1. Kdy a proč se začaly procesory dělit na RISC a CISC? Procesory začaly být děleny na RISC (Reduced Instruction Set Computer) a CISC (Complex Instruction Set Computer) v průběhu 70. a 80. let 20. století. Přechod na tyto architektury byl motivován snahou zlepšit výkon a efektivitu procesorů.
2. Jaké byly zásadní důvody, proč se začaly procesory RISC vyvíjet? Existuje několik zásadních důvodů, proč se začaly vyvíjet procesory RISC:
   * Zjednodušení instrukční sady: RISC procesory používají jednoduché a uniformní instrukce, což usnadňuje návrh a implementaci procesoru.
   * Zvýšení výkonu: RISC architektura umožňuje vykonávat jednoduché instrukce rychleji a efektivněji, například díky možnosti pipeliningu a snížení latence instrukcí.
   * Zvýšení možnosti optimalizace: Jednoduchá instrukční sada RISC procesorů usnadňuje kompilátorům generovat efektivní kód a optimalizovat ho pro různé úlohy.
   * Snížení složitosti hardwaru: RISC procesory mají obecně jednodušší hardwarovou strukturu, což zjednodušuje návrh a zvyšuje spolehlivost.
3. Jaké jsou základní konstrukční vlastnosti procesorů RISC? Základní konstrukční vlastnosti procesorů RISC zahrnují:
   * Jednoduchá instrukční sada: RISC procesory mají malý počet jednoduchých instrukcí, které jsou často omezené na jednoznačné operace.
   * Pevná délka instrukcí: Instrukce mají často pevnou délku, což usnadňuje jejich dekódování a zpracování.
   * Registr-centricita: RISC procesory mají často velký počet registrů, které jsou rychle přístupné a slouží jako hlavní paměťové úložiště pro operace.
   * Load-store architektura: RISC procesory vyžadují, aby přístup k paměti probíhal přes zvláštní instrukce pro načítání (load) a ukládání (store) dat.
4. Jak přispěly jednotlivé charakteristické vlastnosti procesorů RISC ke zvýšení výpočetního výkonu? Jednotlivé charakteristiky procesorů RISC přispívají ke zvýšení výpočetního výkonu tímto způsobem:
   * Jednoduché instrukce a pevná délka instrukcí umožňují rychlé dekódování a zpracování instrukcí.
   * Registr-centricita umožňuje rychlejší přístup k datům a minimalizuje přístupy do paměti.
5. Jaký je princip zřetězeného zpracování instrukcí v RISC procesorech? Princip zřetězeného zpracování instrukcí v RISC procesorech spočívá v rozdělení vykonávání instrukcí do několika fází, které se provádějí současně. Každá fáze zpracování instrukce (např. dekódování, provedení, zápis do registru) se provádí na samostatném funkčním bloku v procesoru. Když jedna instrukce projde první fází, další instrukce se začnou vykonávat na dalších fázích.
6. Jakého zrychlení lze zřetězeným zpracováním instrukcí dosáhnout? Zřetězené zpracování instrukcí umožňuje dosáhnout zvýšení výkonu tím, že se instrukce vykonávají paralelně a nečekají na dokončení předchozí instrukce. Tím se využívá plně kapacita procesoru a snižuje se doba potřebná k vykonání sekvence instrukcí.
7. Jaké problémy přináší zřetězené zpracování instrukcí v procesorech RISC? Zřetězené zpracování instrukcí v procesorech RISC může přinést několik problémů, například:
   * Závislosti instrukcí: Pokud instrukce závisí na výsledku předchozí instrukce, může dojít k tzv. hazardům, které vyžadují přerušení zřetězeného zpracování nebo provedení složitějších technik, jako je přejmenování registrů nebo přerušování toku instrukcí.
   * Paměťové závislosti: Pokud instrukce přistupuje k paměti, může dojít k závislostem na době přístupu k paměti, což může snížit výkon zřetězeného zpracování.
