Druhá část projektu UART – Implementace a ladění

1. Pro vypracování druhé části projektu budete potřebovat nástroje pro simulaci a syntézu číslicových obvodů popsaných v jazyce VHDL. Vyžadovány jsou konkrétně open-source nástroje GHDL a GTKWave. Více informací o tom, jak nástroje získat a používat je uvedeno v následující kapitole.
2. Z informačního systému si stáhněte ZIP archiv zdrojových souborů a seznamte se s jeho obsahem. V archivu naleznete hlavně tyto tři VHDL zdrojové soubory:
   - *uart_rx.vhd* – zdrojový soubor v jazyce VHDL obsahující definici rozhraní (entity) komponenty UART_RX a prázdnou architekturou na doplnění vaši implementace,
   - *uart_rx_fsm.vhd* – zdrojový soubor v jazyce VHDL, který bude sloužit pro popis konečného automatu řídicího ostatní komponenty vašeho obvodu UART_RX,
   - *testbench.vhd* – zdrojový soubor v jazyce VHDL, který reprezentuje ukázkové zapojení pro testování základní funkcionality vámi navržené a implementované komponenty UART_RX v prostředí simulace.
3. Navržený RTL obvod z první části projektu implementujte v jazyce VHDL a uložte do předpřipraveného souboru *uart_rx.vhd*. Kód odpovídající konečnému automatu vložte pro přehlednost do samostatného souboru *uart_rx_fsm.vhd*.
4. Proveďte syntézu a simulaci vašeho VHDL kódu spuštěním připraveného skriptu *uart.sh*. Skript využívá program GHDL pro analýzu, syntézu a simulaci VHDL kódu, programem GTKWave pak zobrazuje průběh simulace (tzv. WaveForm). Na základě textových výstupů programu GHDL v terminálu nejdřív opravte nalezené syntaktické a sémantické chyby a simulaci opakujte. Nakonec zkontrolujte průběh simulace v grafickém okně s ohledem na správnou funkčnost implementovaného obvodu.
5. Doplňte technickou zprávu z první části projektu o snímek obrazovky (Print Screen) programu GTKWave s ukázkou časového průběhu simulace zachycujícího přenos alespoň jednoho datového slova. Jestliže jste se při implementaci odchýlili od vašeho původního návrhu z první části projektu, upravte také nákresy RTL architektury obvodu a grafu přechodů konečného automatu. Finální verze zprávy by měla souhlasit s vytvořenou implementací. Zachovejte formát a obsah zprávy podle ukázky v příloze.
6. Výstupy druhé části projektu budou tvořit:
   - doplněné zdrojové soubory v jazyce VHDL (*uart_rx.vhd* a *uart_rx_fsm.vhd*),
   - soubor *zprava.pdf* s výstupní zprávou ve formátu PDF.

   Všechny tři uvedené soubory zabalte do ZIP archivu s názvem ***<login>.zip***. Archiv odevzdejte skrze informační systém, do termínu pro druhou část projektu.

Před odevzdáním vašeho archivu do informačního systému si ho prosím otestujte skrze zveřejněnou sadu testovacích skriptů připravených v souboru *test.zip*. Podrobný návod na otestování naleznete v přiloženém README souboru.

Otestovaný archiv odevzdejte prostřednictvím informačního systému nejpozději do data uvedeného v informačním systému u termínu pro druhou část projektu. Pozdější odevzdání nebude bráno v úvahu.

Simulace a syntéza obvodu

Můžete si zvolit jednu ze tří následujících možností překladového prostředí pro vypracování a testování druhé části projektu.

Virtuální stroj

Pro účely vypracování projektu jsme pro vás připravili obraz disku virtuálního stroje, kde jsou nainstalovány potřebné nástroje. Obraz obsahuje zejména instalace GHDL a GTKWave. Naistalován je také výukový program LogiSim, který byl zmiňován na přednáškách.

Archiv (2,3 GB) s obrazem disku virtuálního stroje si stáhněte z privátních stránek FITkitu. VMDK soubor (7,3 GB) z archivu vybalte pomocí aplikace 7z.

Pro spuštění virtuálního stroje si nainstalujte volně dostupný program VirtualBox. Při tvorbě VM použijte nastavení Ubuntu (64-bit), alespoň 1024 MB paměti, stažený VMDK disk.

Lokální instalace

Nástroje GHDL a GTKWave jsou dostupné jako open-source bez potřeby jakékoli licence pro jejich používání. Možná je proto jejich jednoduchá přímá instalace na vašem stroji.

Na operačních systémech typu **Linux** by měla být instalace možná přímo standardním balíčkovacím nástrojem (např. *apt*, *yum*, *pacman*). Nainstalujte si nástroj GHDL (balíček *ghdl*) alespoň ve verzi 1.0.0 a nástroj GTKWave (balíček *gtkwave*) alespoň ve verzi 3.3. Ujistěte se, že cesta k nástrojům je součástí proměnné prostředí PATH a tedy je možné přímo v terminálu volat příkazy *ghdl* a *gtkwave*.

