



Univerzita Hradec Králové  
Fakulta informatiky a managementu



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Pohled do vnitra mikroprocesoru

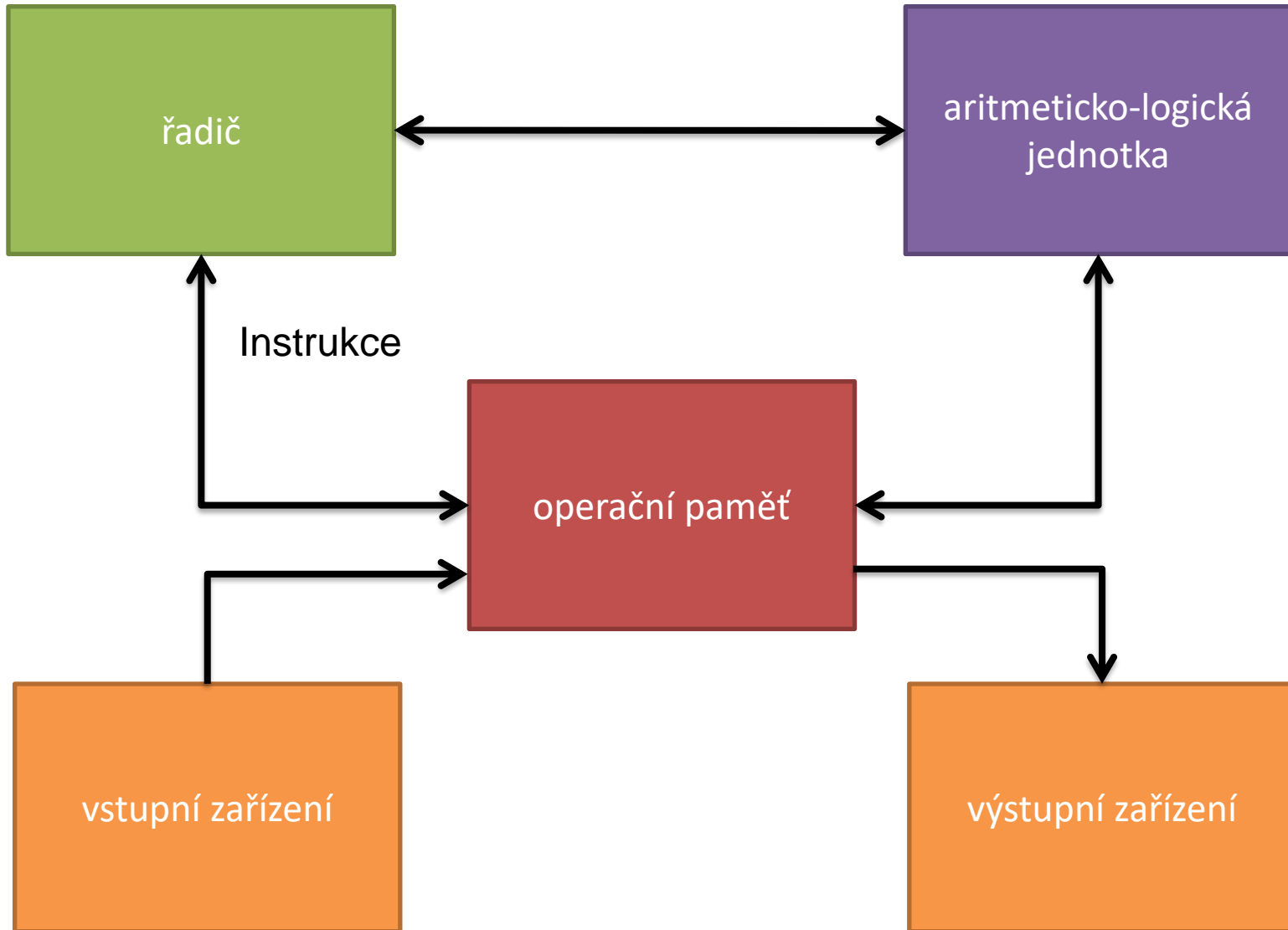
Josef Horálek  
doplnil Peter Mikulecký

- = Vycházíme z von Neumannovy architektury
- = Celý počítač se dle ní skládá z pěti koncepčních bloků:
  - = Operační paměť
  - = Programový řadič
  - = Aritmeticko-logická jednotka (ALU)
  - = Vstupní a výstupní zařízení



