

# Třetí část projektu UART – Testování v hardware

## Analýza a překlad hardware

1. Přípraveny jsou tři možnosti jak realizovat překlad hardwarové architektury UART modulu a jeho zapojení na procesor ARM. Vybrat si je možné jeden z následujících postupů:
  - a. Je možné použít libovolný **počítač v CVT** a na něm nainstalovaný nástroj Vivado v OS Windows. Spuštění nástroje je možné z příkazové řádky (Win+R, cmd, Enter). Po zobrazení okna CMD je potřebné nastavení cesty, v CVT se to provádí příkazem:  
`PATH=Q:\fitkit\Vivado2018\Vivado\2018.2\bin;%PATH%`  
Posléze je možné nástroj již spouštět jednoduše voláním *vivado*. Při každém novém otevření CMD je však potřeba tuto cestu znovu nastavit.
  - b. Přípraven je obraz disku **virtuálního stroje** na [privátních stránkách FITkitu](#), konkrétně jde o archiv *Vivado\_2020.1.7z*. VMDK soubor vybalte pomocí nástroje [7z](#) a použijte pro virtuální stroj v programu [VirtualBox](#). Při tvorbě virtuálního stroje použijte nastavení Ubuntu (64-bit), alespoň 2048 MB paměti, stažený VMDK disk.
  - c. Nástroj Vivado v správné verzi je možné **nainstalovat lokálně** na svůj počítač. Z [privátních stránek FITkitu](#) si stáhněte instalační archiv pro Linux nebo Windows, rozbalte jej a spusťte instalaci. V nastaveních instalace můžete vypnout podporu DocNav, Cable Drivers a všech FPGA čipů kromě Zynq-7000 rodiny. Po dokončení instalace přidejte podsložku *Xilinx/Vivado/2020.1/bin* do globální proměnné PATH a pro fungování programu *make* na Windows také *Xilinx/Vivado/2020.1/gnuwin/bin*.
2. Z informačního systému si stáhněte ZIP archiv zdrojových souborů a rozbalte ho. Podle použité verze nástroje Vivado zvolte složku, se kterou budete dále pracovat. V CVT je dostupná verze 2018.2, jinak (VM nebo vlastní instalace) je použitá verze 2020.1. Seznamte se s obsahem složky, naleznete v ní hlavně tyto části:
  - a. Složku *ip\_cores* obsahující specifické moduly (IP cores). Konkrétně zde jde jenom o modul *simple\_uart\_1.0* implementující kompletní zjednodušenou UART periferii.
  - b. Složku *overlay* obsahující implementaci hlavní architektury zapojení hardwarového obvodu pro FPGA čip včetně jejího ovládání skrze třídu řadiče v jazyce Python.
  - c. Program *uart\_echo.py* implementující jednoduchou aplikaci pro přenos dat skrze UART modul zapojený v hardwarovém overlay.
3. Přesuňte se do složky *ip\_cores* a otevřete *.xpr* soubor Vivado projektu pro editaci modulu *simple\_uart\_1.0*. Otevření je možné z GUI nástroje Vivado skrz možnost *Open Project* nebo voláním následujícího příkazu z CMD: `vivado edit_simple_uart_v1_0.xpr`. Analyzujte vnitřní architekturu modulu pomocí volby *Open Elaborated Design*. Všimněte si zejména zapojení zatím prázdných *UART\_RX* a *UART\_TX* komponent, které jsme právě v projektu nebo na cvičení implementovali.