8. Co to je predikce skoků, proč se používá a jaké způsoby predikce se využívají? Predikce skoků je technika používaná v procesorech k předvídání výsledku skokových instrukcí (např. podmíněné skoky, skoky do podprogramů). Predikce skoků se používá k minimalizaci zpoždění způsobeného výpočtem cílové adresy skoku. Existují různé způsoby predikce skoků, včetně statické predikce, jednobitové dynamické predikce a dvoubitové dynamické predikce.
9. Co to jsou datové a strukturální hazardy v RISC procesorech? Co je způsobuje? Datové hazardy v RISC procesorech se vyskytují, když instrukce závisí na datu, které ještě není k dispozici, protože předchozí instrukce ještě nedorazila do fáze zápisu. Strukturální hazardy se vyskytují, když instrukce Strukturální hazardy v RISC procesorech se vyskytují, když instrukce potřebují přístup ke stejnému hardwarovému prostředku ve stejném časovém okamžiku, což není možné. Například, když dvě instrukce vyžadují přístup ke stejnému registru nebo stejnému funkčnímu bloku procesoru. Datové hazardy mohou být způsobeny různými situacemi, například:

* Závislost na datech (data dependency): Když instrukce čeká na dostupnost dat, která jsou produkována jinou instrukcí v předchozí fázi. Existují tři typy závislostí na datech: závislost na čtení (read-after-read, RaR), závislost na zápisu (write-after-read, WaR) a závislost na zápisu (write-after-write, WaW).
* Závislost na řídících instrukcích (control dependency): Když instrukce čeká na výsledek skokové instrukce nebo na jinou řídící informaci, která je generována jinou instrukcí. Datové hazardy a strukturální hazardy mohou způsobit potřebu přerušit zřetězené zpracování instrukcí a provést dodatečné kroky, aby se zajišťovala korektnost výpočtu. Kompilátory a hardwarové mechanismy, jako je přejmenování registrů nebo předběžné načítání (pre-fetching) instrukcí, se často používají k minimalizaci dopadu hazardů na výkon procesoru.

1. Jak funguje dvoubitová dynamická predikce skoků a proč se využívá? Dvoubitová dynamická predikce skoků je technika predikce skoků, která se často používá v RISC procesorech. Tato technika ukládá historii provedených skoků a jejich výsledků (bránice skoku) a na základě toho předvídá, zda bude skok proveden nebo nikoliv. Při dvoubitové dynamické predikci je použitý jednobitový stavový automat se dvěma bity pro každou bráničku skoku. Stavový automat může mít čtyři možné stavy: not-taken (nebránit se skoku), strongly not-taken (silně nebránit se skoku), taken (bránit se skoku) a strongly taken (silně bránit se skoku). Na základě aktuálního stavu bránice skoku se předpovídá výsledek skokové instrukce. Dvoubitová dynamická predikce skoků se používá k minimalizaci výskytu skokových hazardů a optimalizaci vykonávání skokových instrukcí. Předpověď výsledku skokové instrukce umožňuje procesoru načíst instrukce z cílové větve a pokračovat v zpracování, aniž by čekal na potvrzení skoku. To může výrazně zlepšit výkon procesoru a využít plně jeho zřetězenou architekturu. Dvoubitová dynamická predikce skoků funguje tak, že při každém provádění skokové instrukce se aktualizuje stavový automat bránice skoku na základě skutečného výsledku skoku. Pokud předpověď byla správná, stavový automat se posouvá směrem k "strongly taken" stavu. Pokud předpověď byla nesprávná, stavový automat se posouvá směrem k "strongly not-taken" stavu. Tímto způsobem se stavový automat adaptuje na vzory skoků a zlepšuje přesnost predikce. Dvoubitová dynamická predikce skoků je oblíbená, protože poskytuje poměrně dobré výsledky s relativně jednoduchým hardwarovým řešením. Je však důležité si uvědomit, že předpověď skoků není vždy 100% přesná a může vést k neefektivním výpočtům, pokud dojde k nesprávné predikci skoku. Proto jsou v moderních procesorech používány i další pokročilé techniky predikce skoků, které se snaží zlepšit přesnost predikce a minimalizovat výskyt skokových hazardů.