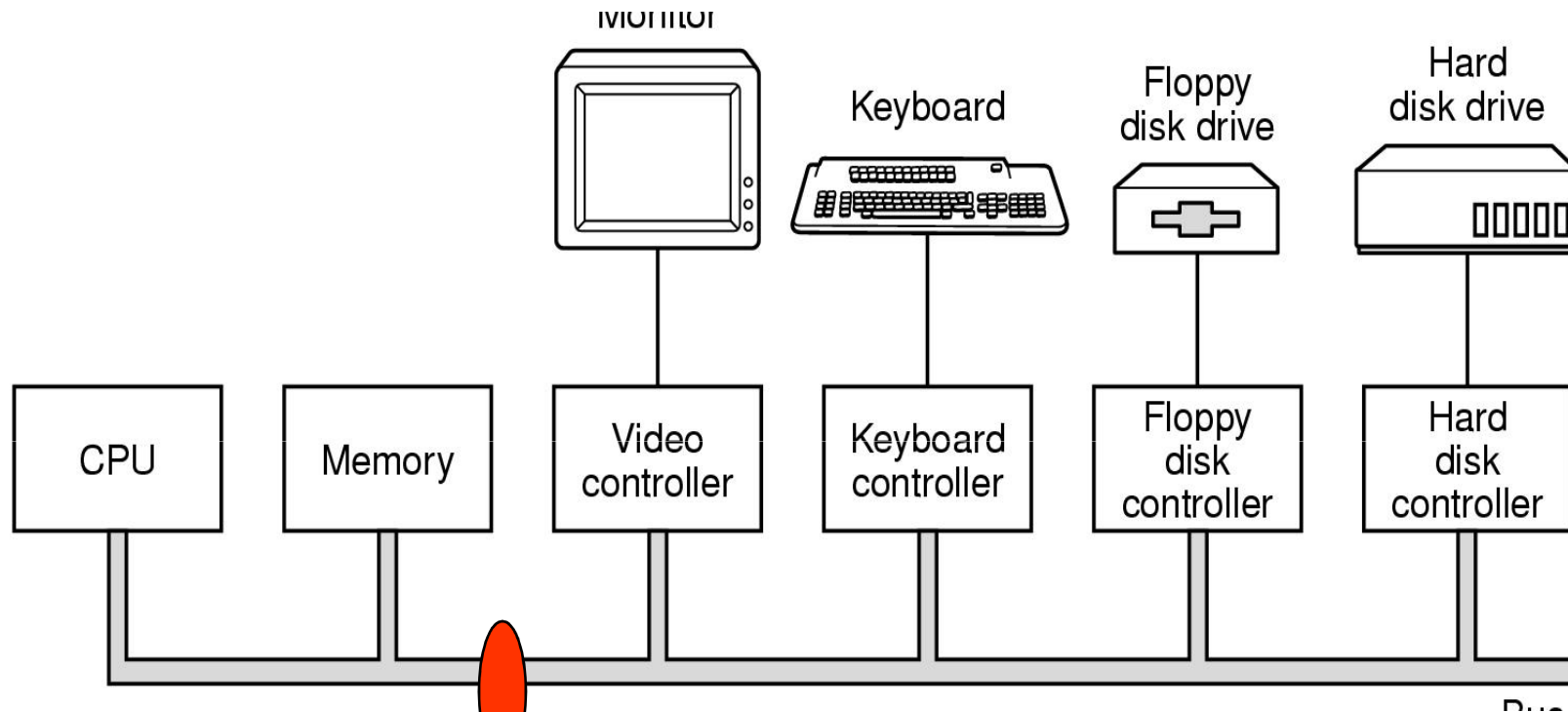


*Pojďme teď připojovat k počítači periferie*

## Poznámka

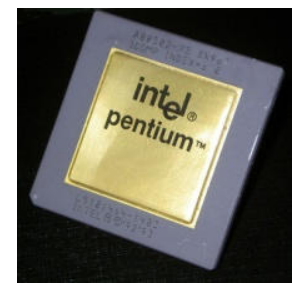
- V dalším výkladu nebude chvíli důležité, zda je procesor typu
  - RISC, nebo
  - CISC.
- K tomu se ještě vrátíme...

# 1. studie - IN, OUT model



?

