

3. Počítačové architektury číslicových strojů

Von Neumann vs Harvardská architektura

Von Neumann

- Myšlenkou architektury Von Neumann je možnost ukládat instrukce do paměti s údaji, na kterých by pokyny měli fungovat.
- *Skládá se ze tří odlišných komponentů:*
- CPU (centrální procesorové jednotky), paměťové jednotky a I / O (vstupní / výstupní rozhraní).
- CPU je "srdcem" systému v počítači, který se skládá ze *tří hlavních součástí*: ALU (aritmetické a logické jednotky), CU (řídící jednotky) a registrů.

Harvardská

- Harvardská architektura vyžaduje dva samostatné procesory RAM (Random Access Memory) a ROM (Read-Only Memory)
- Vzhledem k tomu, že RAM a ROM byly oddělené, v těchto typech architektury vzrostla také potřeba hardwaru.

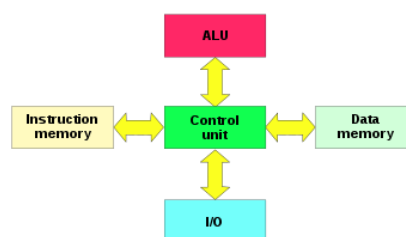
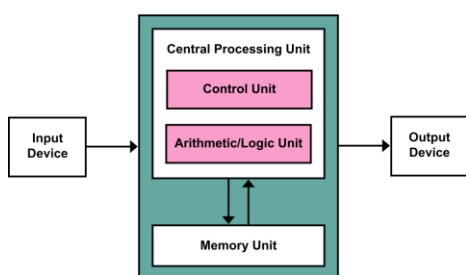
Porovnání obou архитектур

Von Neumann

- Rychlejší
- Jednoduchý design
- Stejná cesta pro instrukce a nainstalované programy
- Bere instrukce buď od uživatele nebo od programu
- Méně prostorově náročná
- Stejný prostor pro datové i programové paměti

Harvardská

- Menší
- Složitější design
- Samostatné jednotky pro instrukce a nainstalované programy
- Přijímá všechny instrukce současně
- Vyžaduje více fyzického prostoru
- Plýtvá se vnitřní pamětí kvůli tomu, že datová a programová paměť jsou uloženy odlišně



Bezpečnost a užití systému

- Týká ochrany před hrozbami a zabezpečení dat v počítači.
- Zranitelnost systému znamená slabé místo, které může být využito k útoku. Je důležité udržovat systém aktualizovaný a pokud plánujeme stahovat soubory z nedůvěřivých internetových zdrojů, tak je dobré využívat spolehlivý antivirový software.
- Obecně bychom měli být obezřetní při otevírání neznámých e-mailů, nedůvěryhodných odkazů a programů.
- Používání silných hesel je také klíčové pro ochranu účtů. Také není na škodu pravidelné zálohování dat, aby bylo možné je obnovit v případě problémů.

Synchronní a asynchronní obvod

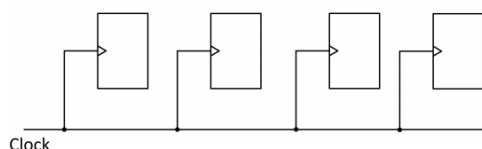
Synchronní

- pracuje synchronně (časově sladěně) s hodinovým signálem. Všechny operace jsou spouštěny na základě změn tohoto signálu.
- Výpočty jsou prováděny přesně v daný okamžik, synchronizovaný s hodinovým signálem. Kvůli přesnému časování jsou synchronní obvody snazší navrhnout a testovat.

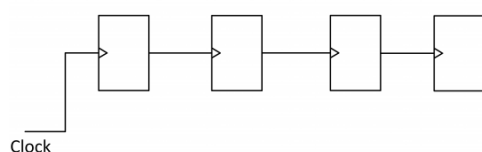
Asynchronní

- Operace v asynchronním (nesoučasným) obvodu mohou být spuštěny nezávisle na dalších částech obvodu. Návrh asynchronního obvodu může být složitější kvůli různým časovým faktorům (například kvůli různému časovému zpoždění).
- Mohou být užitečné tam, kde je důležité minimalizovat časová zpoždění nebo kde jsou vstupy/výstupy nepředvídatelné.

Synchronní čítač



Asynchronní čítač



Synchronní a asynchronní událost

Synchronní

- Při synchronním zpracování čeká program na dokončení určité činnosti, než bude pokračovat dál. To znamená, že v průběhu nějakého úkolu se zbytek programu zastaví a čeká jeho dokončení.
- Synchronní kód se vykonává v pořadí, v jakém byl napsán.
- Často vede k jednodušší logice a je snazší ladit, protože se vývojář může spolehnout na to, že kód bude prováděn v určitém pořadí a úlohy se nemůžou provádět náhodně.

Asynchronní

- Asynchronní kód nečeká na dokončení určité činnosti, ale pokračuje v provádění dalších úloh. To znamená, že program může provádět jiné operace, zatímco čeká na dokončení určité události.
- Může být napsán tak, aby se prováděl vedle jiných úloh, což vytváří nelineární tok řízení programu.

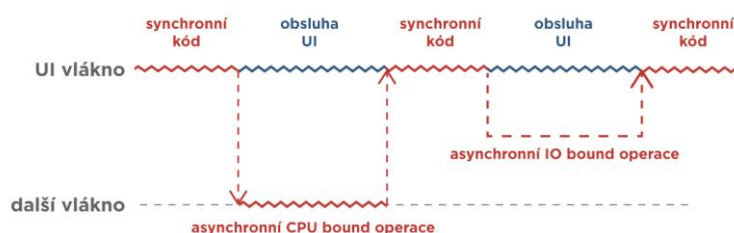
Synchronizace

Hardwarová

- Spoléhá na fyzické součásti, jako jsou oscilátory, hodiny, časovače a synchronizační signály, které jsou navrženy tak, aby poskytovaly přesné časování.
- Metody jsou obvykle preferovány v aplikacích, kde je potřebná vysoká přesnost a minimální časové zpoždění. To může být kritické v průmyslových systémech, telekomunikacích anebo v oblasti finančních transakcí.

Softwarová

- Tato synchronizace se spoléhá na algoritmy a techniky v rámci softwarového kódu. Toto zahrnuje použití zámků, semaforů, časových prodlev a komunikačních protokolů. Je obvykle jednodušší na implementaci a umožňuje vývojářům flexibilitu v řízení časování událostí.
- V operačních systémech a více vláknových aplikacích je softwarová synchronizace nezbytná pro správné řízení přístupu, jako jsou paměťové úseky.



Synchronizace vnějších signálů

- Externí signály, jako GPS nebo rádiové vysílače, poskytují přesný časový referenční bod, který slouží jako základ pro synchronizaci. Kromě času se synchronizují i frekvence signálů.
- To je důležité například při digitálním zpracování signálů nebo při práci s vysokofrekvenčními zařízeními. Přesnost externího signálu je klíčová. Přesný signál je důležitý pro správné fungování systému.
- Lze využít hardwarové prostředky, jako jsou specializované obvody pro zpracování signálů, nebo softwarové algoritmy a techniky pro řízení časování událostí. Důležitým aspektem synchronizace vnějších signálů je také minimalizace latence, aby bylo dosaženo co nejrychlejší reakce a přesnosti v systému.

Synchronizace na úrovni procesů

- Zajišťuje správné provádění paralelně běžících procesů v počítačovém systému.
- *Využívá různé techniky, jako jsou:*
 - Zámky
 - Semafor
 - Mutexy (algoritmus používaný v programování jako synchronizační prostředek)
 - Atomické operace
- Slouží k zabránění konfliktům a nekonzistentním stavům při přístupu k sdíleným zdrojům. Klíčové pro spolehlivé chování paralelního systému.

Systémy s více jádry

- Různé úlohy mohou být prováděny současně na různých jádrech, což zkracuje dobu, kterou trvá dokončení úkolů. Aplikace optimalizované pro více jádrové systémy mohou dosáhnout větší výkonnosti a rychlosti.
- Aplikace mohou běžet na různých jádrech, což minimalizuje zasekávání a zlepšuje reakční doby aplikací. Více jádrový systém může lépe využít dostupnou paměť a ostatní zdroje.
- Umožňuje běh více úloh současně s menším snížením výkonu.

