

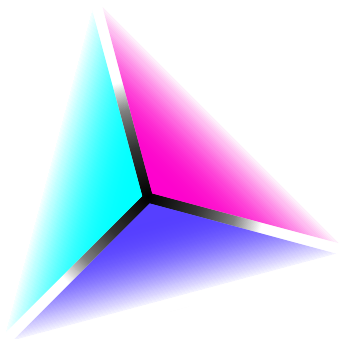
Struktura a architektura počítačů

Katedra číslicového návrhu
Fakulta informačních technologií
České vysoké učení technické

© Hana Kubátová, 2021

Řadič, pipelining

BI-SAP, květen 2021



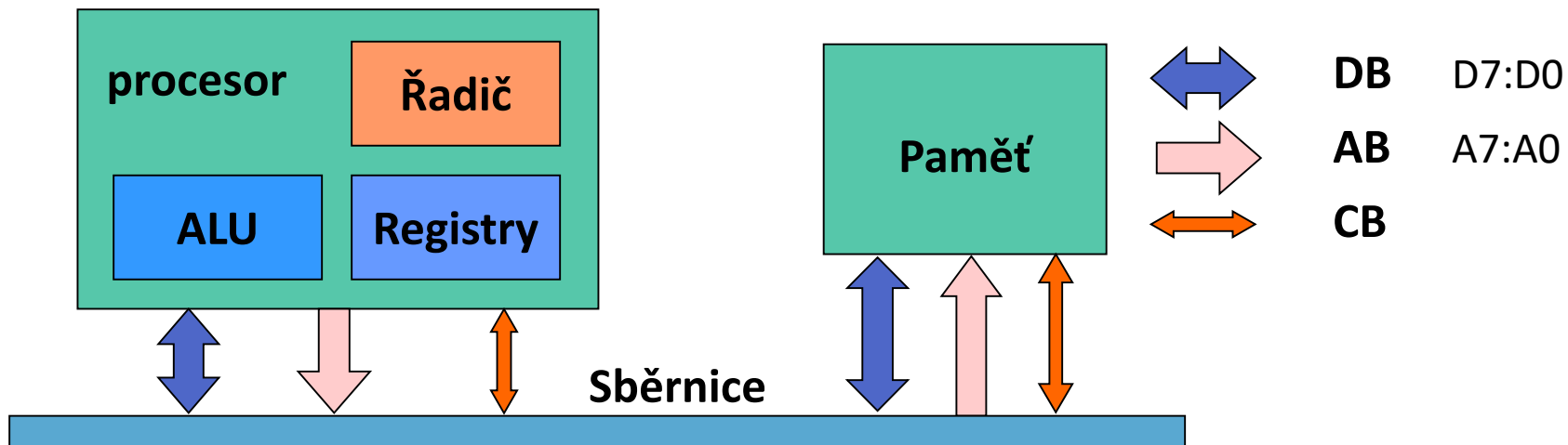
Obsah

- Řídící jednotka procesoru – řadič počítače
- Pipelining, procesory typu RISC a CISC
- Příklad návrhu výkonné a řídící jednotky: přednáška ASI Centrum

HW počítače - shrnutí

- **Všechny počítače se skládají z 5 základních částí:**
 - Datová část (ALU) - v procesoru
 - **Řídící část (řadič) – v procesoru**
 - Hlavní paměť – často mimo procesor
 - Vstupní zařízení
 - Výstupní zařízení
- **Paměťový systém**
 - Registry
 - Caches: rychlé, dražší, kapacitně menší, umísťované blíž k procesoru
 - Hlavní paměť: pomalejší, levnější, větší
 - Vnější paměť: ještě pomalejší, ale velká kapacita
 - Záložní paměť: CD, DVD, flash, magnetická páska
- **Vstupní a výstupní zařízení ... HW a aplikačně závislé**

Procesor: návrh řadiče

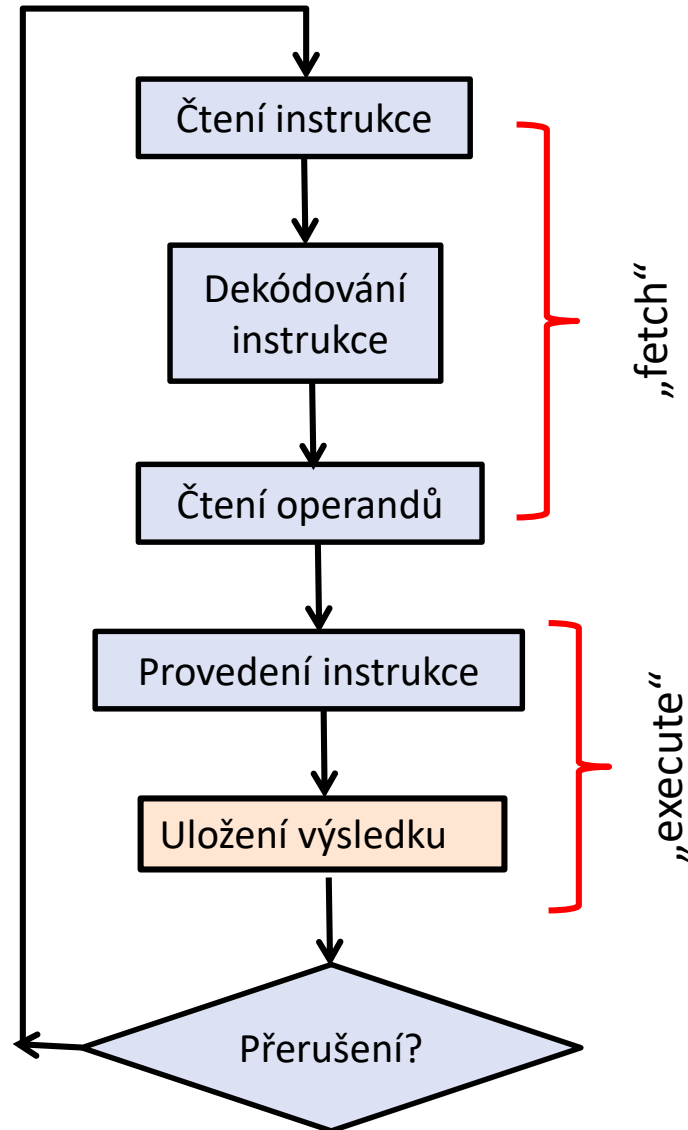


CPU (procesor) musí obsahovat:

- **ALU** výkonnou jednotku (kombinační obvody)
- **řadič** ... řídicí jednotku (**control unit**, sekvenční obvod)
- **registry** pro dočasné uchování dat

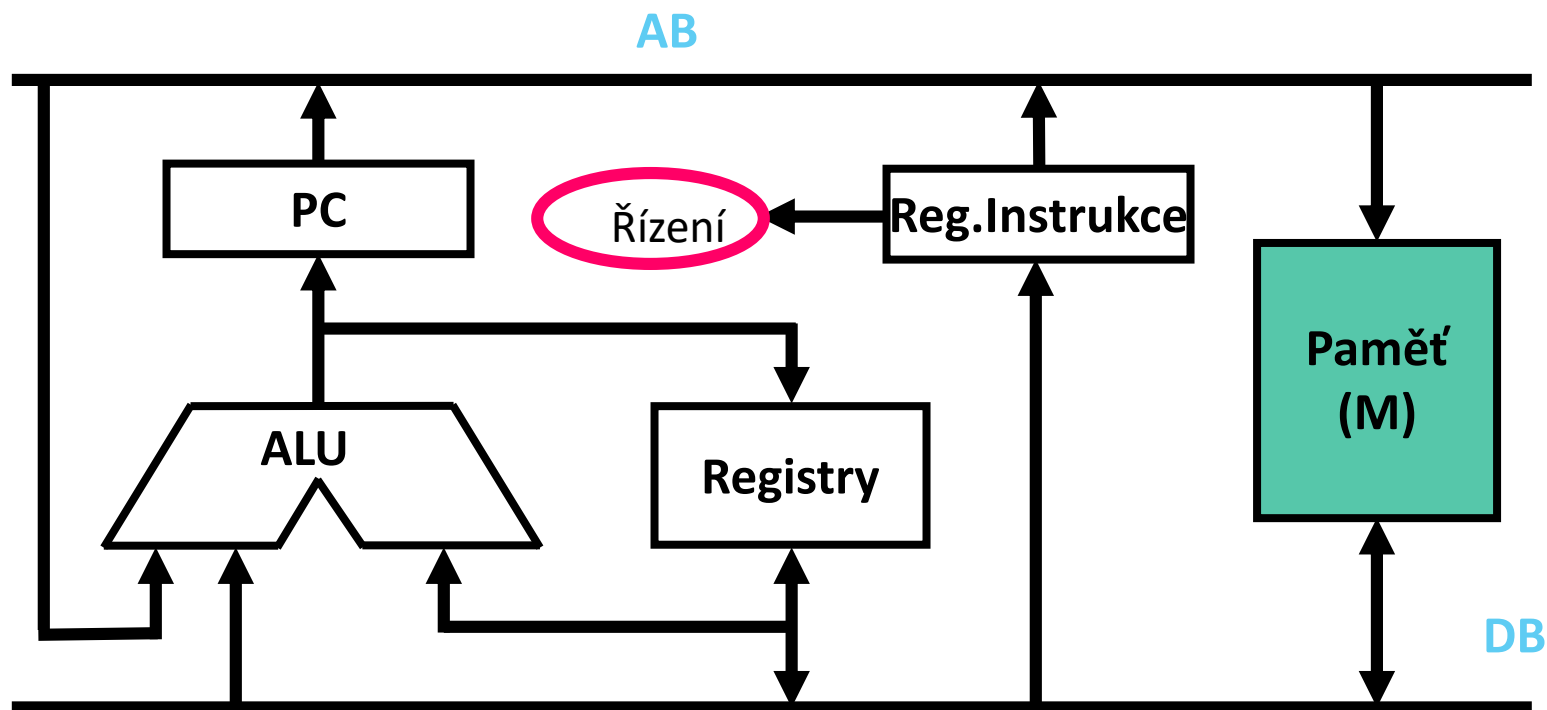
Činnost řadiče je popsána instrukčním cyklem pro všechny instrukce, které procesor umí. Při dekódování instrukce (i jen částečném) dochází k větvení.

Instrukční cyklus



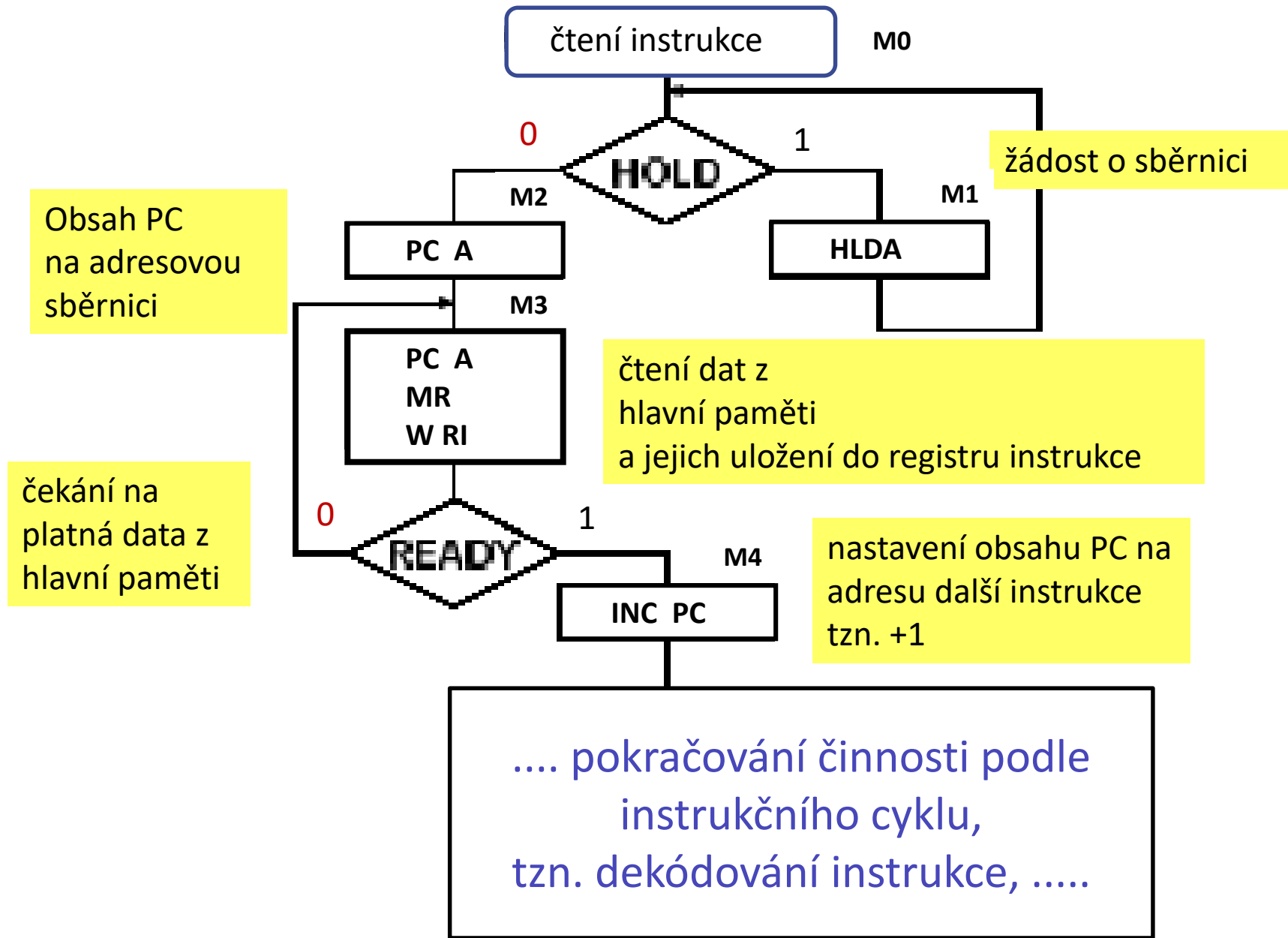
Řadič procesoru

- Složitější řídicí automat
- Pracuje podle instrukčního cyklu
- Řídí činnost všech výkonných jednotek počítače podle instrukcí a jejich (strojového) kódu, podle **instrukčního cyklu**
- Je to sekvenční obvod – závisí na sekvenci **vstupních (stavových)** signálů, které generují výkonné jednotky (ALU, HP - instrukce) a vysílá jim **řídicí signály (výstupy** řadiče)
- Pracuje v **nekonečném cyklu** – řídí zpracování instrukcí
- Navrhuje se podle instrukčního cyklu a výběru ISA (tzn. z grafu přechodů, vývojového digramu)
- Lze ho realizovat **obvodově** (klasicky) nebo **mikroprogramově**

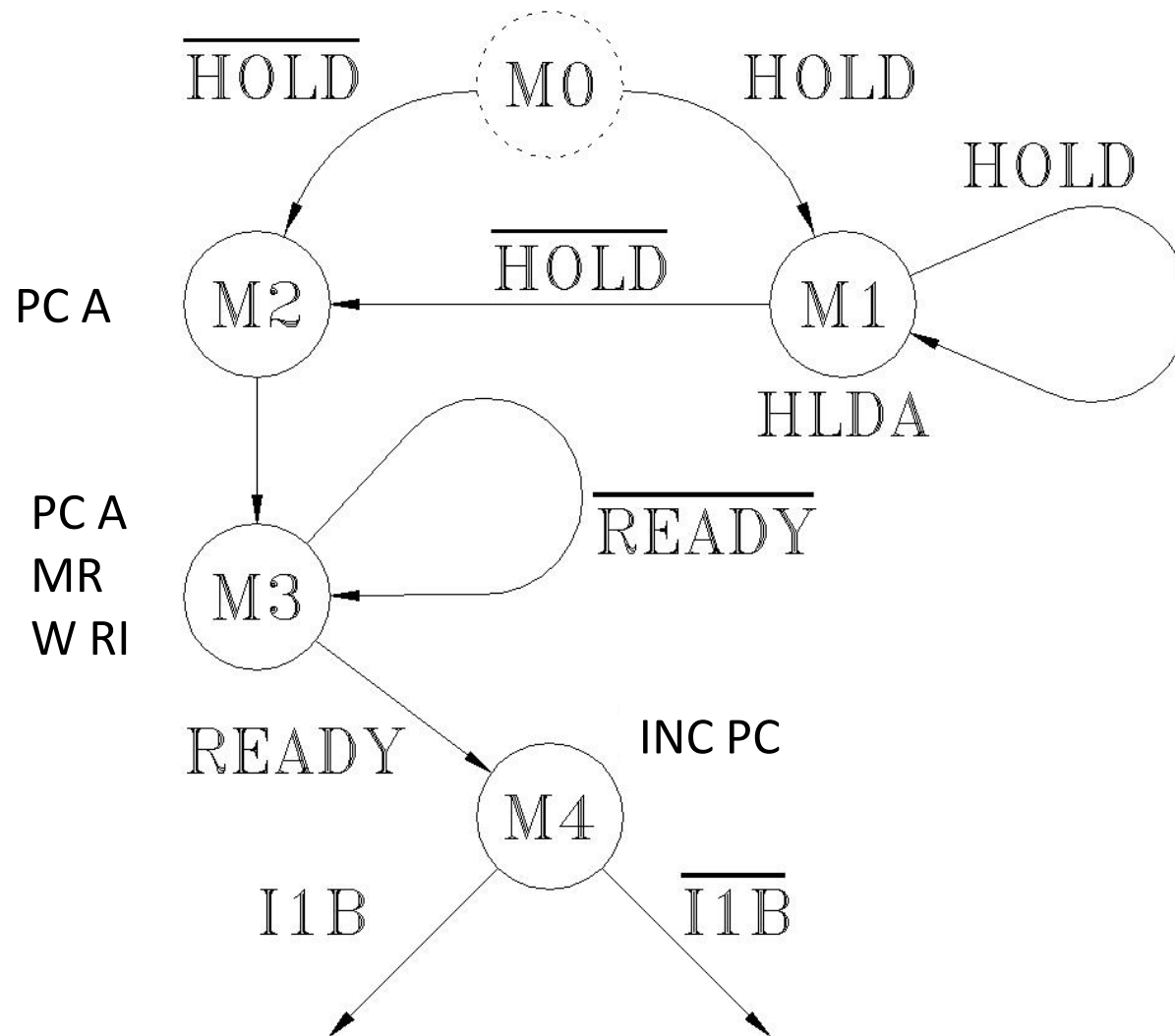


Řízení některé akce mohou/musí proběhnout paralelně.

- nahraj data do registrů - WPC, WRI, Wreg_i
- output enable registrů – OEPC, OReg_i
- inkrement/dekrement registru – IncPC (SP)
- výběr operace ALU – ADD, SUB, CMP,
- čti – zapiš do paměti WM, RM, RESET ... PC = 0

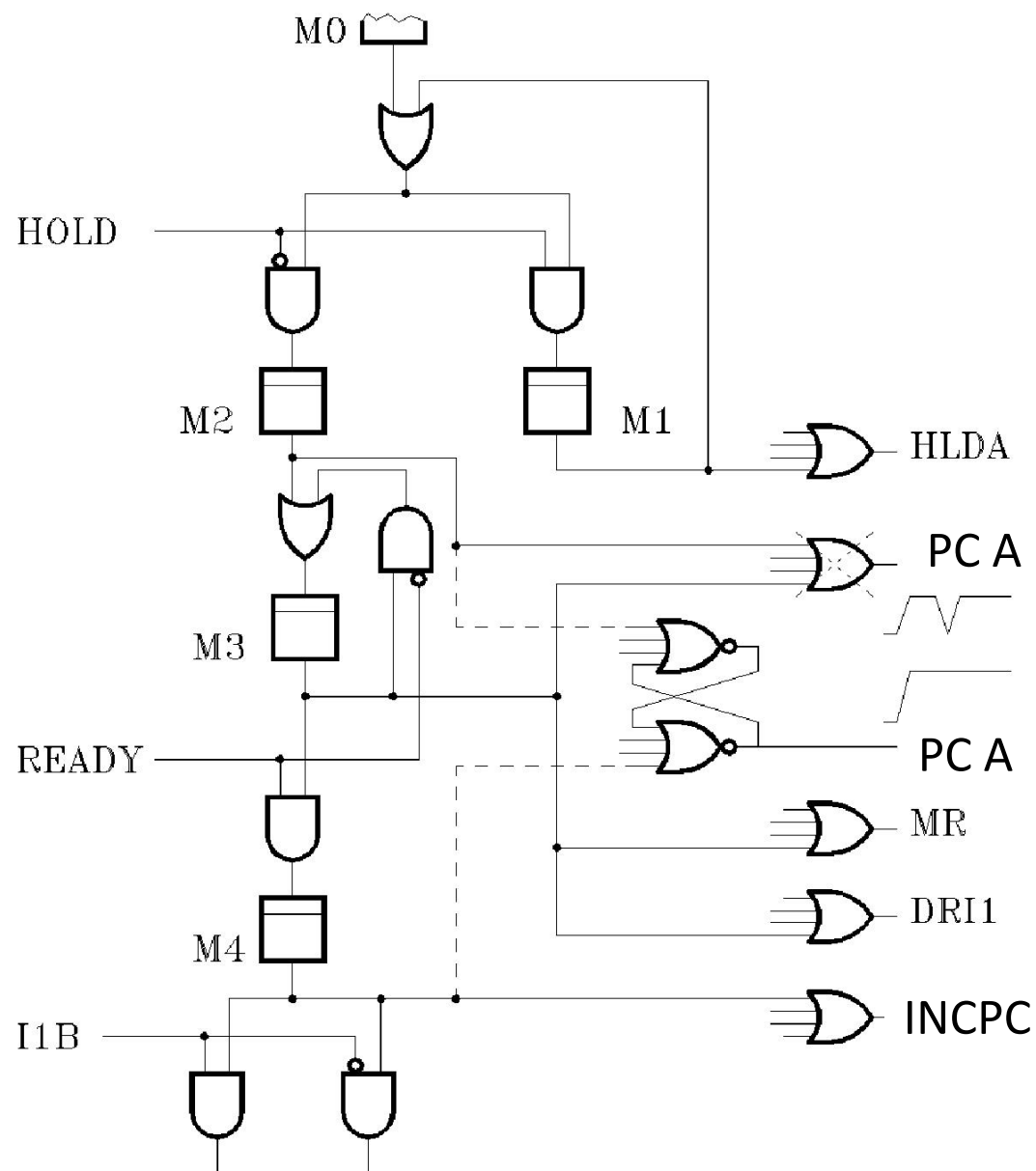


Totéž jako graf přechodů

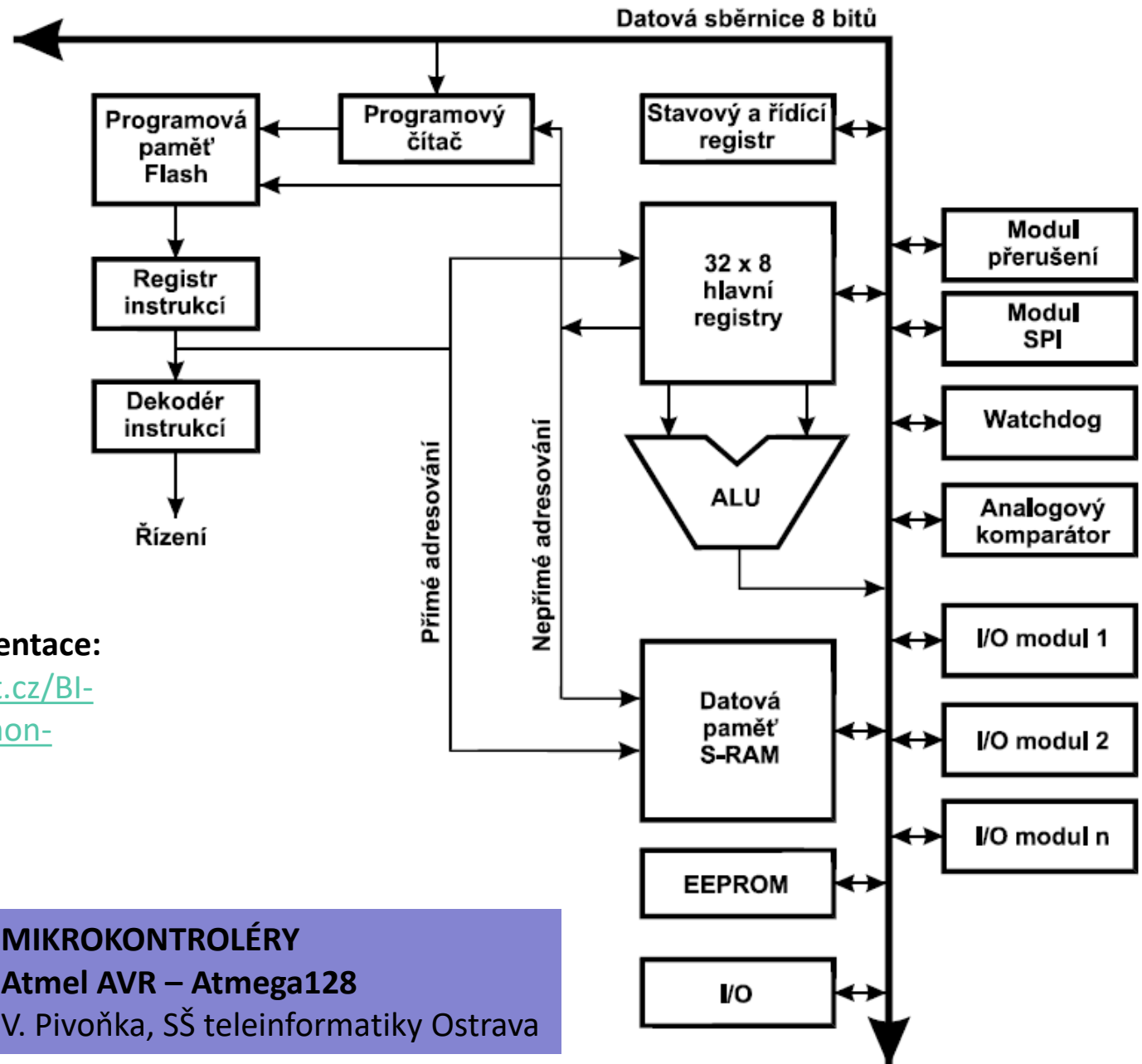


Obvodový řadič

totéž realizované
jako obvodový
řadič v kódu 1 z N:
tzn. každému
stavu M_i odpovídá
jeden D-KO



AVR-data



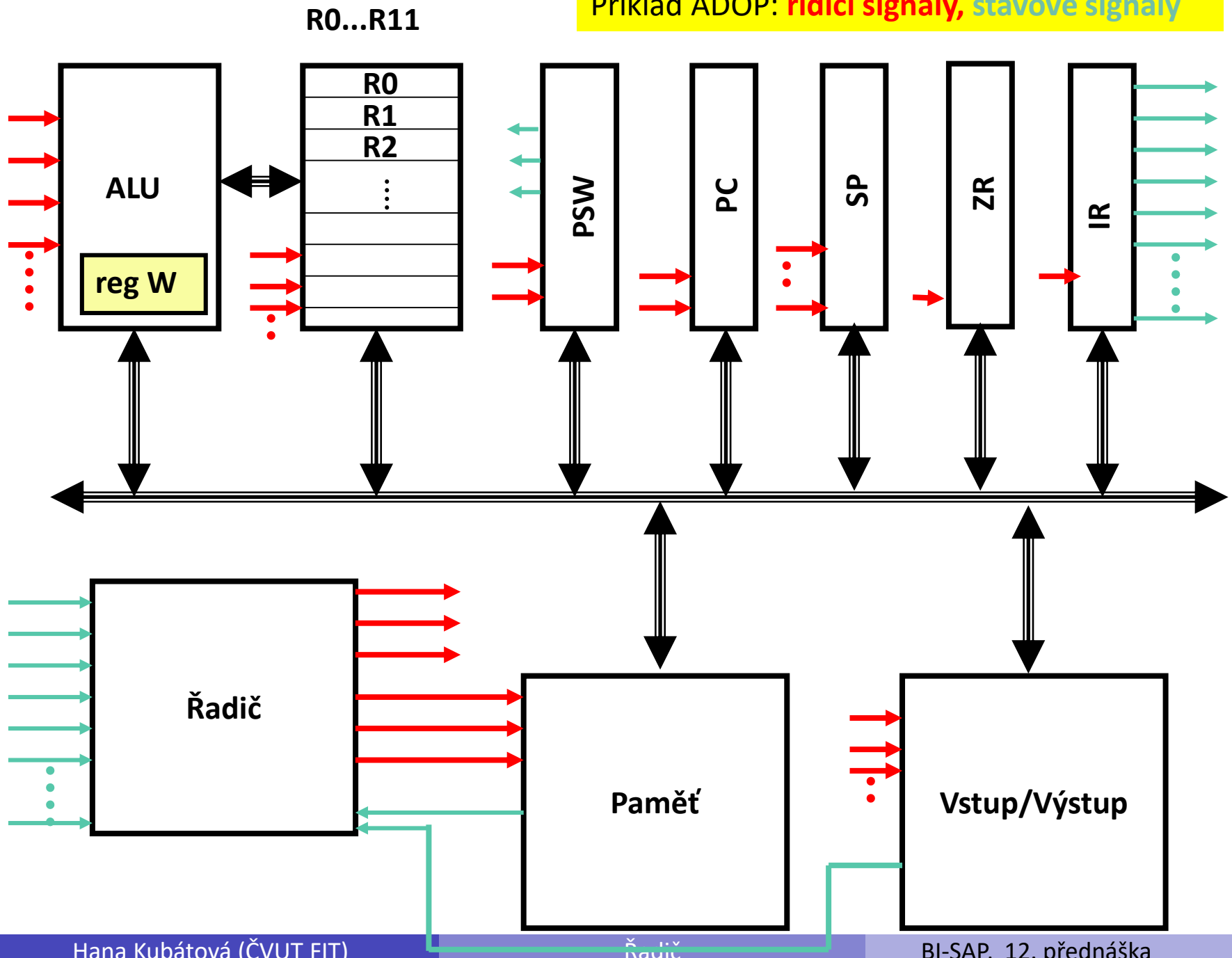
Podrobněji viz dokumentace:

<https://courses.fit.cvut.cz/BI-SAP/media/labs/common-avr/atmega169vl.pdf>

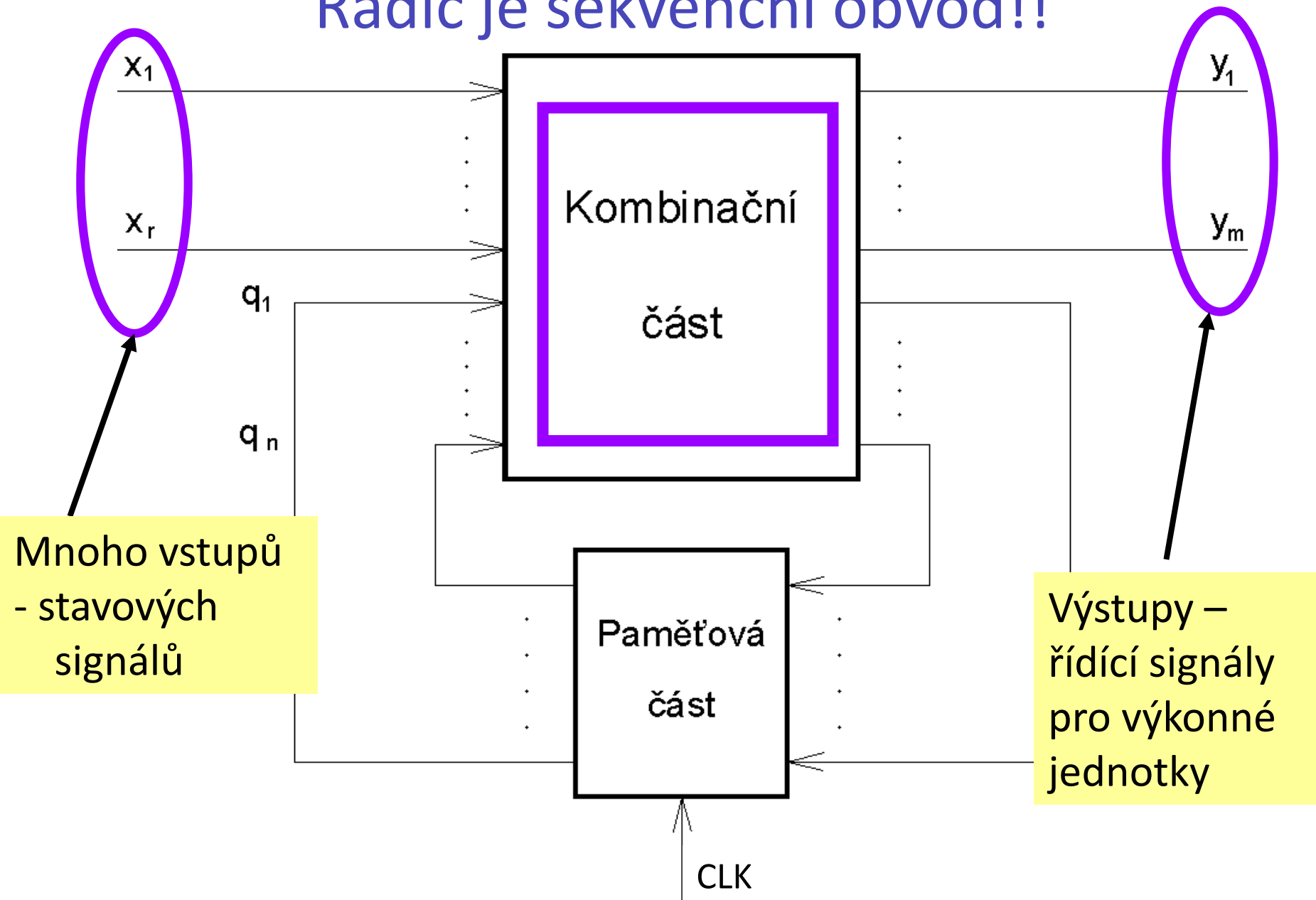
MIKROKONTROLÉRY

Atmel AVR – Atmega128

V. Pivoňka, SŠ teleinformatiky Ostrava

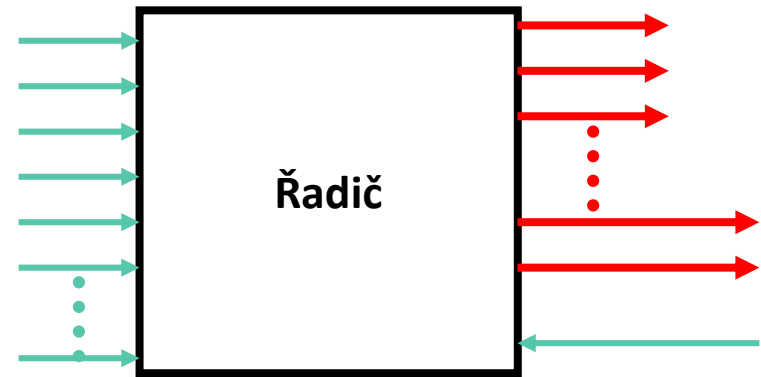


Řadič je sekvenční obvod!!



Realizace řadiče

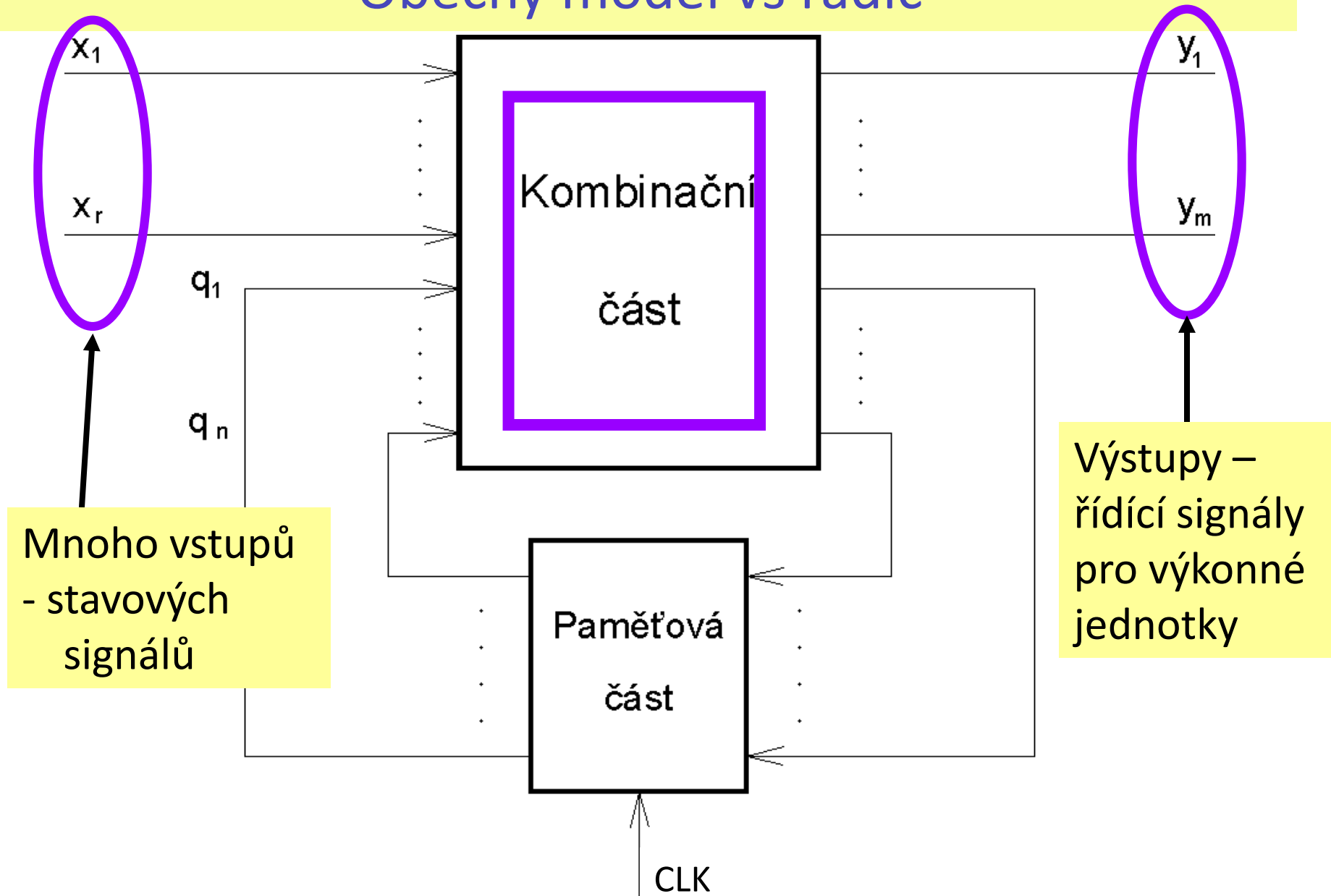
- jde o sekvenční obvod
- mnoho vstupů
- mnoho výstupů
- ale v daném taktu (stavu)
 - se uplatní jen málo z nich, tzn. jeden nebo žádný
- většinou se pokračuje následujícím stavem
 - jde o popis algoritmu tedy činností v instrukčním cyklu,
lze použít vývojový diagram



