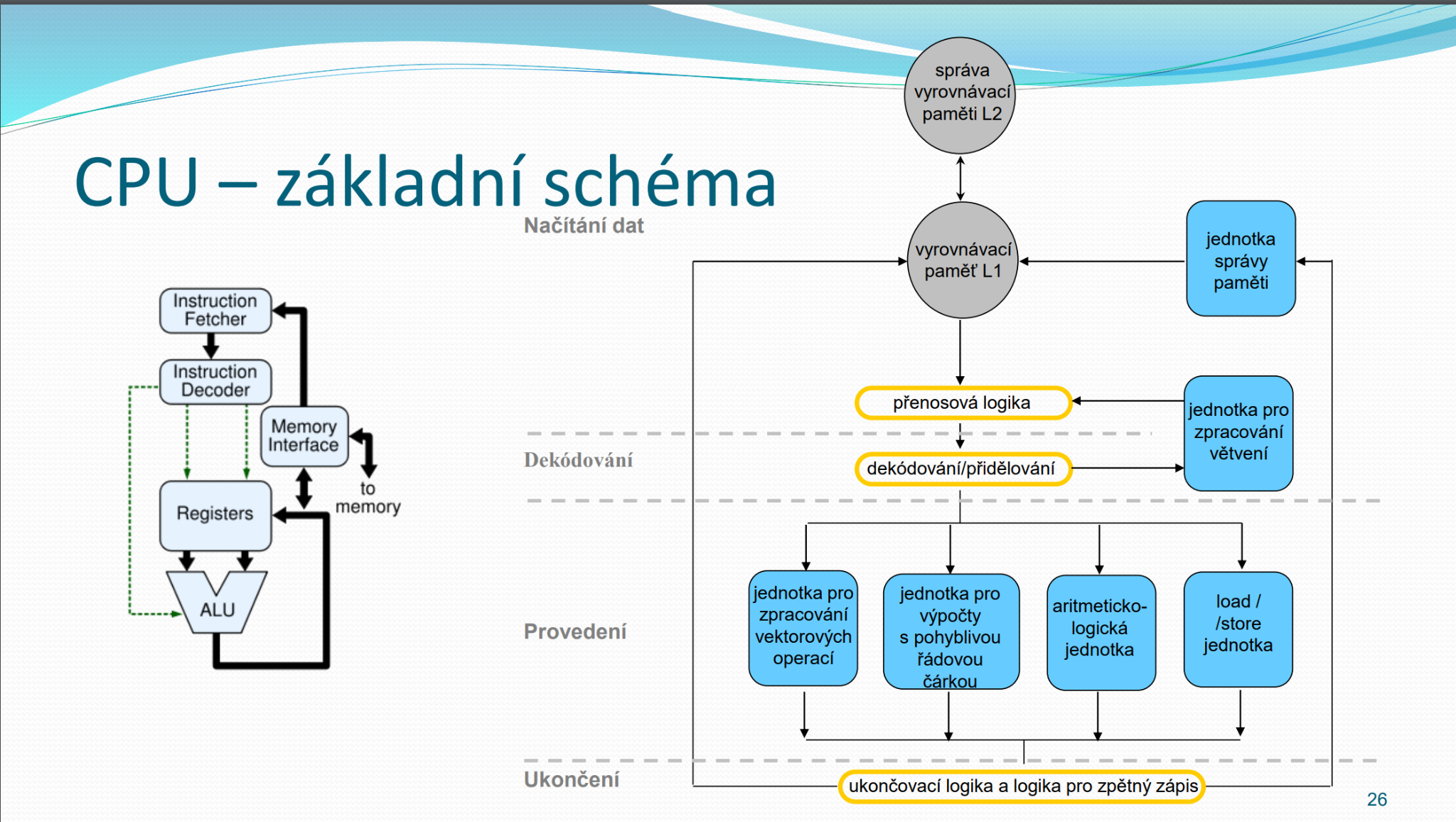
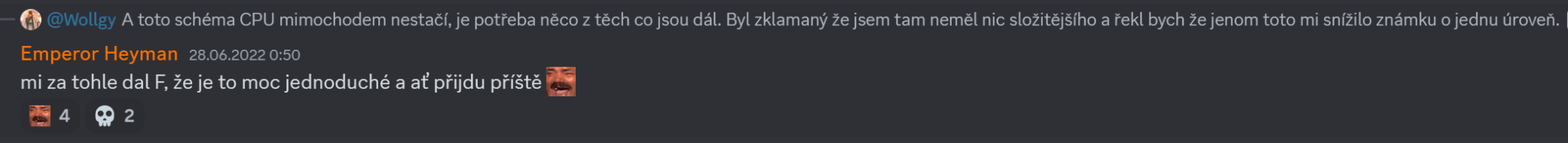
**Princip funkce a přehled**

