

I. IDENTIFIKAČNÍ ÚDAJE

Název práce:	Funkční a Formální Verifikace Bloku Jump Controller pro RISC-V Procesor
Jméno autora:	Jiří Šindelář
Typ práce:	diplomová
Fakulta/ústav:	Fakulta elektrotechnická (FEL)
Katedra/ústav:	Katedra mikroelektroniky
Oponent práce:	Ing. Michal Kajan
Pracoviště oponenta práce:	ASICentrum spol. s r. o.

II. HODNOCENÍ JEDNOTLIVÝCH KRITÉRIÍ

Zadání	náročnější
<i>Hodnocení náročnosti zadání závěrečné práce.</i>	
Zadání diplomové práce hodnotím jako náročnější, protože vyžaduje nastudování technik funkční verifikace založené na simulaci a také technik formální verifikace s využitím formálních vlastností a tvrzení (angl. <i>properties</i> a <i>assertions</i> – <i>Formal Property Verification</i> , dále v posudku jako <i>FPV</i>) pomocí vhodných jazyků, knihoven a softwarových nástrojů.	

Splnění zadání	splněno s většími výhradami
<i>Posuďte, zda předložená závěrečná práce splňuje zadání. V komentáři případně uveďte body zadání, které nebyly zcela splněny, nebo zda je práce oproti zadání rozšířena. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.</i>	
Zadání práce je formulováno dostatečně všeobecně a dává studentovi velkou míru nezávislosti a svobody, jak konkrétně se rozhodne dané téma řešit. Jednotlivé body zadání jsou v práci pokryty a řešeny, ale výhrady mám ke způsobu a přístupu k řešení jednotlivých bodů zadání:	
<ul style="list-style-type: none"> • Řešení prvního bodu a druhého bodu zadání se odráží v úrovni zpracování teoretické části závěrečné práce, kde se student dopouští nepřesností a nevhodných zjednodušení (více v posudku níže, v části Odborná úroveň). Student ze seznamu literatury v zadání využil pouze 1 ze 4 doporučených zdrojů ke studiu, a tím byla bakalářská práce Bc. Martina Laštovky, jehož zdrojové kódy využil ve své praktické části. • Při řešení třetího bodu si dal student v textu více záležet na popisu svého řešení a jeho implementaci. Student ale nepopisuje, jak jednotlivé požadavky z verifikačního plánu pokrýl implementací, alespoň s odkazem na název <i>assertions</i> do zdrojových kódů. Také chybí provázání, kterými testy pro dynamickou simulaci byly úspěšně pokryty požadavky z verifikačního plánu. Testů je velmi málo – tři. Zdrojové kódy dostupné jsou a je vidět, že student nějakou práci odvedl. • K řešení posledního bodu mám také výhrady. V sekci Motivation si student sám stanovil: „During the verification process, we will monitor the effort spent on each of the methods and then compare them.“ Očekával bych, že student si během řešení zaznamená orientačně alespoň čas strávený implementací. V závěru práce to ale diskutováno není. Student nezvolil žádná kritéria, kdy považuje verifikaci za hotovou a žádné měřitelné metriky pro výsledné porovnání dvou verifikačních přístupů. V části 6.2 Comparing the verification approaches využívá k porovnání jen velmi vágní a neurčité pojmy. Kladně ale hodnotím, že se studentovi povedlo najít i několik chyb ve verifikovaném bloku a snažil se popsat jejich povahu a odhadnout závažnost. 	