Typicky: procesory Intel



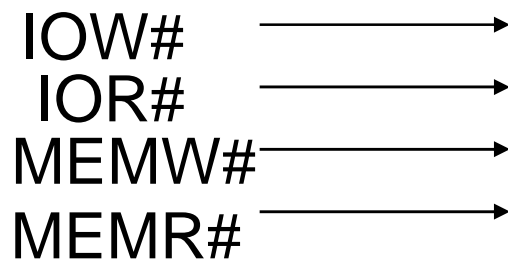
## Jednotlivé komponenty sběrnice



Adresová sběrnice



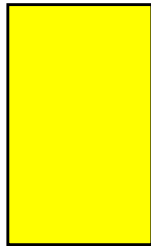
Datová sběrnice



Řídicí sběrnice

Důsledek: oddělené adresové prostory

## Adresové prostory:



registry procesoru

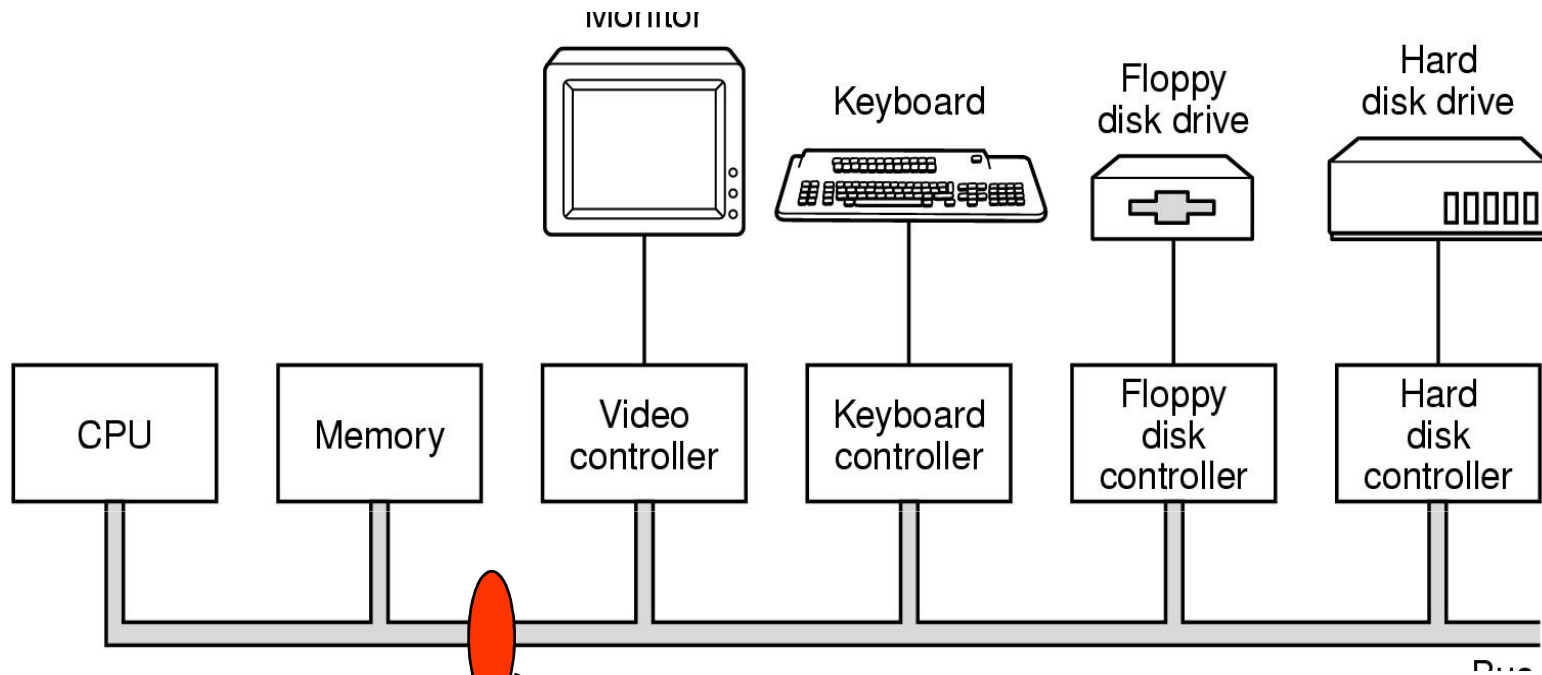


paměť



V/V brány

## 2. studie - MOV model

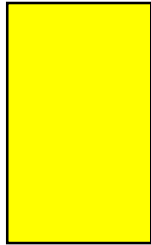


Typicky: procesory Motorola a ARM



## Důsledek: společný adresový prostor

### Adresové prostory:



registry procesoru



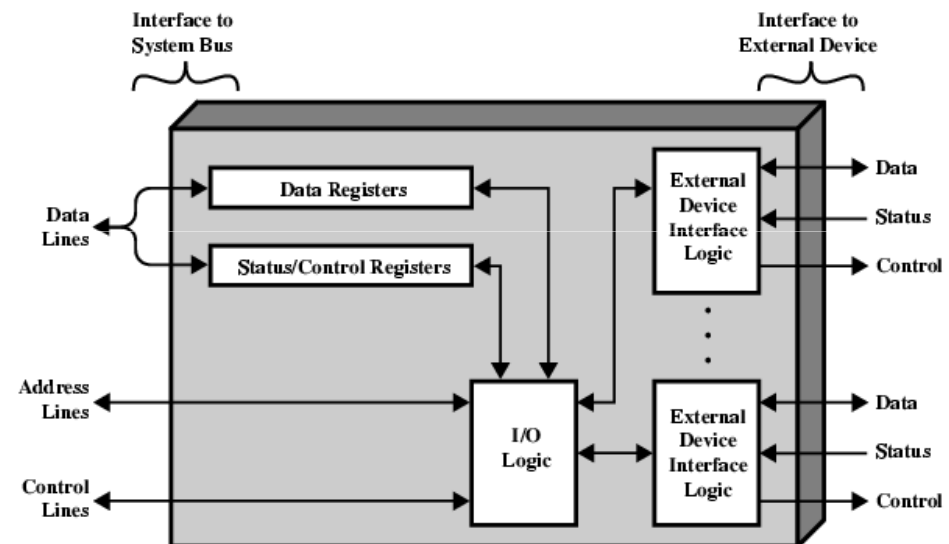
V/V brány  
jsou  
mapované do  
paměti

paměť

*Co to je adaptér?*

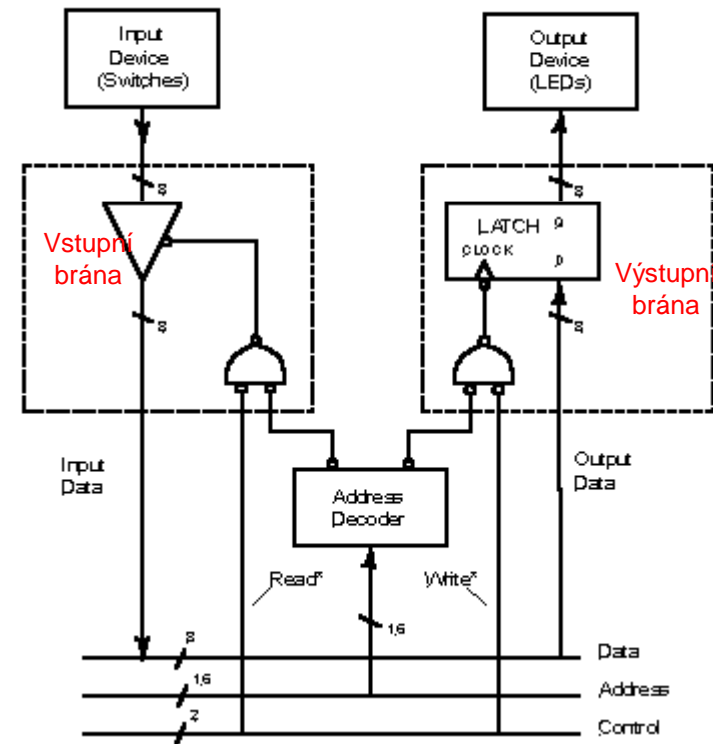
## Je vlastně řadič V/V zařízení, plus interfejs na obě strany

- V PC terminologii se tomuto zařízení říká adaptér.
- Skládá se ze
  - Sběrnicového (v našem případě PCI nebo PCIe) rozhraní,
  - Rozhraní k V/V zařízení a
  - Vlastního řadiče.
- Příklad – řadič disku.