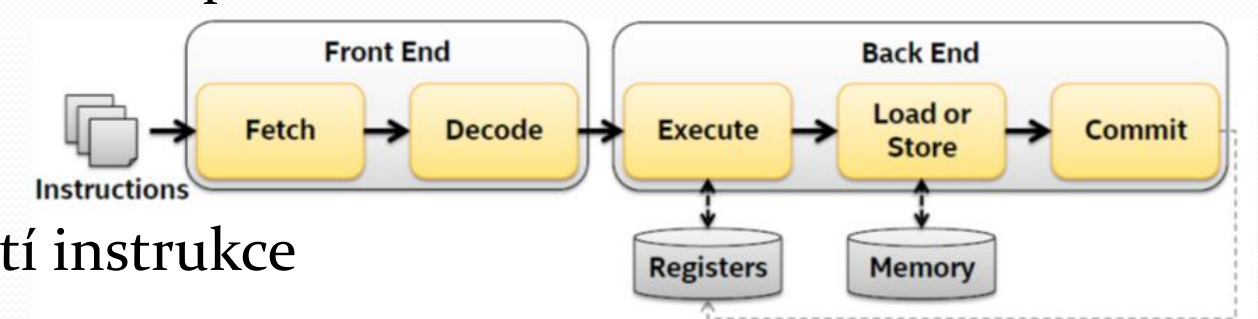
* Integrovaný obvod zajišťující výpočty, vykonává jednotlivé instrukce
* Ovlivňuje výkon celého PC do značné míry, je umístěn na základní desce v socketu
  + **Pin grid array** - piny (AMD - piny na desce)
  + **Land grid array** - dotyk kontaktních ploch (Intel - piny na CPU)
  + **Ball grid array** - připojení přímo přes základní desku
* **Parametry:** Frekvence, počet jader, šířka slova [bity], Vyrovnávací paměť (cache), Počet instrukčních kanálů, počet a typ výkonných jednotek, velikost adresovatelné paměti
* **Moorův zákon** - Zdvojnásobení počtu tranzistorů každé dva roky
* **Části procesoru a instrukce:** Řadič, dekodér instrukcí, registry, výkonné jednotky
  + Jednotlivé instrukce jsou načítány z operační paměti a přes interní sběrnici a instrukční registr putují do dekodéru instrukcí. Tam jsou dekódovány a předány do řadiče, který je třídí na jednotlivé bloky procesoru
  + Instrukce se vykonávají postupně (LOAD A, R1 LOAD B, R2,...)



Ne, ten první obrázek nestačí



* **CISC** - Komplexní instrukční sada, instrukce mají proměnlivou délku i dobu vykonávání, je tedy nutné číst i délku instrukce. Pracuje s RAM přímo, třeba přidání jedničky na nějakou adresu RAM je jedna instrukce.
* **RISC** - Redukovaná instrukční sada, stejná délka instrukcí, doba provedení vždy jeden cyklus, modernější, nepracuje s RAM přímo (najdi v RAM tohle a vezmi to do CPU, přidej jedničku, vrať to tam = tři instrukce místo jedné u CISC)
* **Instrukční kanály (pipelines)**
  + PF **(prefetch)** - výběr instrukce
  + D1 **(decode1) -** dekódování instrukce
  + D2 **(decode2)** - výpočet adresy, určí se adresy se kterými instrukce pracuje
  + EX **(execution)** - vlastní provedení instrukce
  + WB **(write back)** - zápis výsledků
  + **Instrukční kanál (pipeline)** - zpracování rozděleno na několik kroků
    - Instruction fetch (vyzvednutí instrukce), Decode, Execute, Memory access (přístup k paměti), Commit/Writeback (zápis výsledku)
    - **Subskalární zpracování** - vždy jedné instrukce
    - **Skalární zpracování** - každá část CPU jednu instrukci v určité fázi
    - **Superskalární zpracování -** více instrukčních kanálů
    - **Problémy**
      * **Datový hazard** - zpracování instrukce započne předtím, než je dokončena ta předchozí, požadované vstupy ještě tedy nejsou k dispozici
      * **Řídící hazard** - nutnost rozhodnutí před vykonáním instrukce (podmíněné skoky)
      * **Strukturální hazard** - konflikty mezi sdílenými zdroji CPU
      * **Pipeline stall** - zpoždění zpracování následující instrukce (bublina)
        + Lze zabránit - predikcí větvení, využití cache a bufferů, simultánním multithreadingem, superskalaritou,...
      * **Pipeline flush** - vyprázdnění instrukčního kanálu



* **Paralelismus**
  + TLP **[Thread level paralelism]** Na úrovni vláken (multi-CPU, multi-core)
  + ILP **[Instruction level]** Na úrovni instrukcí superskalarita, mikroarchitektura
  + DLP **[data level]** - Na úrovni dat SIMD **[Single instruction - multiple data]**
* **Moderní procesor** - System on Chip (SoC) - více částí systému do jednoho čipu, více jader, Cache, integrované GPU,...
* **Režimy práce procesoru**
  + **Reálný**
    - Základní režim x86 mikroprocesorů, stránkování není dostupné, adresový prostor začíná vždy od začátku operační paměti
  + **Chráněný**
    - **Podpora ochrany paměti**
      * **Segmentace** (každý program získá svůj vlastní adresový prostor)
      * **Stránkování** (virtuální adresový prostor s větší kapacitou, než je kapacita dostupné RAM)
  + **Režim virtuální 8o86**
    - Umožňuje v 32bitovém režimu provádět 16bitové strojové instrukce nativně
  + **Režim správy systému**
    - Provozní režim procesoru neviditelný pro aplikace i operační systém, kontrolu nad procesorem má speciální firmware s vyššími oprávněními na místo OS
  + **Long mode**
    - **Režim kompatibility**
      * Umožňuje 64bitovému OS spouštět 32bitový software bez úprav
    - **64bitový režim**
      * Umožňuje 64bitovému systému spouštět aplikace v 64bitovému adresnímu prostoru, který je rozšířen

**Architektura procesoru**

* **Hyperthreading (**Intel)
  + Současný běh více vláken, založeno na využití pipeline stalls (bublin) v instrukčním kanálu
  + Zvýšení výkonu je proměnlivé, nějaké zdroje jsou totiž sdílené, problémy s bezpečností
  + **Simultaneous multithreading** (AMD)
* **Tepelný monitor** - továrně vyladěné tepelné čidlo na čipu, velmi rychle reagující obvod tepelné ochrany **(tepelná dioda)**