Úvod do architektur personálních počítačů

Cíl přednášky

- Popsat principy proudového zpracování informace.
- Popsat principy zřetězeného zpracování instrukcí.
- Zabývat se způsoby uplatnění tohoto principu v procesorech fy INTEL.
- Stručně se zabývat vývojem procesorů INTEL (parametry, pojmy) a PC.

První PC – rok 1981

- **Parametry prvního PC:**
  - kmitočet 4,77 MHz,
  - mikroprocesor I8088
  - numerický koprocesor I8087
  - RAM 16 - 64 kB
  - ROM 40 kB (BIOS, BASIC)
  - konektory: klávesnice
  - kazetový magnetofon
  - 5 konektorů pro připojení rozšiřujících desek

- **Další vývoj:**
  - 12. srpna 1981 byl počítač veřejně prezentován, do konce r. 1983 - 400 000 prodaných počítačů
  - 1985, 1986, firma IBM získala 55% trhu osobních počítačů
  - 80. léta – výrazné postavení fy IBM.

Posuzované parametry PC

- Typ procesoru.
- Synchronizační kmitočet procesoru.
- Synchronizační kmitočet rozhraní procesoru – rychlost komunikace s okolním světem.
- Šířka sběrnic procesoru – adresová, datová.
- Instalace procesoru – patice (ZIF – Zero Insertion Force).
- Typ systémové sběrnice a její synchronizace (ISA, VLB, PCI, PCI-X, PCI Express).
- Existence dedikované sběrnice v sestavě PC (AGP).
- Technologie pamětí (DRAM, synchronní paměť, DDR, ……).
- Fyzická realizace paměťových modulů (šířka toku dat).
- Existence rychlé vyrovnávací paměti (L1, L2, L3).
- Instalace rozhraní do systémové desky.
- Typy V/V sběrnic PC.

Jak je to s L1, L2 a L3?

- První byla L2 – nebyla součástí procesoru, byla umístěna v samostatných paticích na systémové desce. Problém: rychlost komunikace !
- L1 – ve stejném pouzdře jako procesor – vyšší rychlost komunikace.
- Dokonalejší technologie – L2 se přesunula do pouzdra procesoru, RVP na systémové desce – označena jako L3.
- Pozn.: existují i typy procesorů, kdy L3 je uvnitř pouzdra procesoru a vně pouzdra je L4 (Intel Skylake čip).

Charakter dalšího vývoje – 90. léta

- Existovala řada firem, které kompletovaly počítače tzv. IBM kompatibilní.
- Značkové počítače – vyšší cena, problémy s kompatibilitou (IBM, Compaq), mechanicky nekompatibilní.
- Neznačkové počítače (no name) – výrobci hlavně v Asii – nižší cena.
- Na trhu existovala řada modelů více či méně úspěšných, různě technicky vybavených, různě spolehlivých a v různých cenových relacích.

PC na bázi I80386

- **Charakteristika:**
  - kmitočet 16/20/25/33 MHz
  - mikroprocesor I80386
  - numerický koprocesor - I80387
  - datová sběrnice - 32 bitů
  - adresová sběrnice - 32 bitů
  - paměť realizována jako tzv. krátké SIMM moduly (Single In-line Memory Module),
  - rychlá vyrovnávací paměť externí (L2, level 2) - cache
  - RAM běžně 1 - 16 MB, možnost rozšíření až na 4 GB (šířka adresové sběrnice – 32 bitů)
  - HDD 80 MB - 1 GB
  - rychlost: PC 386/33 MHz - 4 MIPS
    využití v 90. letech: grafika, simulace
    dříve i servery v počítačových sítích.

Poznámka: odlišení 32 bitového rozhraní mikroprocesoru I80386 od I80386SX - I80386 se označoval jako I80386DX (kompatibilita zdola).

PC na bázi I80486

- Charakteristika
  - kmitočet 33/66 MHz
  - mikroprocesor I80486
  - numerický koprocesor - I80487 – integrován do stejného pouzdra jako procesor
  - datová sběrnice - 32 bitů
  - adresová sběrnice - 32 bitů
  - paměť realizována jako tzv. dlouhé SIMM moduly, později moduly DIMM (Dual In-line Memory Module)
  - rychlá vyrovnávací paměť 8 kB (cache) - integrována do procesoru, L1(kromě L2)
  - sběrnice ISA, VLB, PCI
  - rychlá vyrovnávací paměť L1, L2

- Využití: ve své době výkonné servery v rozsáhlejších místních sítích, výkonná pracoviště pro vývojové systémy CAD/CAM.

Stav na konci 90. let a po roce 2000

- Výrazný technologický pokrok nejen v technologii zasahující technologii výroby systémových desek ale i dalších komponent (stupeň integrace).
- Snaha o zavádění inovací, které mají charakter nových architektur, jejichž cílem je snaha o efektivní provádění instrukcí – **proudové zpracování informace**.
- Výrazné zlepšování parametrů včetně zavádění nových typů periferních zařízení.
- Rozvoj systémové sběrnice.
- Zlepšování parametrů především procesorů – u dalších komponent tento trend nebyl tak jasný (viz. např. řadič přerušení, řadič DMA).
- Snaha o uplatňování principů **proudového zpracování informace** – její zdokonalování – **hnací motor architektur procesorů INTEL.**

Další vývoj technologie PC

- Intel® Pentium® 4 Processor in 478-pin Package and Intel® 845 Chipset Platform for DDR Platform Design Guide
- Pojmy:

  Kmitočet, jímž je synchronizován procesor: mezi 3 – 4 GHz

  Intel NetBurst® Microarchitecture – souhrnný pojem

  Hyper-Threading Technology – mohou běžet dvě úlohy

  Hyper-pipelined Technology – úroveň proudového zpracování instrukcí

  1066 MHz, 800 MHz, 533 MHz or 400 MHz – příklady synchronizace přenosů přes rozhraní procesoru => rychlost dostaneme, pokud tento údaj vynásobíme 8 (hodnota v GB/s)

  Integrated 2MB L3 Cache – rychlá vyrovnávací paměť L3

  845 – čipsetová sada, jsou i další sady, postaveno na pojmech severní, resp. jižní most.

Dnešní stav technologie

- Procesory jsou dnes velice výkonné a pracují na vysokých taktech.
- **Dnešní směr se spíše ubírá k většímu počtu jader než k dalšímu zvyšování taktu.**
- Důvod je jednoduchý - pokud se u procesoru navyšuje frekvence, úměrně s tím roste **spotřeba elektrické energie a v závislosti na tom stoupá i teplota**.
- Tím vzniká obrovský problém takto **vysoce taktovaný procesor uchladit.**
- Více jádrové procesory mohou běžet na podstatně nižších frekvencích, ale tím, že si úlohu umí rozdělit do dvou, čtyř či více jader, jsou schopné pracovat podstatně rychleji s nižší elektrickou spotřebou a tím vzniká méně tepla.
- K nejvýznamnějším výrobcům procesorů pro počítače na našem trhu patří Intel a AMD.

Značení procesorů

- I7 – 7820 HK

  i7 – výkonnostní řada

  7 – označení generace a mikroarchitektury

  8 – výkon v rámci řady

  20 – informace o grafickém jádru

  HK – další vlastnosti procesoru

Proudové zpracování informace

- Zpracování informace - velký počet jednoduchých úkonů (elementárních operací).
- Elementární operace – navazují na sebe a vzájemně se doplňují v souhlase s algoritmem požadovaného zpracování.
- Operand – postupně prochází řadou transformací.
- Transformace operandu – buď v jednom funkčním bloku nebo v různých funkčních blocích.
- Následující operace musí vždy čekat na ukončení všech předcházejících operací.
- Doba zpracování T_z – součet časových úseků potřebných pro jednotlivé operace.
- Zpracování informace – instrukce počítače.

Poznámka k proudovému zpracování informace

- Rozdělení realizace instrukce (úlohy) do sekcí – vždy (už v prvních typech počítačů).
- Příklad: čtení instrukce z paměti, dekódování instrukce, čtení operandů z paměti, výpočet, uložení výsledku.
- Srovnání dřívějších architektur s dnešními architekturami: **zpracování instrukce je rozděleno na sekce** – tak, jako dříve,

  sekcí je nyní výrazně více,

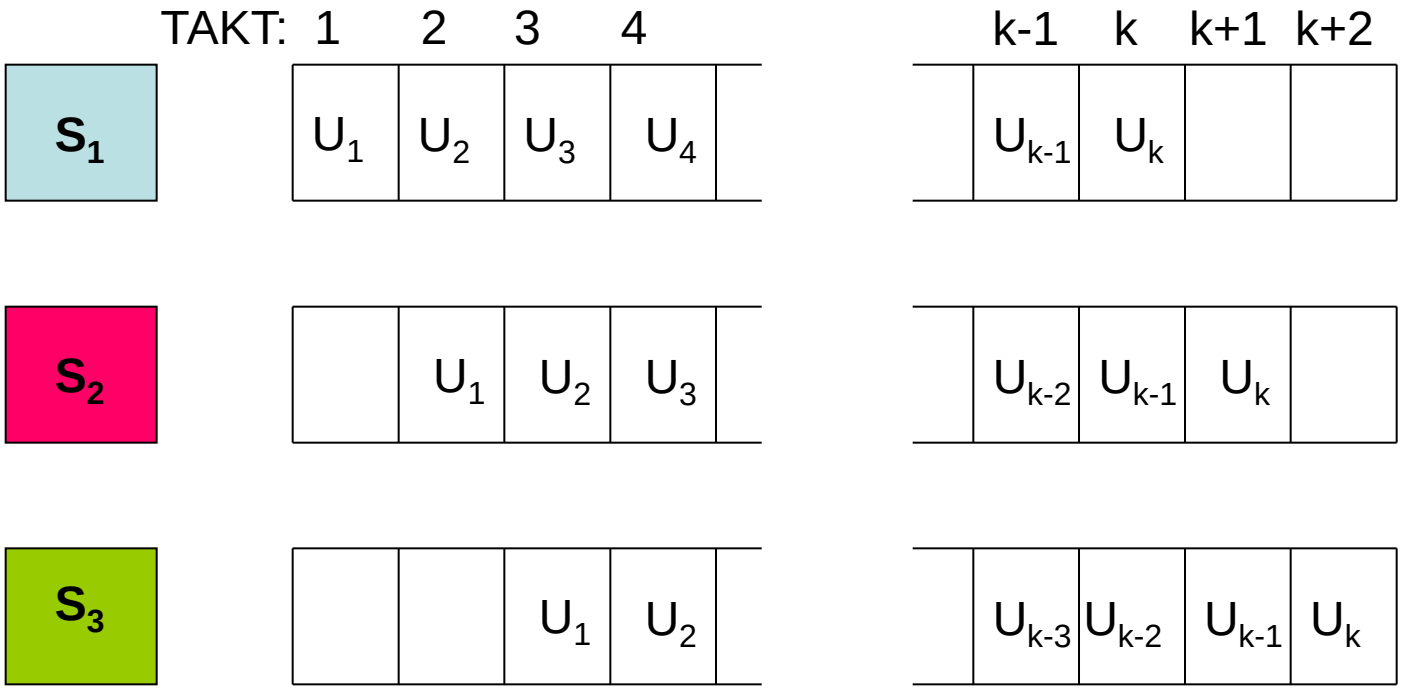
  instrukce jsou zpracovány proudově – instrukce *a* se po přečtení z paměti ve 2. kroku posune do sekce, kde se bude realizovat dekódování, z paměti se bude číst instrukce *b*.
- Sekce - komponenty počítače, úloha – instrukce.

Proudové zpracování informace - pokračování

- Stav bez proudového zpracování: **N** operací se má provést => **N funkčních** bloků je potřeba => po dobu **N – 1** časových úseků je každý funkční blok nevyužit a čeká na příchod dalšího operandu.
- Proudové zpracování: postupné předávání výsledků zpracování mezi funkčními bloky => **možnost využívat všechny bloky současně tak, že v každém bloku se zpracovává jiný operand (je realizována jiná činnost)** – základní myšlenka proudového zpracování informace.
- Proudové zpracování – pohyb operandů (tok instrukcí) připomíná kapalinu proudící potrubím.
- Anglický termín – **pipelining** (pipeline – dálkové potrubí).
- Česká terminologie – **zřetězené zpracování**.

Rezervační tabulka proudově pracující jednotky

Rezervační tabulka – prostředek na zobrazení toku informace v systému se zřetězeným zpracováním informace



S –sekce, U – úloha (instrukce)

Rezervační tabulka - komentář

- Proudově pracující jednotka v našem příkladě sestává ze třech funkčních bloků – S_1, S_2, S_3 – budeme je označovat jako sekce.
- Do sekce S_1 vstupují instrukce/operandy, ze sekce S_3 vystupují výsledky.
- Úloha u_i – činnost (např. instrukce), která je rozdělena na dílčí takty (kroky).
- Rezervační tabulka – obsazení každé sekce úlohou u_i během několika po sobě následujících taktů (kroků).
- Konkrétní takt – každá úloha se zpracovává v každé sekci, v každé sekci je zpracováván dílčí krok realizace instrukce.
- Ideální stav – **doba setrvání v jedné sekci musí být ve všech případech stejná** – celá jednotka pak pracuje synchronně => snaha vytvářet takové sekce, v nichž zpracování trvá stejně dlouho – problém (např. úlohy pracující s pamětí – několik typů pamětí různé rychlosti).

Rezervační tabulka - komentář

- Proudově pracující jednotka se postupně **zaplňuje** – jednotka sestávající z **n sekcí** se zaplňuje po dobu **n-1 taktů**.
- Jednotka pracuje naplno - počínaje n-tým taktem.
- Tento stav trvá až do taktu **k**, kdy přijde na vstup poslední úloha.
- **Vyprazdňování sekcí –** trvá **n-1** taktů, pro 3 sekce je to 2 takty.
- Jistým způsobem se zvýší propustnost celé sestavy v porovnání se situací, kdy se úlohy (instrukce) nezpracovávají proudově.

Součinitel zvýšení propustnosti proudově pracující jednotky - Q

- Součinitel Q – úvahy:
  - Jednotka s *n* sekcemi, **jednotka nepracuje proudově**: jedna úloha (instrukce) trvá *n* taktů.
  - Úloh je *k* => zpracování *k* úloh trvá *nk* taktů.
  - **Jednotka pracuje proudově** – *k* úloh bude trvat:

    $k + n - 1$ taktů.

    Vyprázdnění jednotky na závěr

    k úloh bude trvat *k* taktů

Výpočet součinitele Q

$$Q = \frac{n\,k}{k + n - 1}$$

Čitatel zlomku – jednotka je rozdělena na *n* sekcí, instrukce nejsou zpracovány proudově, instrukce se zpracovává *n* taktů – po dobu *n* taktů se zpracovává jedna instrukce, po jejím dokončení se začne zpracovávat instrukce další – v konkrétním okamžiku se zpracovává pouze jedna instrukce, výsledek instrukce je k dispozici po *n* taktech.