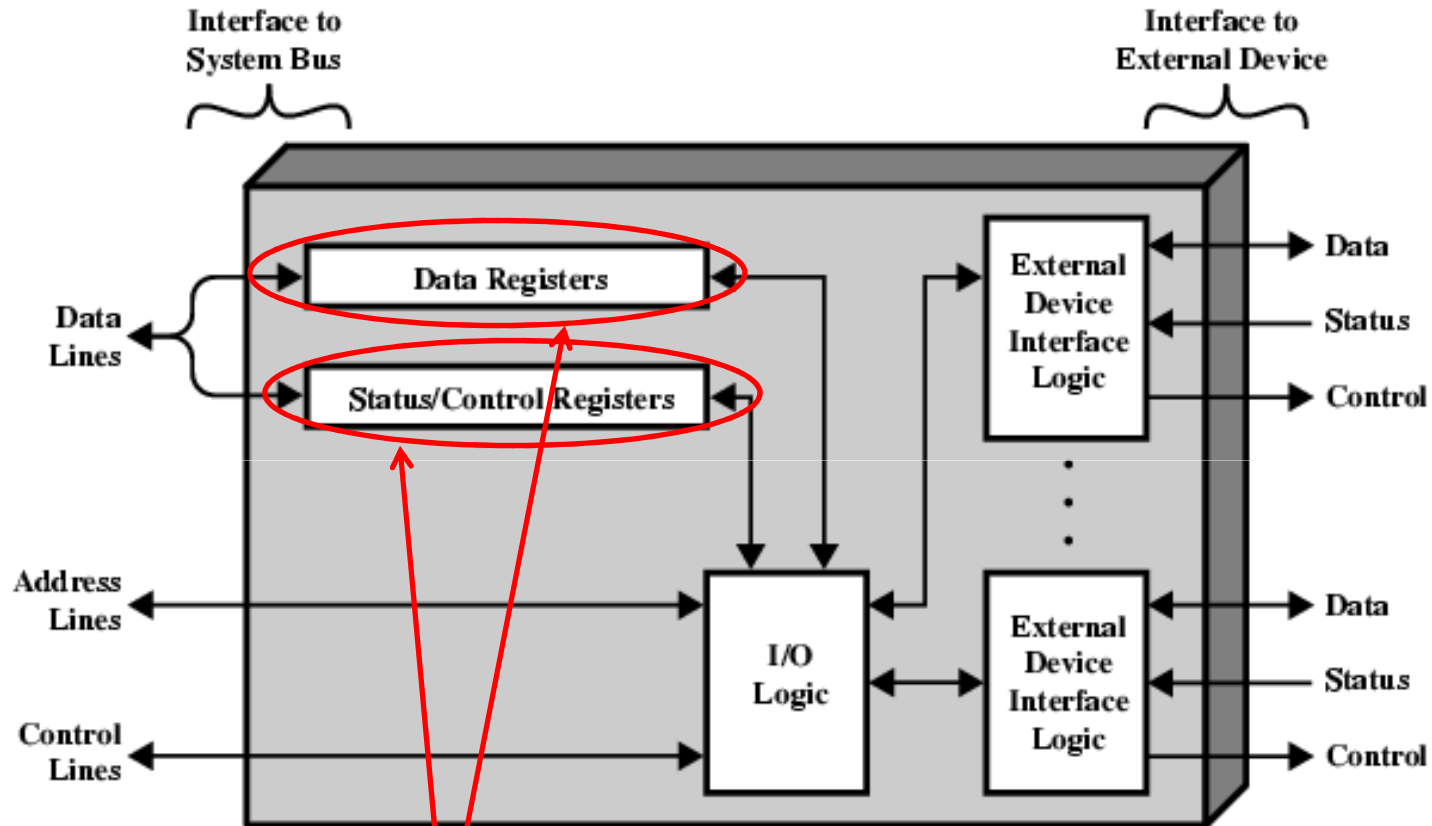


## Brána, anglicky port

- Buňka v adresovatelném prostoru V/V zařízení nebo paměti.
- Stavební prvek interfejsingu.
- Obecně je to 8/16/32 b buňka, registr, ale ...
- uvidíme, že někdy se bez klopných obvodů obejde. Podrobnosti ve cvičeních.



# Blokové schéma adaptéru



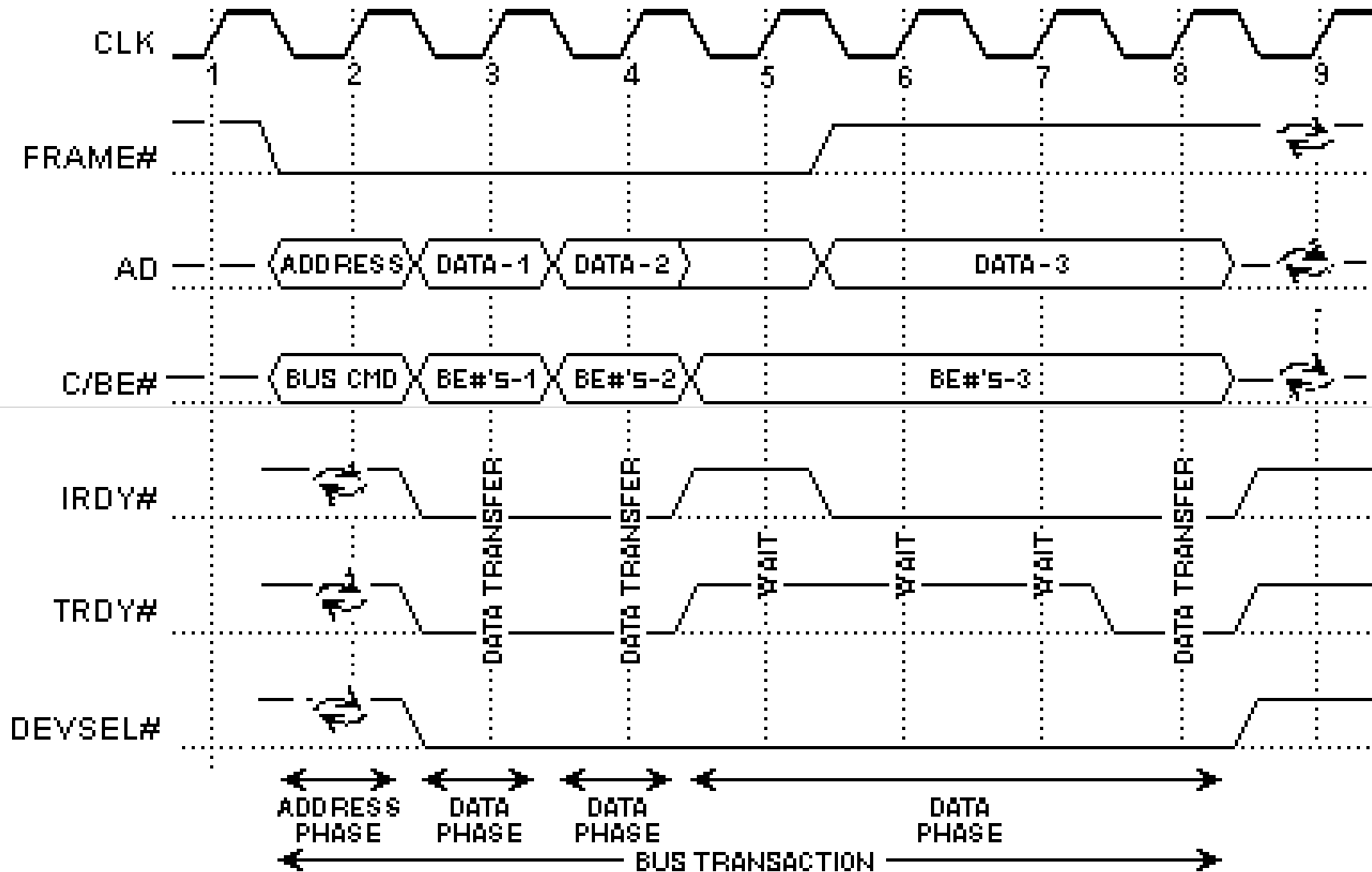
A kde v něm jsou ty brány?

Úkol:

Navrhněte rozhraní adaptéru PCI, který bude nárokovat tři oddělené adresovatelné prostory:

- dva paměťové, každý o velikosti 4 kB,
- jeden V/V o velikosti 16 B.

