

Posudek vedoucího diplomové práce

Student: Bc. Karel Kočí

Název: Simulátor MIPS procesoru

Znalost principu činnosti procesorů je důležitá ne jen pro ty, kdo nové procesory navrhují, ale i pro jejich uživatele, programátory. Umožňuje jim pochopit jakým způsobem je algoritmus realizovaný na úrovni hardware a jaká omezení pro propustnost z toho vyplývají a které operace a jejich kombinace jsou časově nákladné. Tato znalost pak umožňuje optimalizovat jak algoritmy tak slouží jako vodítko při výběru datových struktur. Pro seznámení s principy činnosti procesoru je pro studenty přínosné použít simulátor, který stav procesoru a vykonávání instrukcí zobrazuje. Vytvoření takového simulátoru s názorným zobrazením stavů a signálů bylo zadáním hodnocené diplomové práce.

Vybraná procesorová architektura MIPS je dostatečně jednoduchá pro seznámení s problematikou a přitom na ní lze snadno demonstrovat základní techniky pro zvýšení propustnosti počítačového systému. Navržený simulátor nabízí volbu mezi jednocyklovou variantou a variantou se zřetězeným pětistupňovým zpracováním instrukcí včetně grafického zobrazení hodnot řídicích signálů i zpracovávaných hodnot v datové cestě procesoru. Dále je možné volit zařazení datové a instrukční paměti cache a pro zřetězenou variantu různé možnosti ošetření vzájemné závislosti instrukcí.

Student zvolil pro realizaci vhodnou grafickou knihovnu (Qt5), která zajišťuje programu velmi dobrou přenositelnost mezi různými operačními systémy a platformami. Implementace striktně odděluje vlastní jádro simulátoru od grafické části aplikace. Zdrojové kódy které spravují stav „počítače“ a zpracování instrukcí se nacházejí v adresáři `qtmips_machine` a výsledkem kompilace je sdílená knihovna `libqtmips_machine`. Implementace je velmi přehledná a pro informaci o změnách stavu využívá konceptu signál-slotu z knihovny Qt, který vyžaduje použití třídy `QObject`. Kromě využití Qt řetězců a výjimek je v této části implementace patrná snaha o udržení maximální nezávislosti na nadstavbách jazyka C++. Jádro simulátoru je využité v grafické verzi simulátoru i alternativní aplikaci, kterou je možné spouštět z příkazové řádky. Varianta bez grafického rozhraní je využita i pro automatické testy jádra simulátoru.

Dále kladně hodnotím vysokou konfigurovatelnost emulátoru (s/bez cache, zřetězení, ošetření hazardů), které je dosaženo s využitím jednotné implementace základních bloků bez zbytečného kopírování a úprav kódu pro jednotlivé varianty. Dobře je i zvládnuté zobrazení bohatě konfigurovatelné instrukční a datové cache, které opět využívají společného základu. Podobně je řešeno i zobrazení paměti v režimu dat a instrukcí. Blokovaná schémata jsou při dostatku prostoru na obrazovce škálovatelná, protože zobrazování objektů a signálů je řešeno vektorově.

Vlastní text práce je relativně krátký, ale architektura MIPS je dopodrobna popsána v mnoha pramenech a tak krátký úvod považuji za přiměřený. Přínosem je porovnání s alternativními simulátory, popis využití ve výuce a vlastního řešení aplikace. Popis řešení zabírá téměř polovinu celého textu. Aplikace pak představuje okolo 10 tisíc řádek zdrojového kódu a rozsah odvedené práce a její profesionalitu považuji za bezezbytku splňující požadavky kladené na diplomovou práci. Práci **doporučuji k obhajobě** a hodnotím ji klasifikačním stupněm **velmi dobře (B)**.

Práce má předpoklad pro využití ve výuce, zároveň již při zadání bylo naplánované, že v rámci návazného projektu bude zaintegrovaná zpětná vazba. Pro tuto etapu již byly přislíbeny finanční prostředky. V tuto chvíli chybí zobrazení propojení pro JAL instrukci, návrh vhodného zobrazení výstupů jednotky ošetření hazardů a implementace přerušení běhu v ladících bodech.

V Praze, dne 28. 5. 2018

Ing. Pavel Píša, Ph.D.
Katedra řídicí techniky
Fakulta elektrotechnická
České vysoké učení technické