FIM UHK

## Von Neumannova architektura

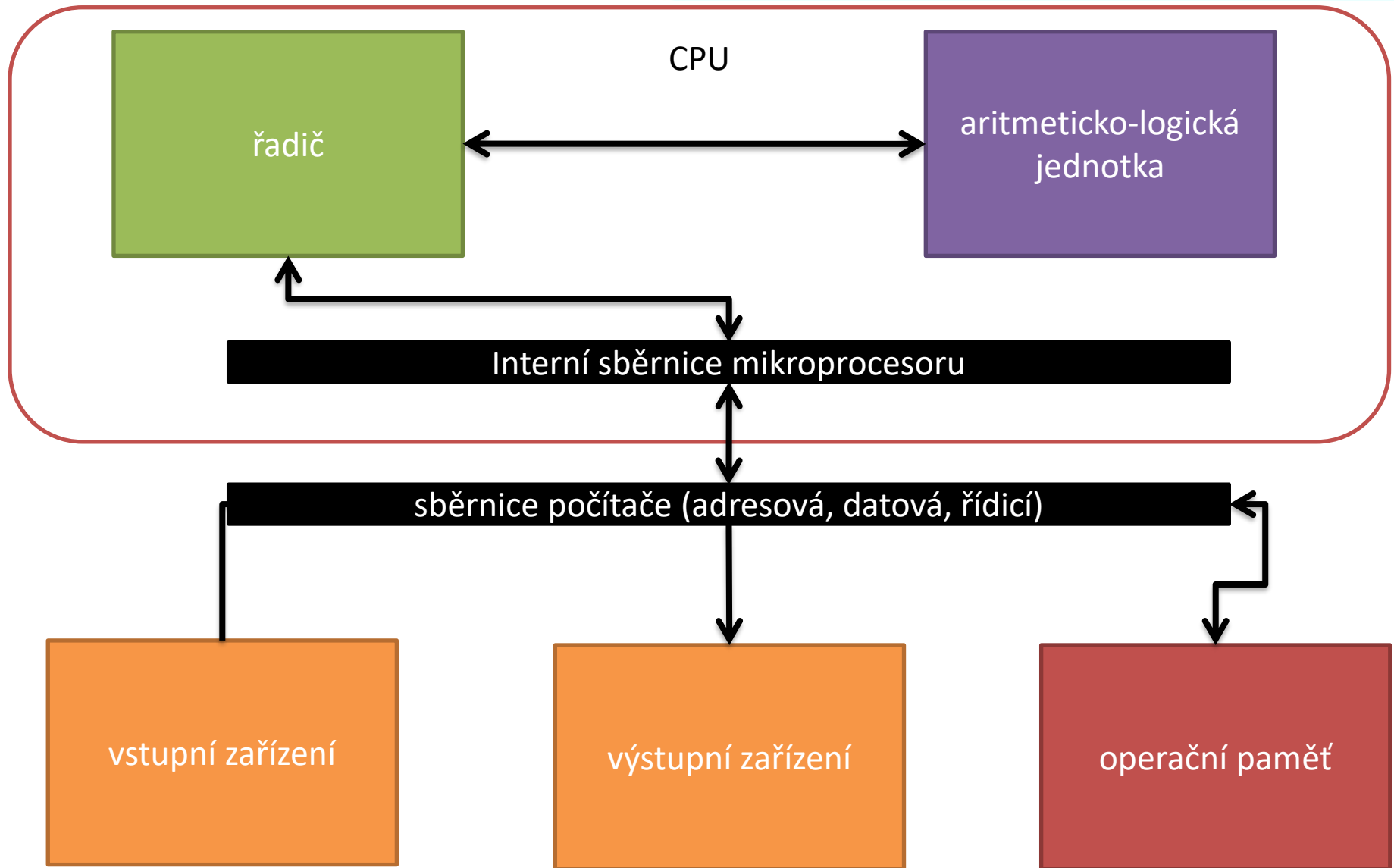


- = Postupná integrace  $\Rightarrow$  modifikace von Neumannovy architektury:
  - = Řadič a ALU spojeny do jednoho čipu = mikroprocesor
  - = Zjednodušení přístupu k operační paměti
  - = Tři typy sběrnic:
    - Adresová
    - Datová
    - Řídící



FIM UHK

## Modifikovaná von Neumannova architektura



- = Architektura mikroprocesoru = uspořádání vnitřních částí ovlivňuje:
  - = Vlastnosti
  - = Způsob programování
  - = Rychlost zpracování přerušení
  - = Princip připojení mikroprocesoru k operační paměti

- =CISC – Complex Instruction Set Computer
- =RISC – Reduced Instruction Set Computer
- =VLIW – Very Long Instruction Word
- =MISC - Minimum Instruction Set Computer

- = Procesory s touto architekturou se vyznačují velmi obsáhlou instrukční sadou
- = Instrukce mají proměnlivou délku kódu i dobu výpočtu
- = To ovšem vede k nutnosti:
  - = Použít složitý řadič  $\Rightarrow$  instrukce trvají i několik desítek taktů
  - = Roste celková složitost mikroprocesoru



- = Poměrně rozsáhlá a složitá sada instrukcí (ISA) obsahovala:
  - = Binární aritmetiku
  - = Instrukce pro práci s textem
  - = Různé numerické formáty
  - = Podporu BCD aritmetiky (dnes využívaná při výpočtech s měnou)

- = Díky složitosti instrukční sady se využívaly mikroprogramy (tj. jednodušší procesory kompatibilní s architekturou CISC, které instrukce emulovaly na základě mikrokódu, tedy mikroinstrukcí, uložených v řadiči)

- = Programátoři a překladače nevyužívali všechny dostupné instrukce
- = CISC procesory jsou hardwarově velmi složité
- = 1980 IBM a University of Berkeley vytvářejí procesor s jednoduchými instrukcemi, menším množstvím adresních režimů nebo víceúčelových registrů, atd.

## = Nové významné vlastnosti:

- = velké množství pracovních registrů, které mohly být ještě více zvětšené pomocí tzv. registrového okna (více sad registrů, mezi kterými se dá přepínat)
- = pracovní registry slouží jako cache paměť nulté úrovně
- = všechny aritmetické a logické instrukce se provádí právě s těmito registry

- = V dnešní době využívá CISC architekturu především firma Intel pro procesory řady x86
- = Ostatní typy a řady dnešních procesorů využívají částečně obě architektury, proto se obvykle uvádí, zda se spíše blíží architektuře CISC nebo RISC
- = RISC architektura je využívána například v procesorech typu ARM nebo integrovaných mikroprocesorech

- = Procesory s architekturou CISC se většinou vyznačují velmi obsáhlou instrukční sadou, mnohdy do značné míry ortogonální (tj. většina instrukcí může být použita se všemi adresními režimy), což vede k nutnosti použití složitého řadiče, instrukce trvají i několik desítek či stovek taktů a celková složitost mikroprocesoru obecně roste.
- = Před několika lety se zdálo, že celá architektura CISC bude opuštěna a nahrazena architekturou RISC. Ve skutečnosti však vznikly jakési hybridní architektury.
- = Dnes nejpopulárnější mikroprocesory jsou sice interně vytvořeny jako RISCové, ale jejich instrukční sada je CISCová. Výhody propojení obou architektur: velká rychlost a interní jednoduchost RISCů a současně kratší kód instrukcí u CISC procesorů (snižují se nároky na vyrovnávací paměti).

- = Mezi velmi úspěšné RISC procesory patří celá rodina mikroprocesorů označovaná MIPS. Mikroprocesory patřící do této rodiny byly použity jak u výkonných grafických stanic firmy SGI, tak i ve zcela opačném trhu: ve vestavěných aplikacích, Nintendo 64, Sony PlayStation, PlayStation 2 atd.
- = Důvod velké oblíbenosti MIPS spočívá především v možnosti propojení jádra mikroprocesoru s dalšími obvody na jednom čipu, což je velmi často využíváno zejména ve vestavěných aplikacích. V praxi to znamená možnost mít celé zařízení umístěné na jednom jediném čipu a programovat přitom na známé, výkonné a mnoha lety praxe odladěné architektuře.

- = Snaha použít co nejjednodušší (a tím i dostatečně rychlé) řadiče mikroprocesoru vedla výrobce k použití speciálního formátu operačního kódu, který byl sestaven tak, že v jedné instrukci jsou uloženy operační kódy pro všechny výkonné jednotky. Výsledkem je architektura nazvaná VLIW



- = Způsob kódování jednotlivých instrukcí v instrukčním slovu je různý.
  - = pro operace ALU se například může třemi bity specifikovat, která operace se má použít (ADD, SUB, AND, OR, XOR, NOT, ROR, ROL), dalšími čtyřmi bity první registr vstupující do operace a následujícími čtyřmi bity registr druhý.
- = Konstanta v instrukčním slově je použita pro naplnění vybraného pracovního registru
- = Horizontální formát instrukcí umožňuje řadiči zůstat velmi jednoduchý
- = Paralelní práce všech jednotek
- = Instrukce mají pevnou délku, obvykle podstatně větší než u RISC (instrukční slovo rozděleno do polí)

## = Přednosti VLIW

- = velká jednoduchost řadiče
- = Možnost zpracování více instrukcí paralelně – „osloveny“ různé prvky mikroprocesoru zároveň

## = Nevýhody

- = veškeré optimalizace se musí provádět v závislosti na konkrétní konfiguraci daného mikroprocesoru
- = není vhodné v případech používání interpretovaných kódů
  - = protože interprety nemají (na rozdíl od překladačů) tolik času na provádění zde zcela nutných optimalizací
- = Nutné řešit konflikty jednotlivých operací nad registry

- = MISC je alternativou k architekturám CISC i RISC, tzv. zásobníková architektura
- = Typická především použitím instrukcí bez operandů
  - = operandy jsou totiž známy implicitně, jelikož se většina operací provádí s hodnotami uloženými na interním či externím zásobníku (stack)
- = V porovnání s RISC jsou výhody:
  - = v menších nárocích na rychlost operačních pamětí
    - = (kratší instrukce, mnohdy odpadá nutnost použití cache pamětí)
  - = rychlejší reakce na přerušení
    - = (nemusí se nikam ukládat stav procesoru atd.)
  - = procesory založené na této architektuře se používají například v oblasti řízení či real-time systémech

- = Důležitou součástí mikroprocesoru jsou takzvané registry.
- = jde o paměti schopné uschovat vždy jedno slovo.
  - = Typická šířka slov, tj. počet současně zpracovávaných bitů, se pohybuje od 4 bitů do 128 bitů, dnes se nejčastěji jedná o 8, 16, 32 či 64 bitů
- = kapacita registrů je velmi malá
- = jedná se o paměť využívanou prakticky všemi instrukcemi
  - = Vytvořeny nejrychlejší dostupnou technologií – jde o statickou paměť (často založená na klopných obvodech typu D či JK)
  - = Vzhledem ke konstrukci ztrácejí uložené hodnoty po odpojení napájení mikroprocesoru – pro trvalé uložení hodnot je nutné využít jiný druh paměti

- = Aritmeticko-logická jednotka (zkráceně ALU), která provádí, výpočet základních aritmetických a logických funkcí je poměrně složitý kombinační logický obvod
- = Princip ALU je však poměrně jednoduchý:
  - = na její vstup jsou v typické konfiguraci přivedena dvě n-bitová čísla, pomocí kterých se zvolí příslušná operace
  - = na výstup ALU je po určitém zpoždění posláno m-bitové číslo, které odpovídá výsledku zvolené operace
  - = na výstupu ALU mohou být dalších signály (flags) nastavované v závislosti na výsledku či průběhu výpočtu
    - = Jedním z příznaků může být příznak nulovosti nebo příznak přetečení.

## = Někdejší jednodušší ALU:

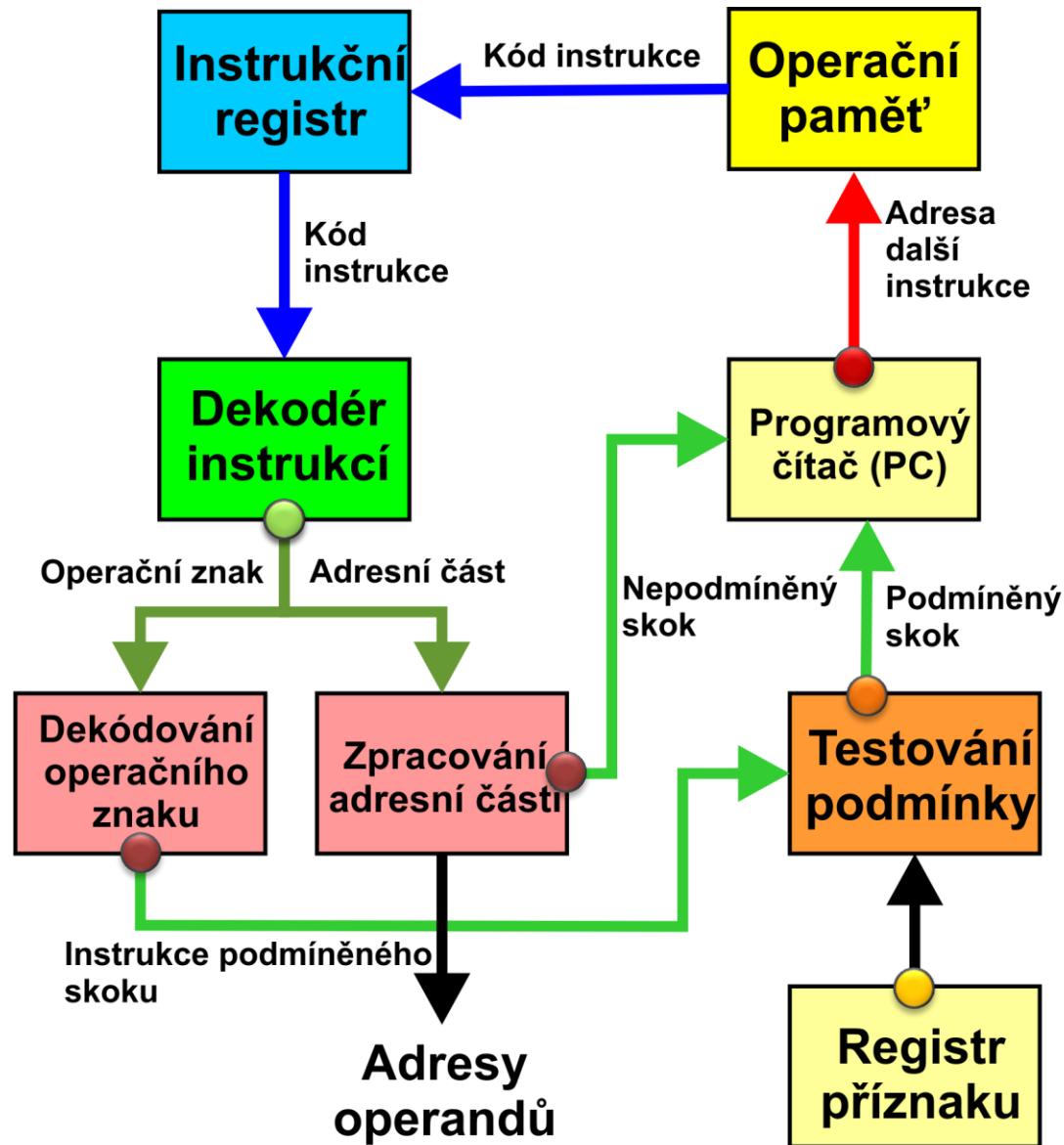
- = Součet dvou čísel
- = Rozdíl dvou čísel
- = Výpočet dvojkového doplňku
- = Porovnání dvou operandů
- = Negace všech bitů prvního operandu
- = Bitový logický součet
- = Bitový logický součin

## = Současné ALU jsou výrazně složitější, umí také:

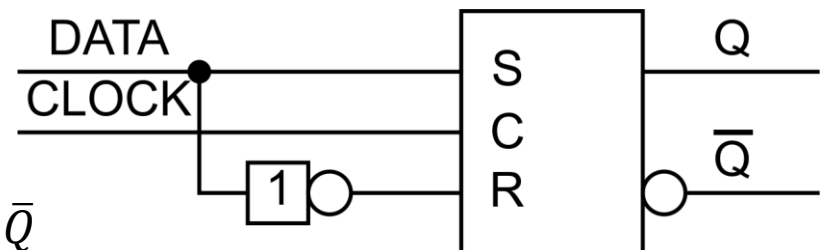
- = Součin dvou čísel
- = Součet dvou čísel
  - = Za ALU se obvykle nepovažuje jednotka provádějící operace v pohyblivé řádové čárce