Na operačních systémech typu **Windows** a **MAC OS** je potřebná u obou nástrojů ruční instalace nebo překlad ze zdrojových souborů. Pro GHDL jsou dostupné instalační archivy a návod instalace. Pro GTKWave je dostupný archiv pro Windows a archiv pro MAC OS. Na Windows si navíc nainstalujte Bash terminál například skrze prostředí Cygwin, MinGW nebo WSL. Ujistěte se, že cesta k nástrojům je součástí proměnné prostředí PATH a tedy je možné přímo v terminálu volat příkazy *ghdl* a *gtkwave*.

Překladový server

Na fakultě je skrze SSH spojení dostupný server *fitkit-build.fit.vutbr.cz*, na kterém jsou potřebné nástroje GHDL a GTKWave již nainstalovány. Pro vzdálené zobrazení grafického okna programu GTKWave je potřeba mít u připojení správně nastavený X11 forwarding.

Server je navíc umístěn v doméně fit.vutbr.cz a umožňuje připojení jen přímo ze sítě fakulty nebo vzdáleně skrze fakultní VPN. Postupujte proto dle návodu uvedeného na následujících stránkách a vždy před přístupem k serveru mimo síť fakulty si VPN spojení aktivujte.

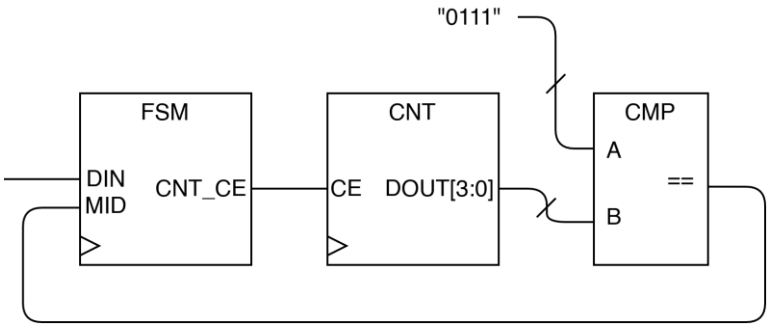
Uvědomte si také, že v ročníku vás je několik stovek a sdílený server nemusí zvládnout nápor všech studentů najednou. Upřednostněte proto jednu z předešlých dvou variant a na vzdálený server se plně nespoléhejte. Nemusí být dostupný nebo plně responzivní zejména v čase těsně před termínem odevzdání projektu.

Příloha: Výstupní zpráva (Ukázka)

Jméno:
Login:

Architektura navrženého obvodu (na úrovni RTL)

Schéma obvodu



Poznámky:

- Pro přehlednost CLK a RST signály ve schématech uvádíme, ale nemusíme zapojovat.
- Několik jednobitových D-KO můžete pro přehlednost spojit do jednoho vícebitového registru, pokud tedy sdílí všechny kontrolní signály jako CLK, RST, nebo CE.
- Jednotlivé vodiče můžete spojovat do vícebitových sběrnic.

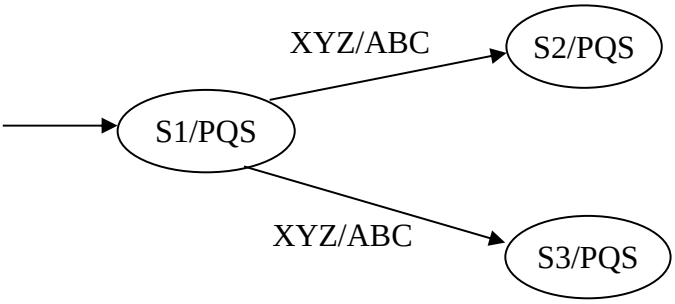
Popis funkce

Stručný slovní popis struktury a funkce obvodu (max. polovina strany A4).

Návrh automatu (Finite State Machine)

Schéma automatu

Legenda:
- Stavy automatu: S1, S2, S3
- Vstupní signály: X, Y, Z
- Mealyho výstupy: A, B, C
- Moorovy výstupy: P, Q, S



Poznámky:
- Použijte vhodné názvy pro stavy, vstupní a výstupní signály tak, aby byl snáze pochopitelný jejich význam.
- Za vstupně/výstupní signály XYZ, ABC a PQS dosaďte do grafu přímo hodnoty 0, 1 nebo X (don't care).
- Signály CLK a RST neuvádíme mezi vstupy automatu ani je nekreslíme do schémat.
- Automat může vhodně kombinovat jak Mealyho, tak Moorovi výstupy.
- Pokud je vstupním signálem vektor bitů, můžete s ním na hranách pracovat jako s vektorem.
- Připomeňte si konvence pro kreslení grafu automatu probírány na přednáškách.
- Nezapomeňte na označení počátečního stavu automatu.

Popis funkce

Stručný slovní popis funkce automatu (max. polovina strany A4).

Snímek obrazovky ze simulací

Zde prosím vložte obrázek (snímek obrazovky) z nástroje GTKWave, který demonstruje funkčnost vašeho obvodu na úrovni simulací. Zachyťte prosím přenos alespoň jednoho datového slova v okně Wave. Pro přehlednost můžete obrázek orientovat např. na šířku stránky A4.