Jmenovatel zlomku – instrukce se zpracovává v *n* taktech, každému taktu odpovídá jedna sekce – po zaplnění se současně zpracovává *n* instrukcí, v každém taktu je na výstupu výsledek jedné instrukce.

Pokud jednotka pracuje nepřetržitě po dlouhou dobu (např. program s velkým počtem instrukcí), pak $k >> n$, dostaneme $Q = n$ =>

proudové zpracování v *n* sekcích může zvýšit propustnost systému nejvýše *n*-krát.

Proudové zpracování programu - shrnutí

- Základní myšlenka: provádění každé instrukce lze rozložit do několika téměř standardních úkonů, které lze realizovat nezávisle na sobě na různých místech v počítači.
- Příklad: instrukce typu aritmetické nebo logické operace se dvěma operandy a jedním výsledkem – jeden operand je uložen v hlavní paměti a druhý je v registru procesoru (paměti), výsledek se ponechá v registru procesoru.
- Posloupnost kroků:
  1. výběr instrukce z paměti,
  2. dekódování instrukce,
  3. výpočet reálné adresy,
  4. výběr 1. operandu z paměti,
  5. výběr 2. operandu z paměti,
  6. provedení operace,
  7. uložení výsledku do registru.

Proudové zpracování programu

- Provedení instrukce – jsou potřebné dva přístupy do hlavní paměti:
  výběr instrukce - **VI**,
  výběr operandu **VO**.
- Po výběru instrukce – dekódování instrukce – **DI.**
- Po výběru operandu – provedení operace **PO**.
- Každá operace s procesorem obsadí celý procesor.
- Každá operace s HP obsadí paměť.

Rezervační tabulka pro jednu instrukci

paměť	VI		VO	
procesor		DI		PO

Využití předvýběru instrukce

- V prvním cyklu je obsazena pouze paměť, procesor je volný => možnost překrýt konec jedné instrukce se začátkem následující => v okamžiku PO1 je možné realizovat VI2, ….
- Stupeň využití paměti se tak zvýší na 66,6 %, zkrácení průměrné doby provedení jedné instrukce na 3 cykly.

Rezervační tabulka s předvýběrem instrukce

paměť	VI1		VO1	VI2		VO2	VI3
procesor		DI1		PO1	DI2		PO2

Plné využití paměti i procesoru

Vyšší stupeň využití – výběr každé nové instrukce je zahájen hned, jakmile je paměť volná.

Rezervační tabulka pro plné využití paměti a procesoru

paměť	VI1	VI2	VO1	VO2	VI3	VI4	VO3
procesor		DI1	DI2	PO1	PO2	DI3	DI4

Počítač je rozdělen do dvou sekcí – paměť, procesor - v každém okamžiku probíhají nejvýše dva úkony.

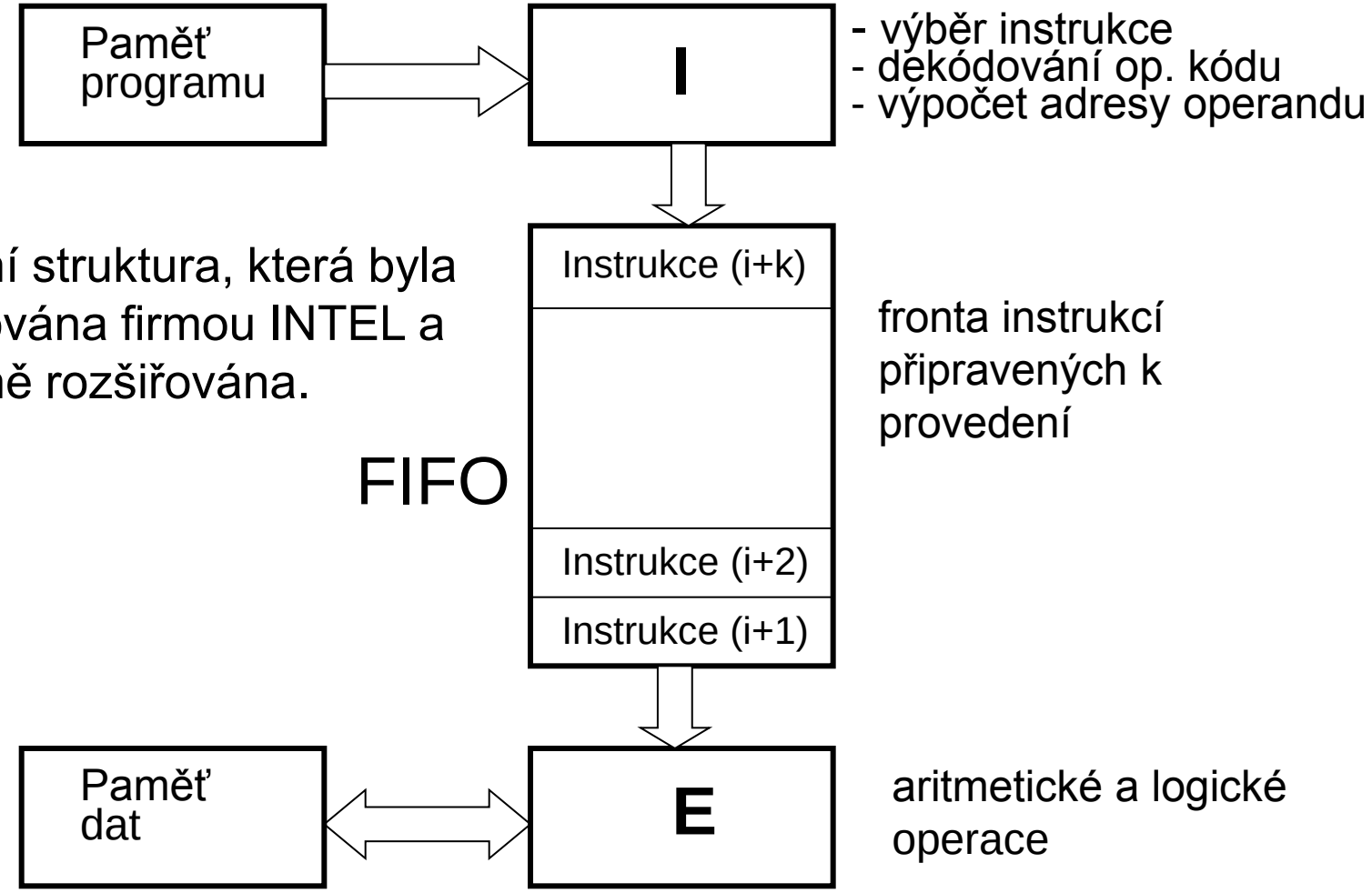
Tato struktura dává zcela minimální možnosti využití proudového zpracování programu – **snaha o rozdělení provádění operace na větší počet kroků – trend v procesorech INTEL**.

Rozdělit realizaci instrukce v procesoru na více kroků → rozdělit procesor na segmenty.

Harvardská architektura

- Snaha o dosažení postupného provádění několika instrukcí současně => nutnost rozčlenit počítač na odpovídající počet sekcí.
- Dopad těchto snah – nutnost oddělit paměť programu od paměti operandů (harvardská architektura).
- Procesor rozdělen na minimálně dvě jednotky – **I** a **E:**
  - Jednotka I: předzpracování instrukce (dekódování kódu, výpočet adresy operandu),
  - Jednotka E: provedení operace podle kódu instrukce (E - execution).
- Výsledek: počítač obsahuje 4 jednotky: I, E, paměť programu, paměť dat.

Blokové schéma počítače využívajícího principy proudového zpracování



Základní struktura, která byla akceptována firmou INTEL a postupně rozšiřována.

FIFO

Blokové schéma – komentář

- Mezi jednotky I a E je vložena paměť fronty instrukcí – ukládají se do ní instrukce vybrané z paměti programu.
- Důvod – vyrovnání rychlosti mezi jednotkami I a E (paměť je výrazně pomalejší než jednotka E).
  Paměťové prvky (registry), v nichž je uložena instrukční fronta, jsou výrazně rychlejší než prvky, z nichž je vybudována paměť.

- Instrukce skoku - problém:
  Instrukce podmíněného skoku v posloupnosti instrukcí načtených do fronty instrukcí – vyhodnotí se až v jednotce E.
  Výsledek: bude se skákat/nebude se skákat.
  Alternativa „bude se skákat“ – instrukce načtené do fronty instrukcí se ruší, musí se načíst instrukce z nové adresy podle výsledku instrukce podmíněného skoku.

- Řešení:
  1) předvídání výsledku skoku,
  2) existence dvou front, do obou se načítají instrukce, jakmile se v jednotce I rozpozná instrukce skoku,
  3) dostatečně dlouhá fronta instrukcí, zvýšení pravděpodobnosti, že obě instrukce budou ve frontě.

Blokové schéma – rezervační tabulka

Paměť programu	VI1	VI2	VI3	VI4	VI5			
Jednotka I		DI1	DI2	DI3	DI4	DI5		
Paměť dat			VO1	VO2	VO3	VO4	VO5	
Jednotka E				PO1	PO2	PO3	PO4	PO5

Plné využití procesoru – současně se zpracovávají čtyři instrukce.

Ideální případ:

1) všechny instrukce jsou stejného typu,
2) doba zpracování v jednotlivých sekcích je stejná.

Pozn.: operace VI a VO probíhají současně, paměť dat je oddělena od paměti programu.

Problémy s typy instrukcí

- **„Problematické“ instrukce**:
  - instrukce skoku, podmíněného i nepodmíněného,
  - instrukce, které vyžadují více taktů (např. pro práci s pamětí) než instrukce jiné,
  - problém, když se v jednotce I rozpozná instrukce podmíněného skoku, její podmínka ale bude vyhodnocena v jednotce E, až se instrukce dostane do fáze provádění.
  - Řešení:
    - zastavit činnost a čekat, jak dopadne výpočet podmínky,
    - pokračovat v plnění fronty, jako kdyby instrukce podmíněného skoku neexistovala.
  - Problém: pokud se skok uskuteční, pak celá fronta „je k ničemu“. Musí se zaplnit instrukcemi z bodu, na nějž ukazuje instrukce skoku, po tu dobu je jednotka E v nečinnosti.
- Jiné možnosti řešení:
  - dvě fronty,
  - bit predikce (architektury RISC a moderní architektury CISC).
- Závěr: dělení na jednotky I a E je velmi hrubé – v procesorech INTEL zavedeno jemnější dělení.

Uplatnění principů zřetězeného zpracování v procesorech fy INTEL - I8088

- Mikroprocesor I8088 – zřetězení dvou činnosti – realizace instrukcí a komunikace s vnějším světem
  - operační jednotka - EU (Execution Unit) –realizace instrukcí,
  - řídicí jednotka sběrnice - BIU (Bus Interface Unit), která řídila komunikaci s vnějším světem (např. čtení instrukcí z paměti).
- Mikroprocesor I8088 existoval v době, kdy principy proudového zpracování nebyly příliš propracovány, tomu odpovídá úroveň využití těchto principů.
- V době existence a nasazení takto koncipovaná architektura vyhovovala.

I8088 – proudové zpracování

- Zvýšení efektivity provádění instrukcí: BIU načítala ve volných chvílích mezi přístupy do paměti instrukce do **instrukční fronty** (4 slabiky).

  => pokud se neporuší sekvenční posloupnost instrukcí (skoková instrukce nebo volání podprogramu), vybírá se další instrukce z této instrukční fronty

  => instrukce je k dispozici podstatně rychleji, než kdyby se celá četla z paměti.

- Další vývoj : vyšší granularita (vyšší počet jednotek)

Uplatnění principů zřetězeného zpracování v procesorech fy INTEL - I80286

- Snaha o jemnější dělení činností souvisejících s realizací instrukcí => 4 jednotky.

  EU (Execution Unit) - operační jednotka

  BU (Bus Unit) - sběrnicová jednotka

  IU (Instruction Unit) - jednotka předzpracování instrukcí

  AU (Address Unit) - adresace paměti

Uplatnění principů zřetězeného zpracování v procesorech fy INTEL - I80386

Vyšší granularita:

- BU (Bus Unit) - jednotka řízení sběrnice
- PU (Prefetch Unit) - jednotka předzpracování instrukcí - čte dopředu až 4 slabiky instrukcí
- DU (Decode Unit) - dekodovací jednotka
- EU (Execution Unit) - operační jednotka
- SPU (Segmentation and Paging Unit) - segmentace a stránkování paměti (jednotka správy paměti provádějící transformaci logické adresy na adresu fyzickou).

- Trend: zvyšování granularity, snaha o navyšování počtu kroků, na něž je provedení instrukce rozděleno.

Uplatnění principů zřetězeného zpracování v procesorech fy INTEL - Pentium

- Architektura Pentia byla klasifikována jako superskalární architektura – tzn. obsahovala minimálně dvě fronty zpracování instrukcí.
- Obecně – superskalární architektura sestává ze dvou a více front instrukcí – nutno vypracovat párovací pravidla.
- Další pojmy – skalární, hyperskalární.
- Další podrobnosti – samostatná přednáška.

Optimalizace struktury proudově pracující jednotky

Důležité poznatky:

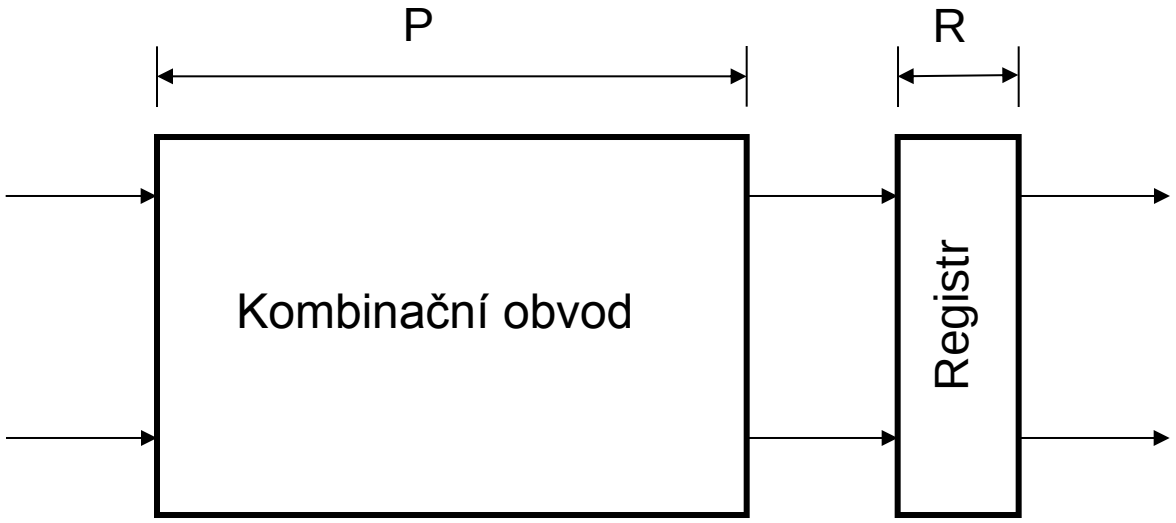
- Čím větší počet sekcí, tím větší počet úloh může jednotka zpracovávat současně.
- Frekvence, s níž se výsledky objevují na výstupu (výkonnost), roste s počtem sekcí (čím větší počet sekcí, tím vyšší kmitočet synchronizace – můžeme si to dovolit, protože zpracování v jednotlivých sekcích trvá kratší dobu).
- Na každou sekci musí navazovat registr, výsledek úlohy musí být zaznamenán (synchronizován) => větší počet sekcí – větší počet registrů – **nutno brát v úvahu zpoždění způsobené průchodem informace přes registry**.
- **Zařazení registrů – zvýšení ceny, zvýšení objemu elektroniky, zvýšení spotřeby energie** => otázka, jaký počet sekcí je optimální?