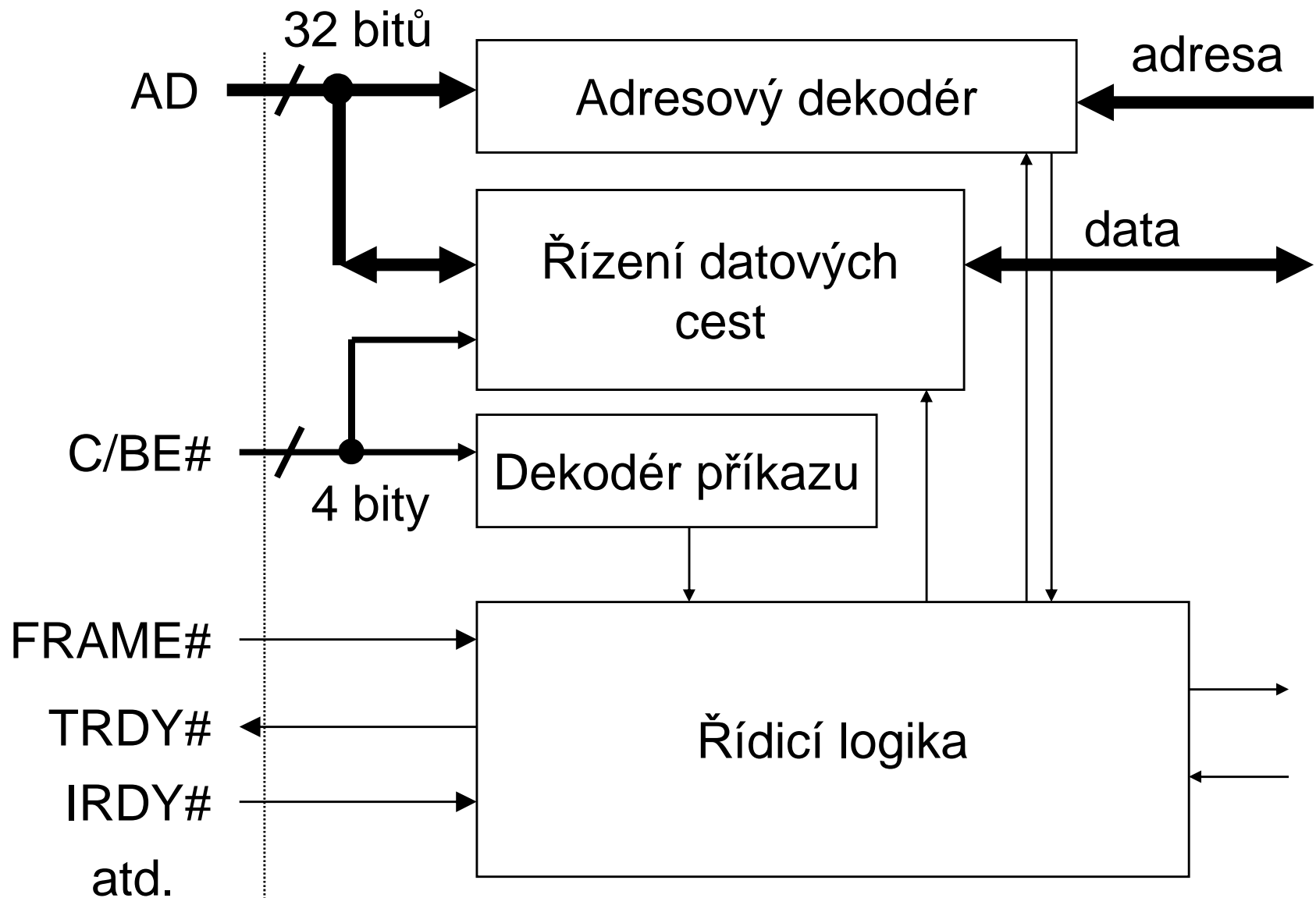
# Sběrnicový cyklus (transakce)



# Bloky rozhraní

- Datová sběrnice, blok řízení datových cest,
- adresové vodiče, adresový dekodér,
- dekodér příkazu,
- řídicí logika,
- (generátor přerušovacího signálu INT#)
  - pokud adaptér využívá přerušení,
- blok žádosti o přidělení sběrnice (arbitrace)
  - navrhujeme-li kartu jako typ *master*.

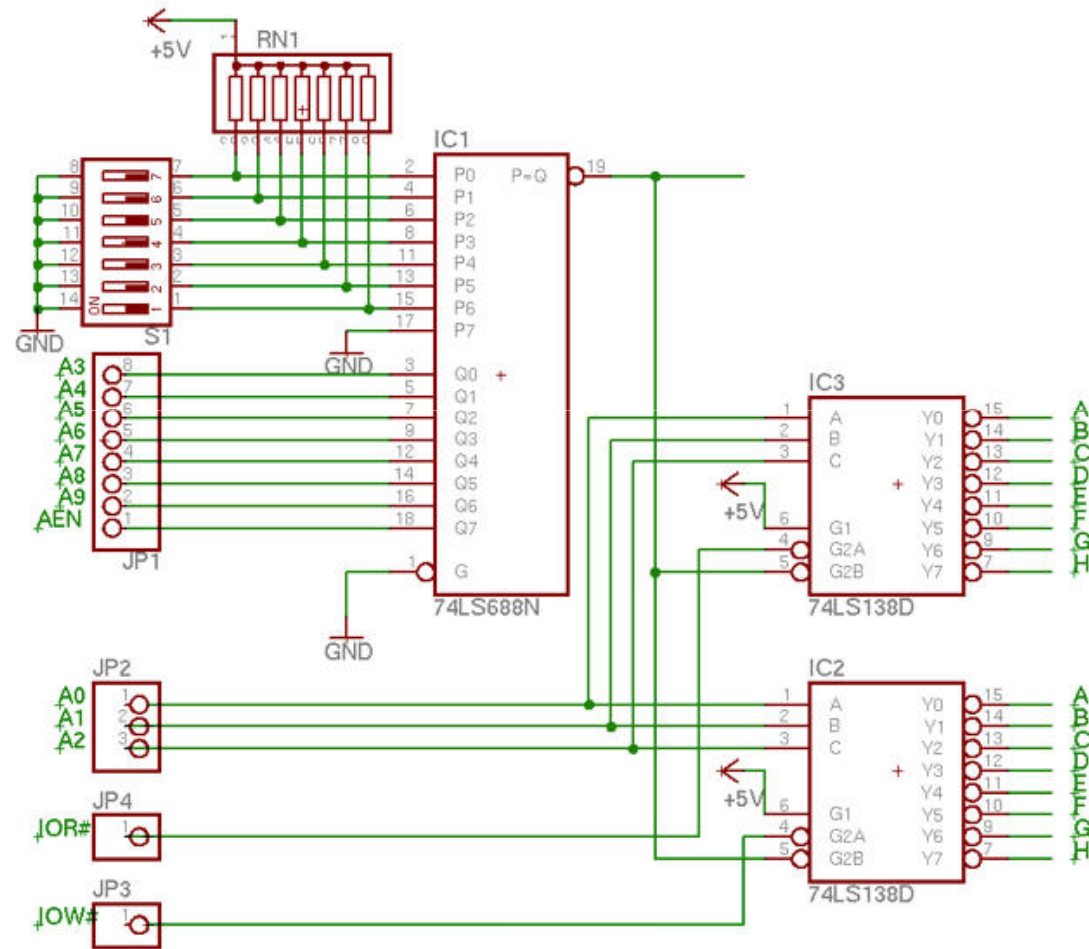




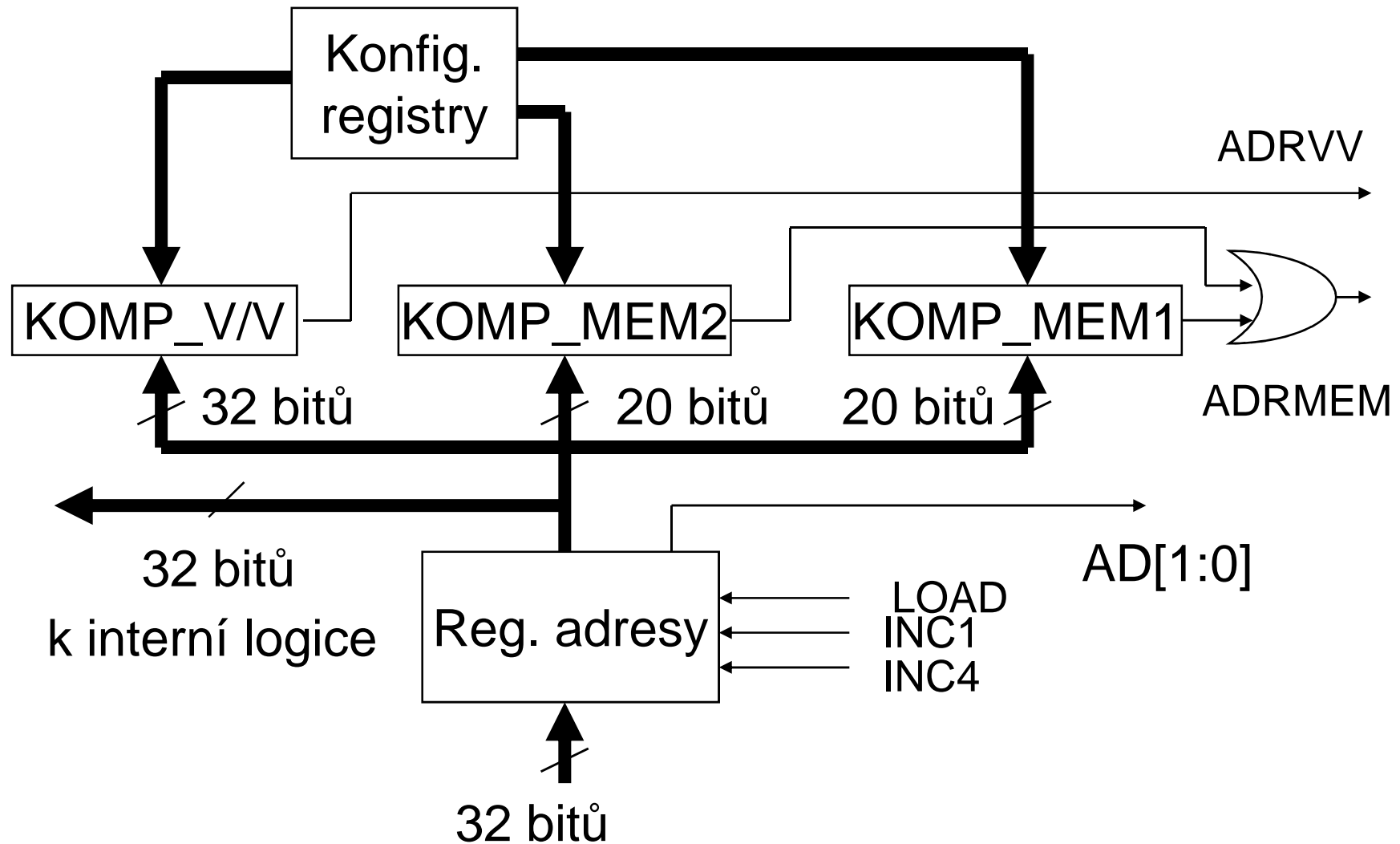
# Adresový dekodér

- Základem je komparátor adres:
  - porovnává adresu na adresové sběrnici a adresu v některém bázovém registru konfiguračního prostoru (pro každý registr jeden komparátor).
- Adresa je na AD sběrnici pouze v první fázi sběrnicevého cyklu  $\Rightarrow$  adresa se musí zapamatovat v *registru adresy*.
- Blokovaný přenos dat s autoinkrementací  $\Rightarrow$  registr adresy bude mít podobu čítače s *paralelním přednastavením (LOAD)*.
- Relokovatelný adresový dekodér? Zrcadlení?

# Příklad řešení



# Adresový dekodér



- **ADRVV**
  - rovnost adresy V/V prostoru.
- **ADRMEM**
  - rovnost některé adresy paměťového prostoru.
- Registr adresy je synchronní čítač s paralelním přednastavením (synchronním),
  - LOAD - synchronní zápis adresy,
  - INC1 - zvýšení hodnoty o 1,
  - INC4 - zvýšení hodnoty o 4.
- AD[1:0] - k interní logice - informuje o typu burst módu.

## Další bloky

- Konfigurační prostor
  - registrové pole o velikosti 256 B pro čtení/zápis,
- kontrola parity dat (generování signálu PERR#),
- kontrola chyb, např. přetečení čítače adres při souvislém přenosu dat (generování signálu SERR#).

## *Poznámka:*

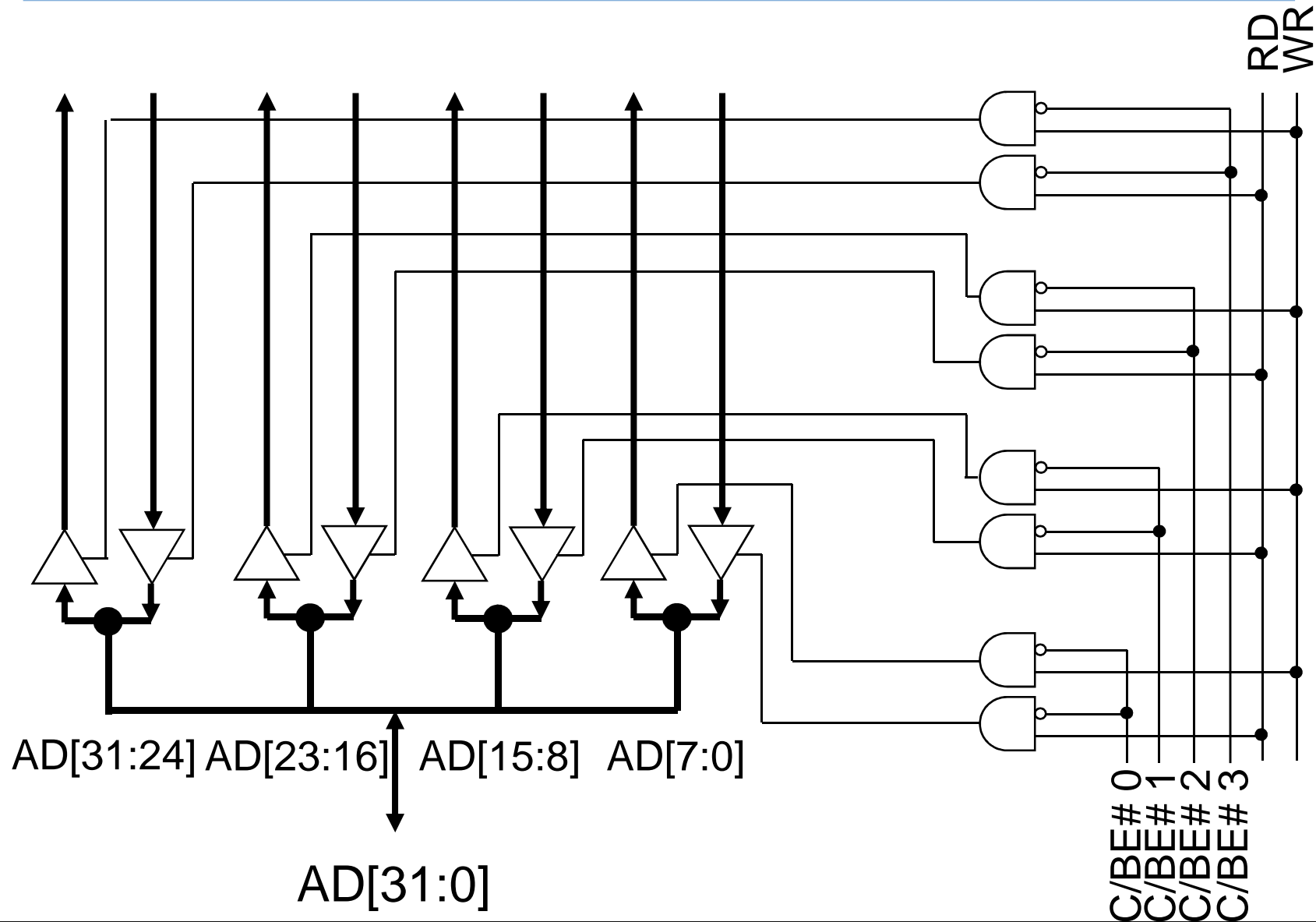
- Protože musí být při čtení/zápisu z/do paměti adresa vždy dělitelná 4,
- ale při V/V operaci ne,
- musí čítač zvyšovat hodnotu o 1 nebo o 4.