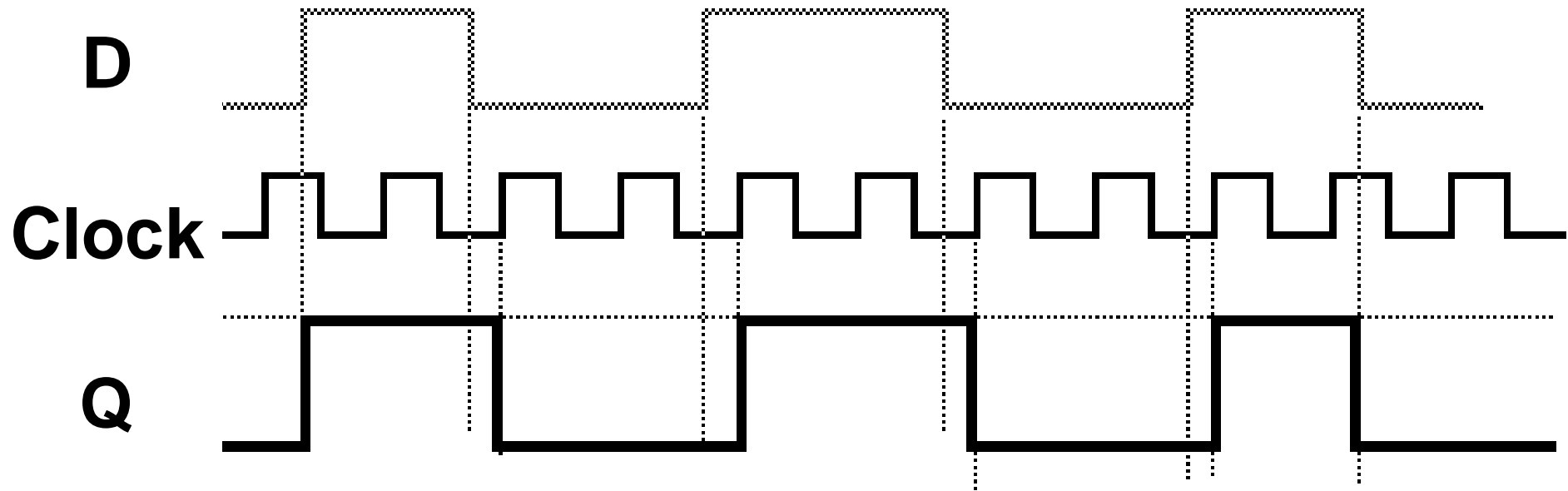
## Příklad zapojení ALU v MCU



- = Kromě ALU se v mikroprocesoru musí nacházet i řadič a sada registrů.
- = Tyto bloky mikroprocesoru se z velké části skládají ze sekvenčních logických obvodů
  - = obvodů, jejichž výstup závisí nejenom na stavu vstupů, ale také na předchozích stavech obvodu
  - = obsahuje nějakou formu paměti
- = Mezi nejjednodušší patří klopné obvody (paměťové členy)
  - = klopný obvod je možné považovat za jednobitovou paměťovou buňku
  - = Základním klopným obvodem je RS
    - = má dva vstupy označené symboly R (reset) a S (set)
    - = a dva výstupy označené symbolem  $Q$  a  $\bar{Q}$







Časový diagram klopného obvodu D

- = Paralelním spojením několika klopných obvodů typu D (nebo i RS) vznikne registr
- = Umožňuje uschovat n-bitovou hodnotu
  - = V mikroprocesorech je obecně umístěno větší množství registrů, typicky bývají minimálně dva registry na vstupech ALU a jeden registr pracuje ve funkci ukazatele na zpracovávanou instrukci



Univerzita Hradec Králové  
Fakulta informatiky a managementu

Děkuji za pozornost...