Problém je možné definovat takto:

- implementace proudového zpracování představuje nárůst elektroniky, která zvyšuje náklady – je potřeba najít kompromis.

Hledání optimálního počtu sekcí

Původní obvod – bez proudového zpracování

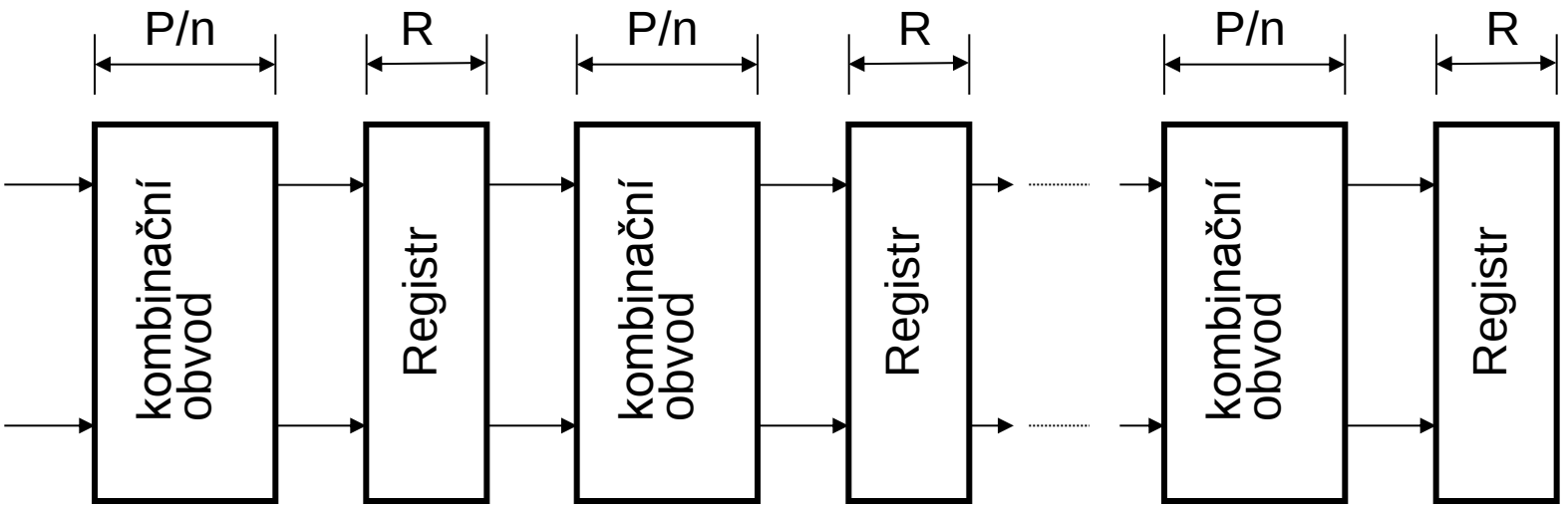


P – doba průchodu signálu kombinačním obvodem

R - doba průchodu signálu registrem

Úkol: rozdělit kombinační obvod do **n** sekcí a za každou sekci vložit registr – dostaneme proudově pracující strukturu

Hledání optimálního počtu sekcí



Součet všech dob průchodu (zpracování) je roven době P.

Cena registrů je stejná jako cena původního registru.

Úkol – snažíme se najít optimální počet sekcí **n**, tzn. takovou strukturu, pro kterou je nejvýhodnější podíl výkonu k nákladům, vyjádřený součinitelem kvality **Q = V/N**.