# Datové cesty a jejich řízení

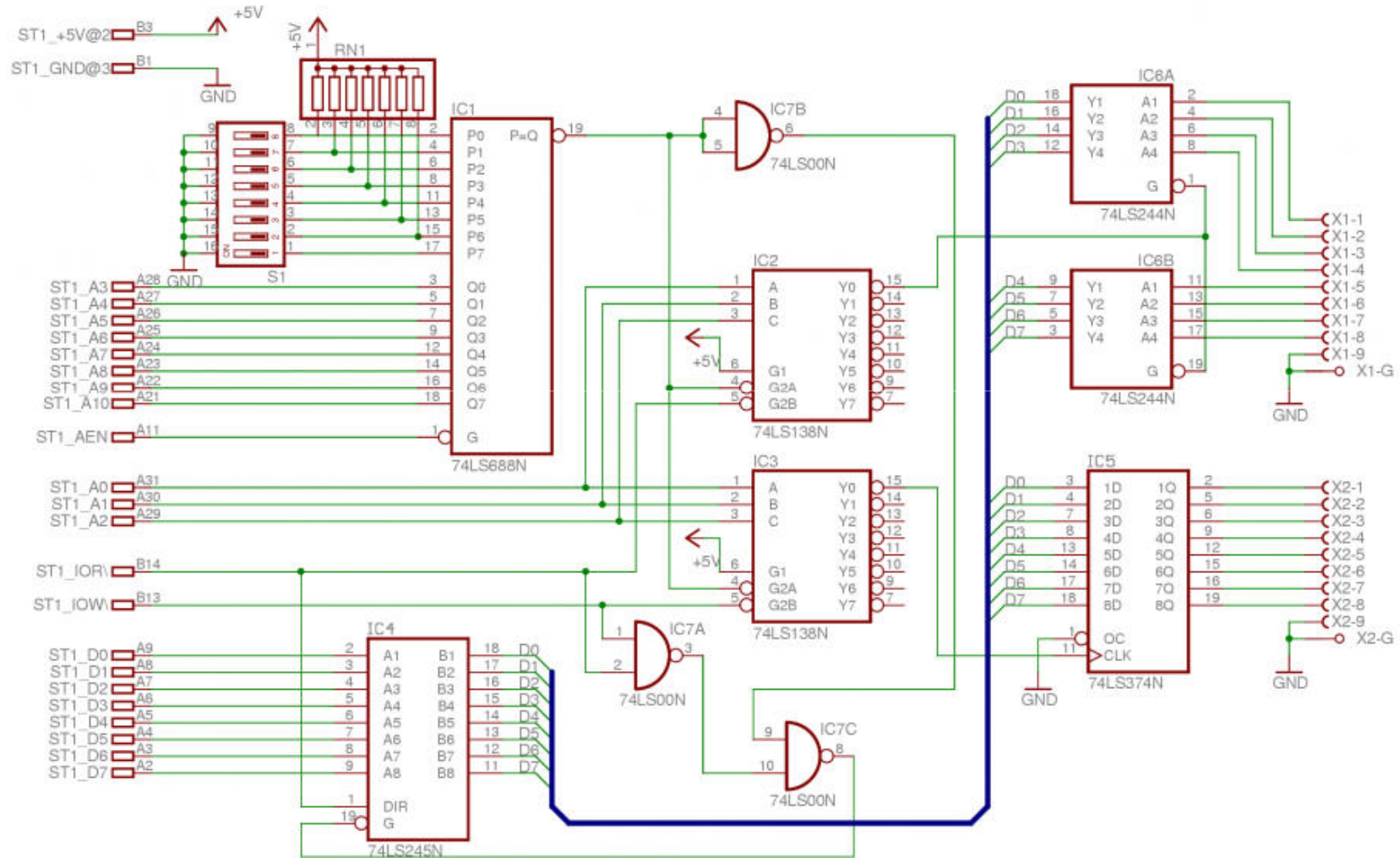
- Obousměrná datová sběrnice  $\Rightarrow$  rozhraní budou tvořit obousměrné třístavové budiče.
- Datové přenosy 8, 16, 32 bitů  $\Rightarrow$  řízení směru a třetího stavu bude odvozeno od typu příkazu (čtení/zápis) a od **masky platnosti dat C/BE#** (po osmi bitech).



# Datové cesty



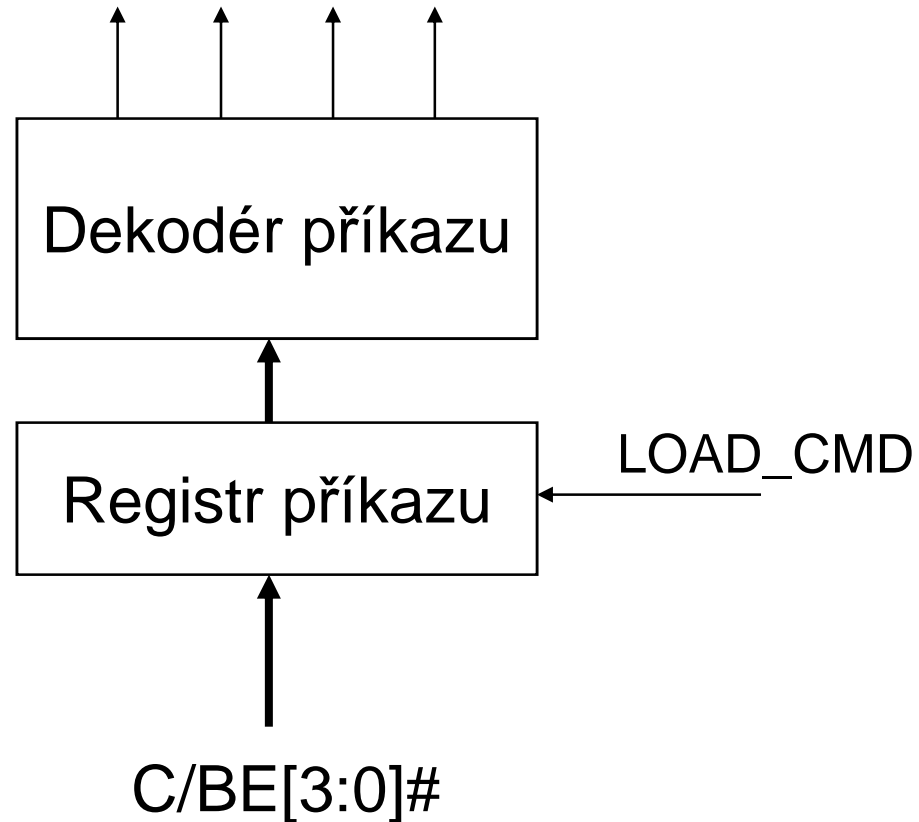
# Příklad řešení: brány včetně dekodéru



# Dekodér příkazu

- Příkaz zachycen do registru příkazu, vlastní dekodér je kombinační obvod.
- Výstup - řídicí signály, které např. určí:
  - směr transakce (čtení/zápis),
  - typ transakce:
    - operace s V/V prostorem, operace s paměťovým prostorem, konfigurační cyklus, žádost o přerušení).
- Usnadní návrh řídicí logiky
  - srovnej s operačním kódem a řízením aritmetických operací, viz přednášku „Procesor“.