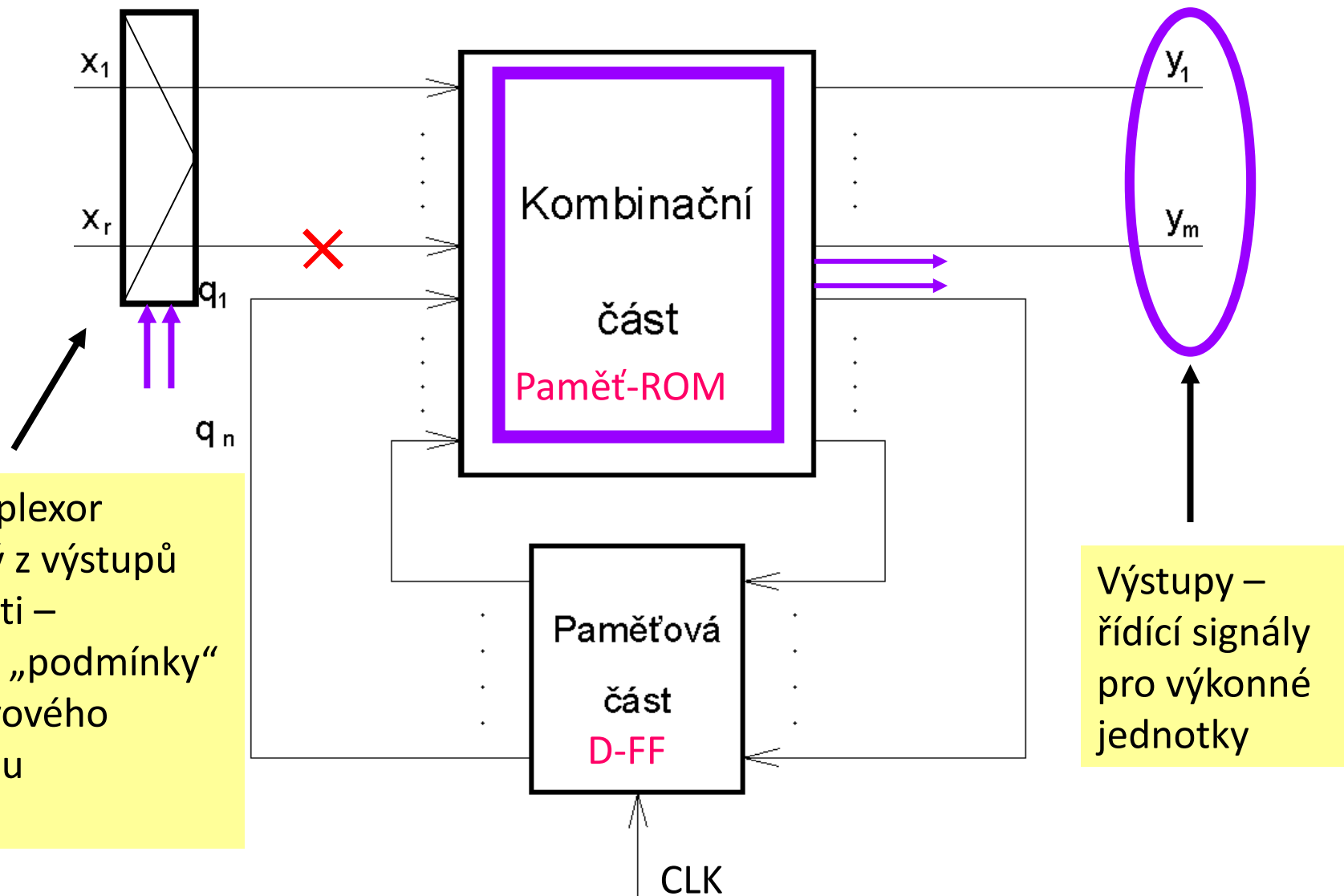
Realizace řadiče

- Podle způsobu jeho realizace existuje tzv. **obvodový** (klasický) řadič a **mikroprogramový** řadič
- **Obvodový řadič** – návrh klasického sekvenčního obvodu ... usnadnění ... kód vnitřních stavů 1 z n – pak lze návrh provést z vývojových diagramů popisující činnost procesoru při provádění instrukcí podle instrukčního cyklu
- **Mikroprogramový řadič** –
 - Sekvenční obvod s kombinační částí realizovanou pamětí (nazývá se *řídící paměť*, *paměť mikroprogramů*, „*control memory*“)
 - Jednotlivé dílčí operace, které se provádějí při zpracování instrukcí jsou uloženy v této paměti a říká se jim **mikroinstrukce**
 - Soubor mikroinstrukcí tvoří mikroprogram, soubor všech mikroprogramů je mikroprogramové vybavení - **firmware**

Obecný model vs řadič

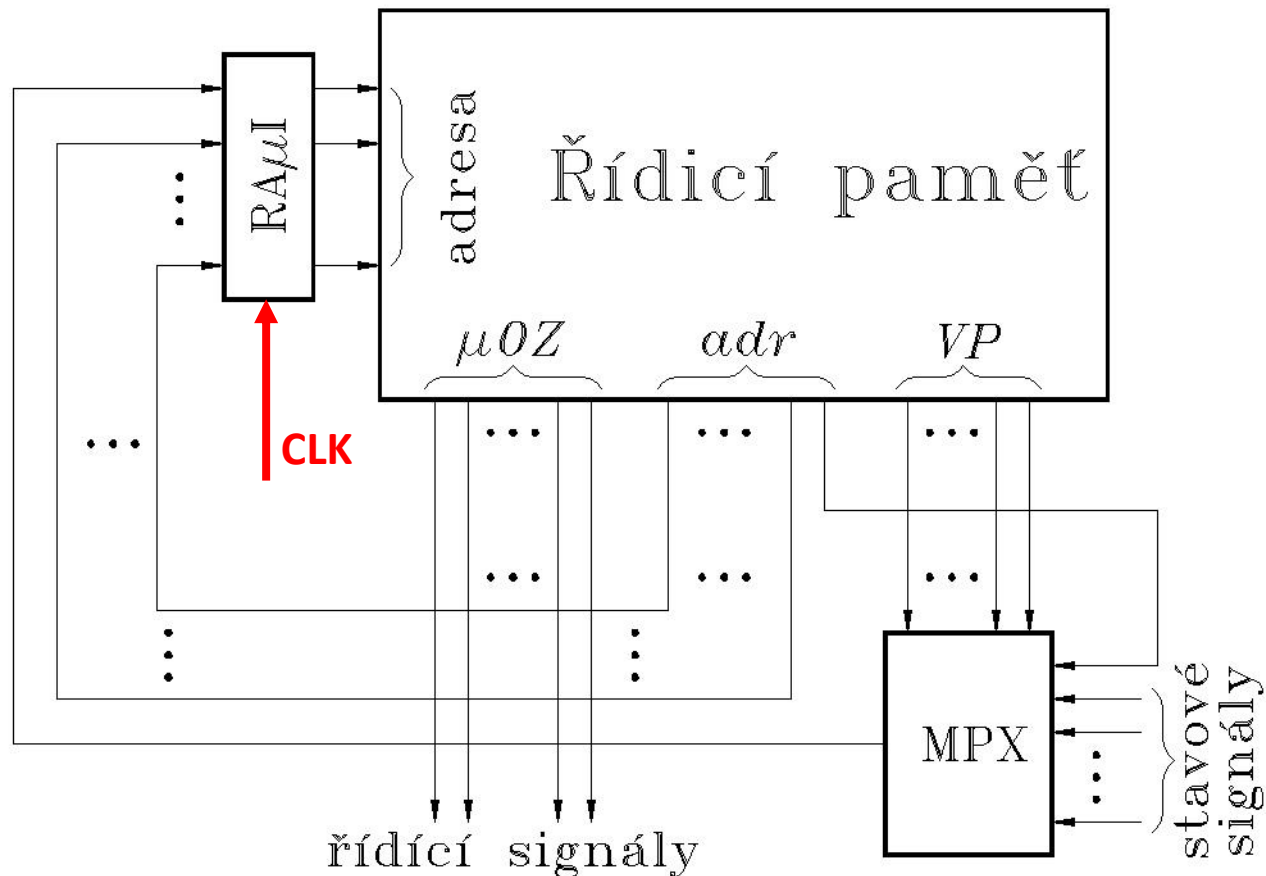


Mnoho vstupů (stavových signálů) ... ale rozhoduje se podle nich jen někdy a jen podle některých:



Mikroprogramový řadič

Moorův automat



Mikroprogramový řadič

- jsou možné další úpravy a vylepšení podle souboru instrukcí a výkonných jednotek
- čítač adres mikroinstrukcí, protože většinou se pokračuje následující
- např. pro často se opakující části instrukcí zavést možnost podmikroprogramů a HW zásobník jako součást řadiče a čítač taktů s možností přednastavení pro počet opakování cyklů mikroinstrukcí
- Horizontální a vertikální mikroprogramování

Terminologie

program se skládá z instrukcí

instrukce ... se provádí ve několika taktech a skládá se z mikroinstrukcí

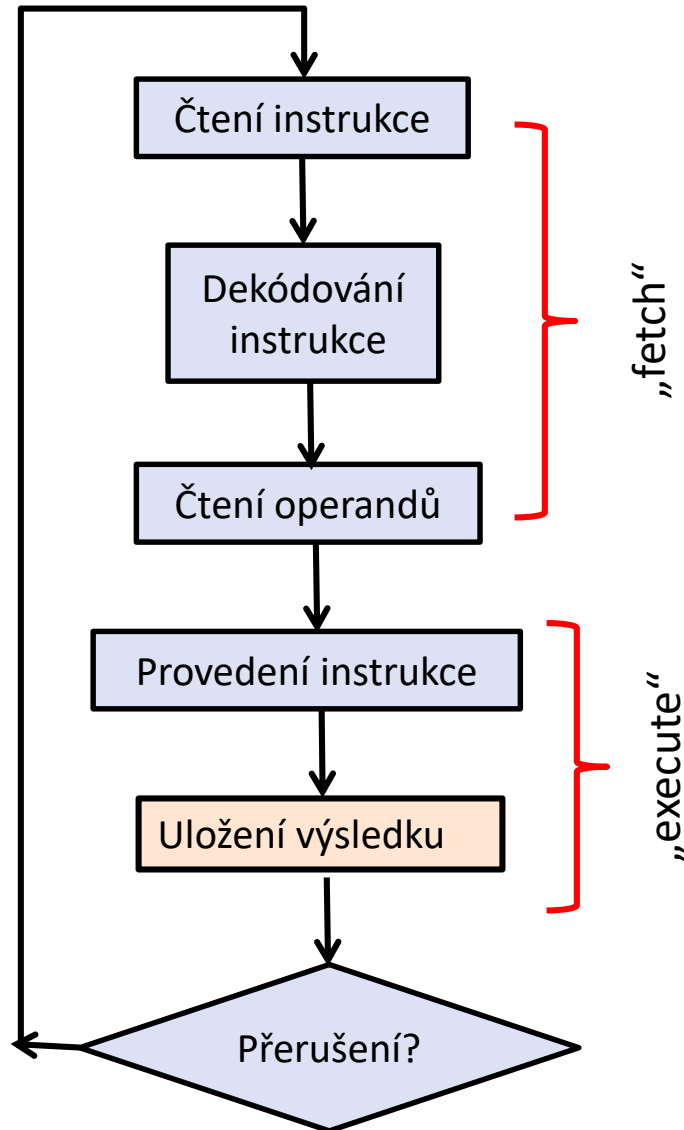
mikroinstrukce ... okamžitý stav procesoru skládá se:

- z řídících signálů pro výkonné jednotky
- určení následného stavu (kde se bude pokračovat)
- volby vstupů, které jsou v příštím taktu významné

firmware ... soubor všech mikroprogramů, tzn. popisů činností každé instrukce v taktech, též mikroprogramové vybavení

Pipelining

- urychlení:



Proudové zpracování instrukcí

Pipelining = princip výrobního pásu, zpracování instrukce po částech, v čase paralelně

- každá jednotka provede část operace
- jednotky pracují současně

jednotka pro:

instrukce

| | | | | | | |
|--------------------|---|---|---|---|---|-------|
| čtení instrukce: | 1 | 2 | 3 | 4 | 5 | 6 |
| dekódování OZ: | | 1 | 2 | 3 | 4 | 5 |
| čtení operandů: | | | 1 | 2 | 3 | 4 |
| provedení operace: | | | | 1 | 2 | 3 |
| uložení výsledku: | | | | | 1 | 2 |
| | | | | | | čas → |

...pipelining

- triviální případ - předčítání instrukcí, jedna instrukce se čte, další dekoduje, provádí ...
- v ideálním případě je v každém taktu dokončena jedna instrukce

konflikty:

- datový - potřebná data dosud nejsou uložena
- skokový - adresu skoku zatím nelze určit

řešení ... počkat (to nejjednodušší, ale ne jediné), predikce skoků, ... vždy nějaký HW navíc + algoritmus

Počítače typu RISC

Reduced Instruction Set Computers

- jak navrhnout rychlý procesor? Co nejvíce výkonných instrukcí?
- statistika .. co nejefektivnější mají být ty nejpoužívanější (přesuny, skoky, srovnání, ...), výkonné se používají málo ... vyplývá z Amdahlova zákona:

Výpočet výkonového zisku, tedy vylepšení celkového zrychlení počítače podle dílčích vylepšení. Podle těchto výpočtů vychází, že je lepší zefektivnit **nejčastěji** prováděné instrukce.

Složité instrukce ... chyby v mikroprogramech, instrukce mají různou délku (zpomalení dekodování, problémy s pipeliningem)

Počítače typu RISC - charakteristika

- malý počet jednoduchých instrukcí (<128)
- krátká doba provedení instrukce - dokončení v jednom taktu
- pipelining
- obvodový řadič
- malý počet formátů instrukcí (≤ 4)
- malý počet způsobů adresace (≤ 4)
- velký počet registrů (> 32)
- komunikace s pamětí pouze instrukcí „přesun“

protipól CISC - Complex Instruction Set Computers
počítače s rozsáhlým souborem instrukcí

ASIC

- Application Specific Integrated Circuit:
 - obvod vyvíjený na zakázku pro jednu konkrétní aplikaci jednoho zákazníka
 - vyplatí se pro sériovou výrobu
 - vysoká cena vývoje
 - dlouhá doba „time-to-market“
- Ale typicky malá velikost, nízký příkon

Příklad

Návrh jednoduché výkonné jednotky a jejího řízení

Příklad: Navrhněte číslicový obvod pro násobení dvou čtyřbitových nezáporných čísel včetně řízení

- výkonná jednotka ... sčítačka, obvody pro posuv
- řadič ... generování signálů pro zápis do registrů, posuv a okamžik sčítání
- registry pro násobenec, násobitel, dočasný výsledek a celkový výsledek