4. Z informačního systému si stáhněte svůj *xlogin00.zip* archiv s odevzdanou implementací komponenty *UART\_RX* z druhé části projektu. Ve složce *ip\_cores/simple\_uart\_1.0/hdl* nahraďte prázdné soubory *uart\_rx.vhd* a *uart\_rx\_fsm.vhd* právě touto vaší implementací.
5. Podobně jako v předešlém bodě nahraďte také soubory *uart\_tx.vhd* a *uart\_tx\_fsm.vhd* svou vlastní implementací kontroléru vysílací strany, nebo použijte vzorovou implementaci zveřejněnou v informačním systému.
6. V GUI nástroje Vivado proveďte *Reload* otevřeného schématu pro načtení právě změněných souborů. Všimněte si, že komponenty *UART\_RX* a *UART\_TX* již nejsou prázdné. Kliknutím na symbol + v jejich horním rohu můžete prozkoumat architekturu těchto implementací tak jak ji pochopil nástroj. Následně prozkoumejte zvenku viditelné parametry celého IP modulu v dialogu otevřeném volbou *Package IP*. Zastavte se v sekci *Addressing and Memory* kde jsou definovány ze softwaru přístupné registry. Změnu implementace části modulu nakonec potvrďte volbou *Re-Package IP* na konci dialogu.
7. Po zavření nástroje Vivado se přesuňte do složky *overlay*. Zde proběhne překlad FPGA architektury realizující propojení UART IP modulu s ARM procesorem platformy Zynq. Spusťte z CMD připravené překladové TCL skripty nástrojem *make* nebo přímo příkazy:  

```
vivado -mode batch -source build_blockdesign.tcl -notrace
```

```
vivado -mode batch -source build_bitstream.tcl -notrace
```

Zatímco běží překlad, pokračujte následující kapitolou, posléze se sem vraťte.
8. Po skončení překladu se seznamte s architekturou implementovaného napojení UART modulu na čipu Zynq. Projděte si zejména:
  - a. Hardwarový Vivado projekt vytvořený ve složce *overlay/inc\_uart* v průběhu překladu. Otevření podobně jako dříve, pomocí *.xpr* souboru přes GUI volbu *Open Project* nebo voláním příkazu z CMD: `vivado inc_uart.xpr`. Náhled na použité zapojení IP modulů poskytuje volba *Open Block Design*. Náš jednoduchý UART modul je propojený s procesním systémem čipu Zynq pomocí AXI sběrnice a je také připojen na dva externí signály.
  - b. Definování mapování externích signálů na konkrétní piny pouzdra čipu Zynq v souboru *overlay/inc\_uart.xdc*. RXD sériový vodič je připojen na pin W14 a TXD sériový vodič na pin Y14. Podle [dokumentace platformy PYNQ-Z2](#) od výrobce jsou tyto piny čipu Zynq připojeny na piny 1 a 2 portu PMODB.
  - c. Implementaci Python tříd v souboru *overlay/inc\_uart.py* pro ovladač UART periferie a její zapojení v hardwarové architektuře. Třída *UartCtrl* implementuje ovladač a zapouzdřuje specifika nízko úrovněvého ovládání komponenty do lehčeji uchopitelných funkcí. Třída *INC\_UARTOverlay* reprezentuje celkovou architekturu zapojení, v tomto případě zde není definováno nic navíc.
9. Po úspěšném dokončení překladu jsou ve složce *overlay* vytvořeny soubory *inc\_uart.bit* a *inc\_uart.hwh* s konfigurací pro programovatelnou FPGA logiku čipu Zynq.

## Zprovoznění kitu PYNQ-Z2

Kit PYNQ-Z2 včetně základních doplňků si můžete půjčit z knihovny fakulty. Doplňky kitu by měli zahrnovat minimálně micro-SD kartu s nahráním PYNQ-Z2 obrazem OS, micro-USB kabel, a RJ45 síťový kabel. Pro potřeby této demonstrace je navíc potřeba jeden DuPont M-M kablík. Při organizovaném laboratorním cvičení z předmětu INC všechny tyto pomůcky studentům zajistí cvičící.

K platformě PYNQ existuje [oficiální portál](#) s veškerou potřebnou dokumentací. Zde se také nachází sekce speciálně věnovaná [zprovoznění PYNQ-Z2 kitu](#). Postup je popsán formou videa nebo v textu sekcí odkazované stránky. Základní zprovoznění zahrnuje zejména:

1. Základní zprovoznění a zapnutí kitu:  
video 1:54 až 4:01 a 6:04 až 7:01,  
nebo textové sekce *Board Setup* a *Turning On the PYNQ-Z2*.
2. Zapojení USB kabelu pro přístup k příkazové řádce skrz sériovou linku:  
video 9:28 až do konce,  
nebo textové sekce *Opening a USB Serial Terminal*.