# Dekodér příkazu



# Význam bitů C/BE[3..0]#

C/BE[3..0]#	Typ příkazu
0000	Potvrzení přerušení (Interrupt Acknowledge)
0001	Speciální cyklus (Special Cycle)
0010	Čtení z portu (I/O Read)
0011	Zápis na port (I/O Write)
0100	Rezervováno (Reserved)
0101	Rezervováno (Reserved)
0110	Čtení z paměti (Memory Read)
0111	Zápis do paměti (Memory Write)
1000	Rezervováno (Reserved)
1001	Rezervováno (Reserved)
1010	Konfigurační čtení (Configuration Read)
1011	Konfigurační zápis (Configuration Write)
1100	Memory Read Multiple
1101	Dual Address Cycle
1110	Memory Read Line
1111	Memory Write and Invalidate

- výstupní signály dekodéru
  - RD - operace čtení,
  - WR - operace zápisu,
  - VV - operace s V/V prostorem,
  - MEM - operace s paměťovým prostorem,
  - CONF - čtení/zápis do/z konfiguračního prostoru,
  - INT - příkaz Interrupt Acknowledge.

# Řídicí automat rozhraní

- Rozpoznává začátek a konec cyklu,
- generuje DEVSEL# a řízení registru adresy, registru příkazu, dekodéru příkazu, detekuje pro vnitřní logiku karty prodloužení cyklu na základě signálu IRDY#.
- Vstupními signály jsou:
  - FRAME#, IRDY#, ADRVV, ADRMEM, MEM, VV.

# Řídicí logika

- Sekvenční obvod, chování popíšeme konečným automatem.

## Otázka:

Budeme navrhovat automat typu Moore nebo Meally nebo je to jedno?

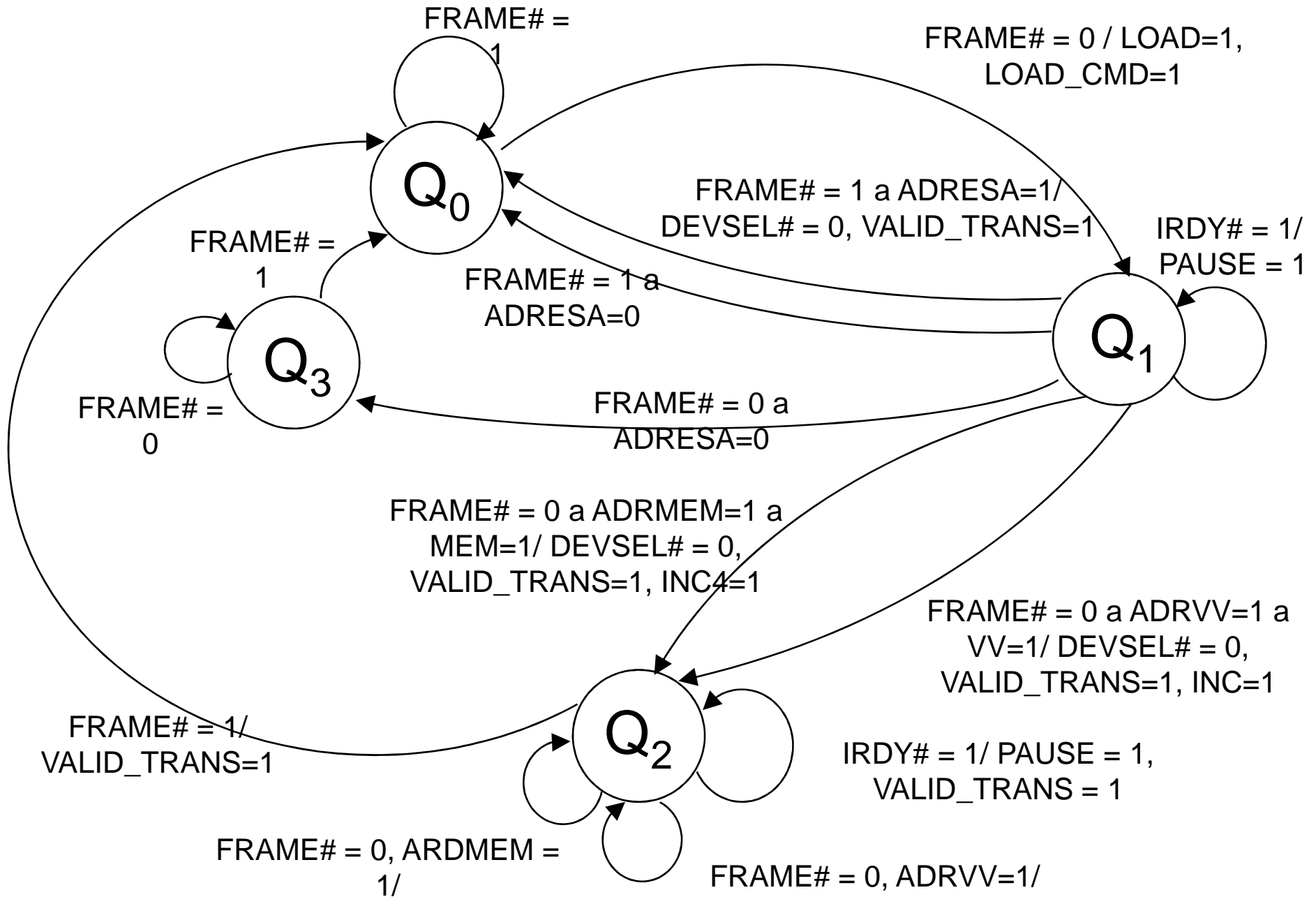


## Odpověď:

Musíme navrhovat Meallyho automat, protože řídicí signály musí být platné ještě před vzestupnou hranou signálu CLK.

- Dohoda
  - všechny interní řídicí signály budou v pozitivní logice.

- Výstupní signály
  - LOAD, LOAD\_CMD, DEVSEL#, VALID\_TRANS, INC, INC4, PAUSE (prodloužení cyklu - pro vnitřní logiku).
- Úmluva:
  - v grafu přechodů jsou zobrazeny jen ty výstupní signály, které jsou aktivní.
  - ADRESA = ((ADRVV=1 a VV=1) nebo (ADRMEM=1 a MEM=1)).