Zvolený postup řešení	částečně vhodný
<i>Posuďte, zda student zvolil správný postup nebo metody řešení.</i>	
Student si při řešení zadání definoval stanovenou sérii kroků pro oba přístupy k verifikaci bloku prediktoru skoků, ale nedefinoval si kritéria, na základě kterých by mohl určit, že je proces verifikace ukončen – a to ani pro jeden z přístupů. Proces verifikace, nezávisle na použité metodě, vyžaduje stanovení měřitelných kritérií, aby ho mohlo být možné řídit a určit, kdy je možné ho ukončit.	
Také si nedefinoval měřitelná kritéria pro vzájemné porovnání využitých verifikačních přístupů. Nedostatečně vysvětluje pojem pokrytí (coverage). V práci je využito pouze funkční pokrytí, ale není to explicitně uvedeno. Zároveň mohlo být využito i pokrytí strukturální – tj. zdrojového kódu verifikovaného bloku s vhodnými metrikami (bloky, výrazy, ...), které mohlo efektivně posloužit k výslednému porovnání využití obou verifikačních metod, a to na základě konkrétních a ověřitelných dat. Čtenář během čtení závěrečné práce nabývá dojem, že se student pustil do implementace a při nalezení několika chyb	

se s takovým výsledkem jednoduše spokojil. Pro funkční verifikaci implementoval student pouze 3 testy. Není uvedeno, zda je tento počet již postačující. Pro část zabývající se FPV analýzou není uveden celkový přehled toho, co bylo implementováno.

Odborná úroveň

E - dostatečně

Posuďte úroveň odbornosti závěrečné práce, využití znalostí získaných studiem a z odborné literatury, využití podkladů a dat získaných z praxe.

Předložená práce má jen dostatečnou odbornou úroveň. V teoretické části práce v sekcích zaměřených především na popis technik formální verifikace se student dopouští nepřesností a nesrovnalostí, využívá víc své dojmy než fakta z citovatelných zdrojů a z textu práce je celkově poznat, že využití zdroje nastudoval narychlo a v rozsahu jen nejnútnejším pro implementační část a dosažení alespoň minimálních výsledků.

Tvrzení v sekci Motivation na straně 3 je nepřesné: „The functional properties, defined by the verification engineer, are designed to specify the good behavior of the design, and when proven by the formal methods, they tell us that no allowed stimulation of the design will cause it to behave poorly.“ Tzv. *properties* slouží k abstrakci a popisují očekávané chování obvodu nebo i chování, které nesmí nikdy nastat. Výsledek aplikace nástrojů pro FPV na tyto konstrukty je ale platný jen v jejich rozsahu a taková verifikace bude jenom tak dobrá, jak dobře jsou dané *properties* a *assertions* napsány. Nezaručuje to ale, že za jiných podmínek, které nejsou pomocí *properties* a *assertions* popsány, bude testovaný blok fungovat správně. Pro posouzení této dostatečnosti mohou sloužit například další metriky – např. pokrytí zdrojového kódu, které ale v práci využity nebyly. Je škoda, že se jim student nevěnoval.

V části 1.1.4 Blackbox vs whitebox student nevhodně popisuje *assertions*: „Assertions are not synthesizable code, because of that, they need to be wrapped in a synthesizer directive that disables the synthesis of code.“ Žádné speciální direktivy nejsou pro jazyky PSL a SVA, použité v této práci, potřebné. Takové direktivy jsou používány pro takzvaný pomocný kód pro *assertions*, který pro syntézu není nijak relevantní. To však v práci zmíněno není.

Tvrzení v sekci 1.2 na straně 13 je nepravdivé: „The positive of the formal verification approach is that all of the design bugs caused by corner-case states of the circuit are found.“ Nástroje založené na FPV toto samy o sobě nezaručí a vždy je nutný vhléd verifikačního inženýra, aby přemýšlel nad vhodným popisem chování obvodu pomocí *properties* a *assertions* tak, aby takové chyby formální nástroje odhalily.

V sekci 1.2 na straně 13 student rozděluje formální metody na dvě třídy: „The formal methods can be categorized into two classes, which are formal equivalence checking (FEC) and formal property verification (FPV).“ Přístupů je víc, ne jen dva.