*Poznámka:* Na počítači v CVT si stáhněte program *putty* z odkazu v textové sekci. Stahujte přímo *.exe* soubor, který je možné spustit bez instalace.

*Poznámka:* Na Linuxu ve VM spusťte *putty* jako root a zvolte sériový port */dev/ttyUSB1*. Nezapomeňte také při použití VirtualBoxu předat zprávu nad USB zařízením *Xilinx TUL USB* do rukou běžícího virtuálního stroje.

*Poznámka:* Jestli po připojení sériové linky vidíte prázdné okno terminálu, stiskněte Enter.

3. Připojení Ethernet kabelu pro přenos souborů a webový přístup:  
video 4:01 až 6:04 a 7:01 až 9:28,  
nebo textové sekce *Network connection*.

*Poznámka:* Na počítači v CVT je nástroj Vivado spouštěn ze síťového disku Q. Před odpojením síťového kabelu a připojením PYNQ-Z2 dokončete používání nástroje a nachystejte si potřebné výsledky překladu.

Teď byste měli mít kit PYNQ-Z2 připraven k použití. Za zmínku stojí, že kromě demonstrační aplikace připravené v následující kapitole jsou přímo na kitu dostupné i jiné ukázkové aplikace ve formě Jupyter Notebooků. Přístupné jsou skrze síťové připojení ke kitu z webového prohlížeče počítače.

## Spuštění na kitu

1. Vytvořte si složku v souborovém systému na SD kartě kitu PYNQ-Z2. Nakopírujte do této složky následující soubory (podsložku *overlay* v cestě zachovejte):
  - *uart\_echo.py* s implementací demonstrační aplikace,
  - *overlay/inc\_uart.py* s implementací tříd pro přístup k hardwarové architektuře,
  - *overlay/inc\_uart.bit* s přeloženým binárním souborem konfigurace FPGA,
  - *overlay/inc\_uart.hwh* s popisem přeložené architektury a její komponent.
2. Připojte se *putty* terminálem ke kitu PYNQ-Z2 a získajte root práva (*su* s heslem *xilinx*).
3. Spustěte demonstrační aplikaci pro UART komunikaci voláním *python3 uart\_echo.py*. Aplikace nejdřív nahraje konfiguraci do FPGA a inicializuje přenosy. Pak je možné zasílat a přijímat zprávy přes sériové datové vodiče UART sběrnice. Co napíšete do konzole a potvrdíte klávesou Enter se odešle na TXD vodič. Co je přijato na RXD vodiči se automaticky čte a vypisuje do konzole. Kombinace *Ctrl+C* běh aplikace ukončuje.
4. Propojte DuPont kablíkem piny 1 a 2 PMODB portu, na které jsou z FPGA vyvedeny RXD a TXD vodiče UART sběrnice. Port PMODB je na desce pojmenován, nachází se vedle čtyř tlačítek. Piny 1 a 2 jsou první dva zprava v horním řádku portu. Propojením RXD a TXD vzniklo takzvané *loopback* zapojení, všechno co vysílací strana odesílá na RXD pin se zapojeným vodičem vrací na přijímací stranu TXD pinem. Cokoliv je teď v demonstrační aplikaci odesláno by se mělo vzápětí přijmout a vypsat, každý řádek tedy vidíte v konzoli dvakrát.
5. Rozpojte *loopback* propoj své UART sběrnice a propojte se se sousedem. Pozor, vždy propájejte piny do kříže. Tedy svůj pin 1 (RXD) napojte na pin 2 (TXD) druhého kitu a svůj pin 2 napojte na pin 1 druhého kitu. Teď by mělo být možné posílat si zprávy přes UART mezi kity. Při spuštění demonstrační aplikaci na obou kitech je v konzoli jednoho vypsáno cokoli co bylo druhým odesláno a opačně.