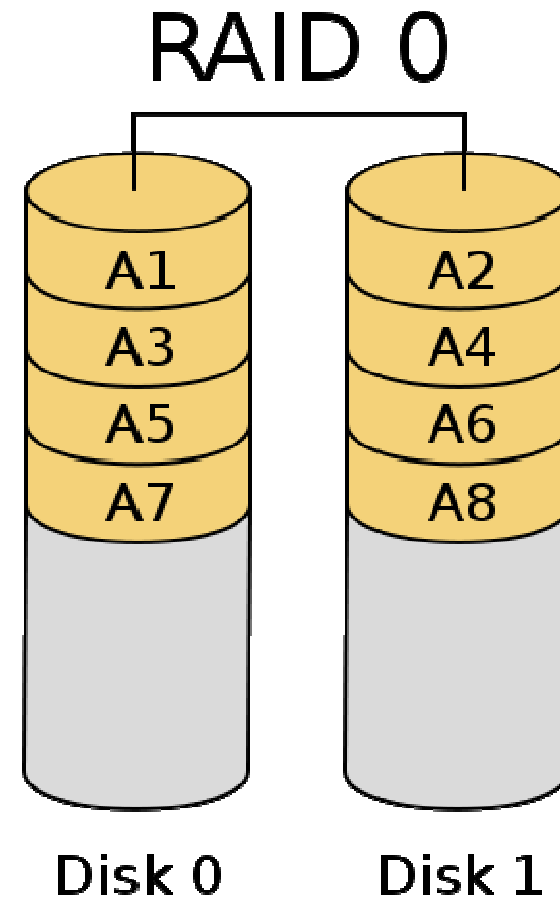


- řízení směru a třetího stavu datových budičů může být odvozeno na základě logického součinu signálu `VALID_TRANS` a výstupu `RD`, resp. `WR`, dekodéru příkazu

Nejdůležitější periferií je disk. Tak ho musíme zrychlit...

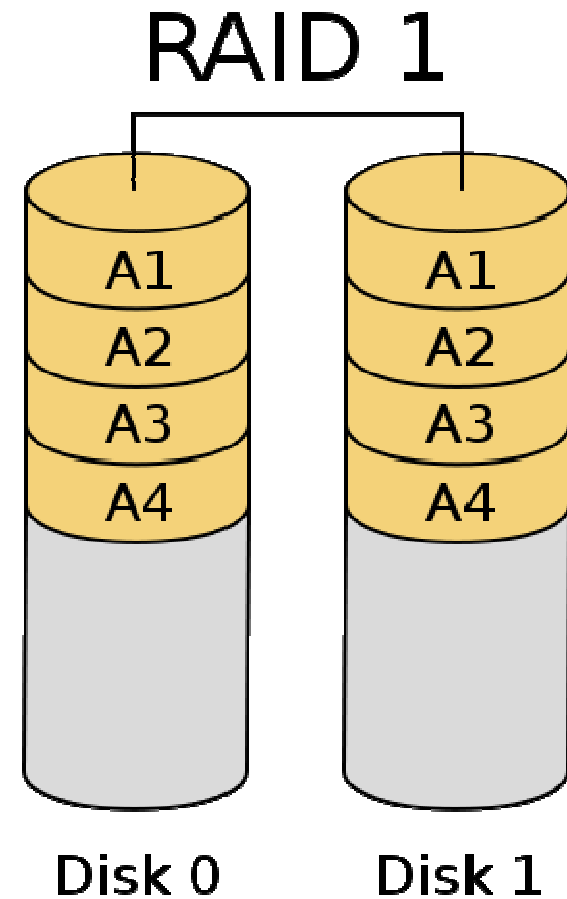
## RAID 0

- Pro zvýšení výkonu systému pevných disků.
- tzv. “stripping” (proužkování)



## RAID 1

- Pro zvýšení spolehlivosti uložených dat.
- Označuje se jako “Mirroring”.
- Nezrychluje, ale zvyšuje spolehlivost.



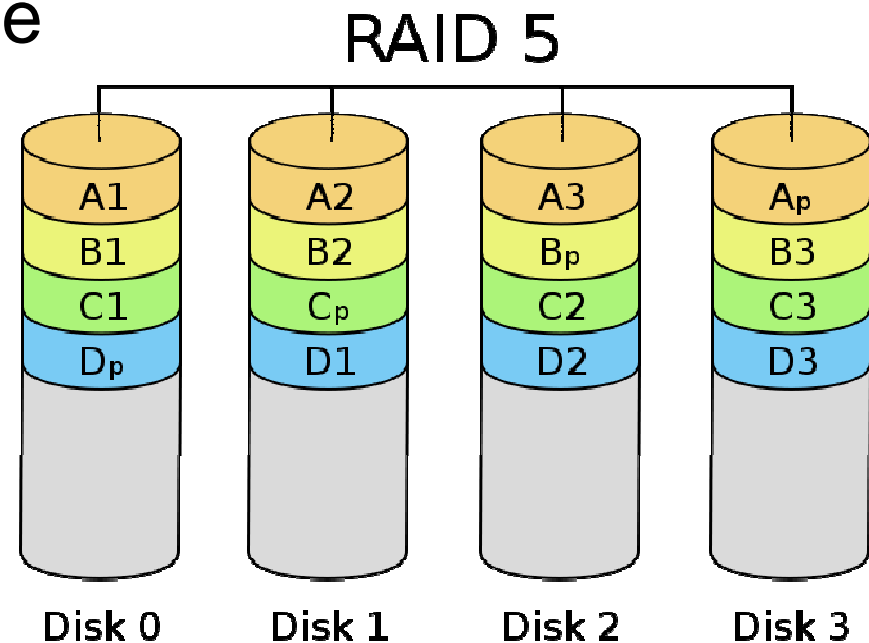
## RAID 10

- Kombinace obou výše popsaných.
- Vytvoří se RAID 0 a ten se pak zrcadlí na RAID 1. Výsledkem jsou vlastně dva RAID 0 obsahující identická data.
- RAID 10 zvyšuje jak výkon, tak spolehlivost, musíte ovšem použít nejméně čtyři disky, nejlépe se stejnými parametry.



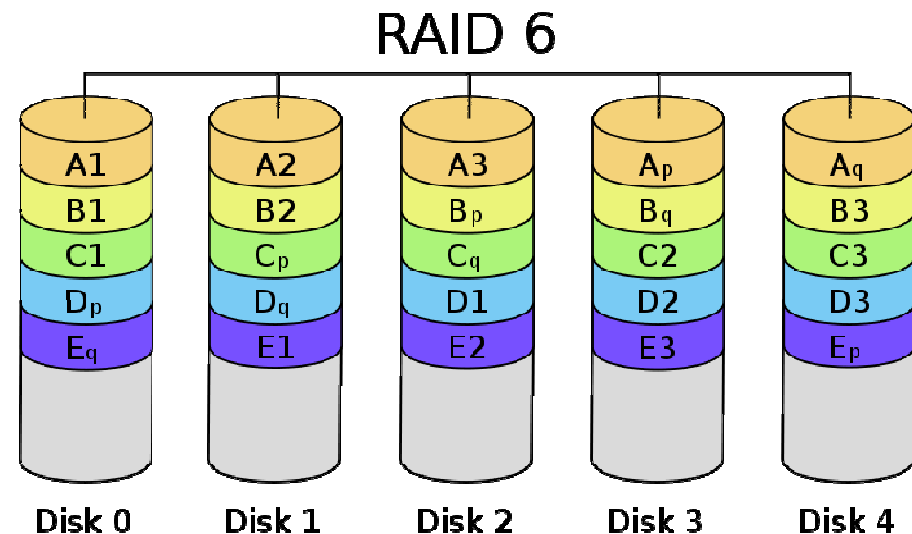
## RAID 5

- Ukládá paritní informace, nikoli však na jeden vyhrazený disk.
- V degradovaném režimu se musejí data uložená na vadném disku odvodit z dat zbývajících disků a parity.
- Zrychluje čtení, zpomaluje zápis.



## RAID 6

- Obdoba RAID 5, používá dva paritní disky s různě vypočtenou paritou.
- Odolný proti výpadkům 2 disků.
- Rychlost čtení jako RAID 5, zápis ještě pomalejší.



- RD, WR
  - signály z vnitřní logiky (čtení/zápis), odvozují se od typu příkazu
- C/BE#
  - maska platnosti dat (vnější vstupní signály).