Část 1.2.4 tvrdí: „The problem with formal verification is that there are lots of languages that can be used.“ Student zřejmě vychází ze staršího zdroje, který v dané sekci cituje. Ale daný text už neodráží současnou realitu. Jazyky samotné pro formální verifikaci (zmiňovanou FPV) problémem nejsou a v praxi jsou využívány ty, které mají velmi dobrou podporu v komerčně dostupných nástrojích. To lze odvodit i ze studie, kterou student používá jako jeden z citovaných zdrojů. *Open Verification Library* navíc není jazyk.

Zcela chybí popis pojmů „assertion“ a „property“, které jsou stěžejní pro využití formální verifikace založené na FPV, které nakonec student sám používá a implementuje jako technický výstup své práce. Student zcela opomenul problém stavové exploze a problém při analýze a dokazování *assertions*, který vede za daných nastavení k jejich nerozhodnutému stavu (angl. undetermined – ani dokázáno, ani dostupný protipříklad). Student při řešení své práce na takový případ nenarazil a nemusel ho řešit třeba dalšími technikami, byť pokročilými, které by tuto práci obohatily.

V části zaměřené na FPV je pojem a koncept „UVC“ použit nevhodně. UVC je zkratka pro UVM Verification Component a používá se pro funkční verifikaci s využitím knihovny UVM. Vhodné označení pro modul s *assertions* je verifikační komponenta, angl. verification component. Přiložené implementované soubory mají dokonce příponu *vcomp*. Použití označení „UVC“ pro účely FPV analýzy v textu práce je zavádějící a matoucí.

Některé sekce by mohly být, pro lepší pochopení a orientaci, umístěny logicky v textu jinak. Např. sekce 1.1.3 Constrained random stimuli mohla následovat hned za sekci 1.1 Functional verification, ve které jsou právě vstupy do testovaného bloku popisovány.

V práci postrádám přehled výsledků pro jednotlivé testy po dokončení jejich simulace. Také v práci postrádám přehled výsledků po dokončení FPV formální analýzy v nástroji Jasper.

Formální a jazyková úroveň, rozsah práce

D - uspokojivě

Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku.

Text práce je psán v anglickém jazyce. Jazyková úroveň práce je uspokojivá. Text obsahuje zbytečné překlepy, gramatické chyby a těžkopádné až nesrozumitelné větné konstrukce. Popisovaná problematika by si zasloužila vyšší odbornou úroveň a finální jazykovou korekturu, ke které nedošlo. Text se čte velmi složitě, obsahuje přílišná zjednodušení a jen čtenář dostatečně znalý řešené problematiky se v něm neztratí.

Obsah a seznam obrázků a tabulek by mohl být kompaktnější pro jednodušší orientaci v celé práci během čtení a dohledávání kapitol a podkapitol. Tyto úvodní sekce působí dojmem, že byly roztaženy s rozšířenými mezerami mezi řádky jen pro dosažení vyššího celkového počtu stránek.

Obrázky 1.1, 2.1 a 2.2 nejsou odkazovány z textu. V úvodní části chybí správný odkaz na bakalářskou práci Bc. Martina Laštovky, která byla využita při řešení.

Obrázek 1.2, který zobrazuje blokové schéma prostředí UVM, nese jiné označení bloků, než popisuje text.

Strana 20 obsahuje český odkaz v anglickém textu: „viz.: Table 2.2“.

Zkratka CPU na straně 32 není napsána jako zkratka (je uvedena jako Cpu) a není v seznamu zkratk.

Text práce bez příloh je dle mého názoru v očekávaném rozsahu. Práce by si ale s ohledem na složitější téma a při pečlivém zpracování zasloužila mírně větší rozsah, než má.

Výběr zdrojů, korektnost citací

E - dostatečně

Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení závěrečné práce. Charakterizujte výběr pramenů. Posudte, zda student využil všechny relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků a úvah, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami.

Student využil pouze 1 ze 4 doporučených zdrojů k úvodnímu studiu. Proč?