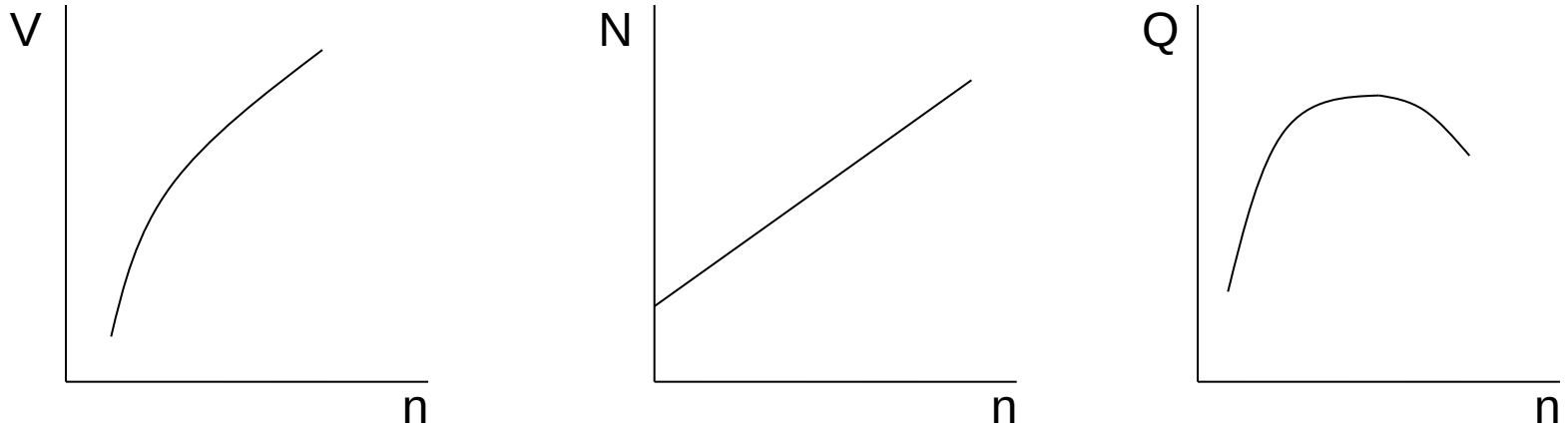
Místo kombinačního obvodu si můžeme představit sekci procesoru (počítače) provádějící dílčí kroky související s realizací instrukce.

Hledání optimálního počtu sekcí

- Výpočet výkonu:
  - Zpoždění v jedné sekci T = P/n + R.
  - T je časový interval, po jehož uplynutí můžeme na vstup proudově pracující jednotky vložit další operand.
  - Výkon V – maximální frekvence, s níž můžeme na vstup přivádět operandy, tzn. převrácená hodnota T.
  - **V = 1 / (P/n + R).**

  - Výpočet nákladů – cena kombinační sítě je konstantní, k ní se připočte **n** násobek ceny registru.
  - **N = nL + C**
  - **Q = V / N = 1 / ((P/n + R) * (nL + C))**

Hledání optimálního počtu sekcí



Funkce **Q(n)** má maximum, potřebujeme získat hodnotu **n** v maximu funkce – 1. derivaci položíme rovnu 0.

Výsledek: $(PC/LR)^{1/2}$

P – doba průchodu signálu kombinačním obvodem
R - doba průchodu signálu registrem
C – cena kombinační sítě
L – cena registru

Hledání optimálního počtu sekcí při proudovém zpracování instrukcí

- Situace: počítač (procesor) máme rozdělit na sekce, které se budou podílet na proudovém zpracování toku instrukcí.
- Řešení: obtížnější než v předcházejícím případě.
- Důvod: na zpracování instrukce se podílejí komponenty s odlišnou dobou „práce na instrukci“.
- Příklad: několik typů pamětí – operační paměť (DRAM), rychlá vyrovnávací paměť (L1, příp. L2, příp. L3) => **různá rychlost získávání operandů**.
- Důsledek: podle toho, kde je operand přítomen v okamžiku jeho potřeby, různá doba trvání zpracování v patřičné sekci.
- Řešení v architekturách RISC: operandy jsou uchovávány v registrech, dokonalejší forma – sady registrů.
- Přepínání mezi úlohami – přepínání mezi sadami registrů.
- Architektury CISC: obtížně se realizuje požadavek, aby doba zpracování každého kroku instrukce v jednotlivých sekcích trvala stejnou dobu.

Typy architektur proudového zpracování - terminologie

- Architektura s jednou frontou – skalární architektura.
- Front instrukcí je více => superskalární architektura.
- Superskalární architektura – problémy s koordinací činností v jednotlivých frontách – musejí spolupracovat nejenom komponenty řazené za sebou ale také jednotlivé paralelní fronty – problém tzv. párování instrukcí.
- Další problém – párování instrukcí do front tak, že tyto instrukce mohou být prováděny paralelně – párovací pravidla.
- Superskalární struktura - multiprocesorový systém na jednom čipu.
- Příklady superskalárních struktur – Pentium mělo dvě fronty instrukcí (u, v).
- Hyperskalární architektura – navyšování počtu sekcí v jedné frontě (hyper-pipeline architecture), hranice 10 sekcí, existují architektury i s více sekcemi.

Shrnutí

- Principy zřetězeného zpracování instrukcí – mít představu, jak se tyto principy vyvíjely.
- Skalární / superskalární / hyperskalární architektura – rozdíly.
- Harvardská architektura – princip, souvislost se zřetězeným zpracováním instrukcí.
- Problém hledání optimálního počtu sekcí zřetězené architektury.
- Problémy s instrukcemi skoku při proudovém zpracování instrukcí.

Shrnutí

- Vývoj architektur rychlých vyrovnávacích pamětí.
- Součinitel zvýšení propustnosti proudově pracující jednotky.