Student si dohledal samostatně další zdroje, které hodnotím jako dostatečné zejména pro praktickou realizaci – implementační část práce, kde student prokázal, že dokáže použít knihovnu UVM, vytvořit testovací prostředí v jazyku SystemVerilog, vytvořit moduly s konstrukty *property* a *assert* a použít nástroje Xcelium a Jasper.

Pro teoretickou část práce by bylo vhodné použít i další zdroje zaměřené celkově na principy funkční a formální verifikace, nejen ty zaměřené na konkrétní jazyky, knihovny a rozhraní. Zejména první knihu ze seznamu doporučené literatury. A také věnovat dostatek času jejich nastudování. Teoretická část práce by takto mohla obsahovat více odkazů pro podložení prezentovaných tvrzení, aby bylo možné jasně odlišit vlastní dojmy a úvahy od skutečného současného stavu poznání a současné praxe.

Další komentáře a hodnocení

Vyjádřete se k úrovni dosažených hlavních výsledků závěrečné práce, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, publikačním výstupům, experimentální zručnosti apod.

Student správně popisuje důležitost verifikačního plánu a jak ho aspoň v hrubých rysech sestavit a co má být jeho obsahem. Jeho plán však postrádá konkrétní položky, čím daný požadavek z plánu pokryl – jakým testem (pro funkční verifikace), či jaký *assertion* použil (pro FPV).

Příloha práce by měla obsahovat nejen tabulku s verifikovanými požadavky a jakou z metod jsou pokryty, ale také čím přesně jsou pokryty. Chybí trasovatelnost do implementační části a priority, dle kterých postupoval.

Zdrojové kódy jsou strukturované a kód je vhodně odsazen. Zdrojové soubory ale neobsahují žádnou hlavičku, obsahují málo komentářů pro lepší orientaci a pochopení jeho smyslu. Nejdůležitější soubor pro FPV *rise_covers_vcomp sv* obsahuje mnoho zakomentovaného kódu (zhruba 40% celkového počtu využitých řádků) – k čemu takový kód slouží?

Není uvedeno, jakou verzi nástrojů student použil. Příložené skripty pro spuštění neobsahují ani stručný návod, jak je použít a jak nakonfigurovat prostředí, ze kterého mají být spuštěny.

Navržené řešení je funkční – student objevil několik chyb ve verifikovaném bloku. Část funkční verifikace by si ale zasloužila vyšší počet testů než 3 nebo dostatečné vysvětlení, proč je tento nízký počet postačující. Část pro FPV nepopisuje přesně, kolik práce bylo odvedeno.

III. CELKOVÉ HODNOCENÍ, OTÁZKY K OBHAJOBĚ, NÁVRH KLASIFIKACE

Shrňte aspekty závěrečné práce, které nejvíce ovlivnily Vaše celkové hodnocení. Uvedte případné otázky, které by měl student zodpovědět při obhajobě závěrečné práce před komisí.

I přes výše uvedené výhrady doporučuji posuzovanou práci k obhajobě.

K obhajobě mám následující otázky:

- 1. V části Components na str. 10 popisujete blok Scoreboard. Vysvětlete prosím vámi uvedený příklad s ohledem na funkci této komponenty pro kontrolu výstupu verifikovaného bloku (DUT): „An example of a situation where a difference is expected is when DUT should present a random number (RN) to the interface. In this case, the predicted response can be corrected with the actually received RN before comparing.“**
- 2. Proč je důležité v procesu aplikace FPV sledovat i funkční pokrytí modelovaného chování verifikovaného bloku, zejména na začátku tohoto procesu? Co nám nepokryté chování (popsané uživatelsky pomocí sady příkazů cover property ...) může napovědět? Využil jste automaticky generované pokrytí, které nástroj Jasper sám generuje pro uživatelsky implementované *assertions*?**

Předloženou závěrečnou práci hodnotím klasifikačním stupněm **E - dostatečně**.

Datum: 31.1.2024

Podpis: