



## Laboratorní cvičení z předmětu CITE

Posuvný registr / sériová komunikace, Johnsonův čítač, LFSR



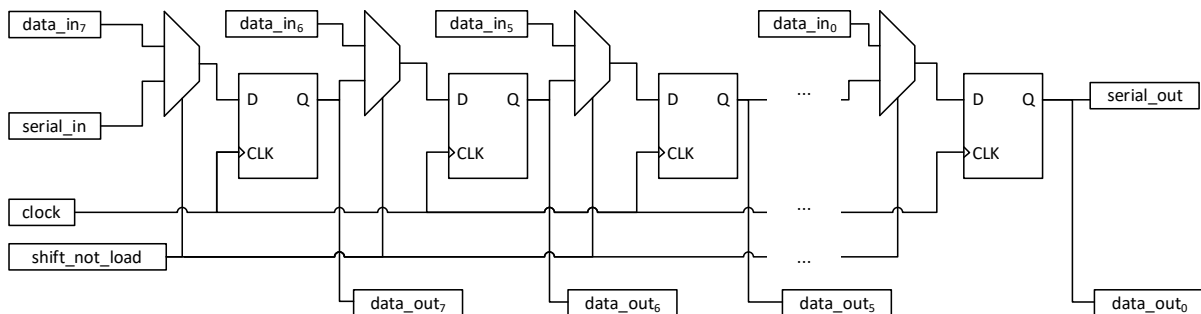
1. Z kurzu předmětu na [elearning.tul.cz](http://elearning.tul.cz) stáhněte a rozbalte projekt LAB07b.zip.

*Cesta k projektu nesmí obsahovat diakritiku, mezery nebo speciální znaky. Vhodné umístění je v laboratoři A107 disk D:, ve kterém si můžete vytvořit podadresář.*

2. Dvakrát klikněte na soubor LAB07b.xpr uvnitř dekomprimované složky. Otevře se prostředí Vivado.

## Posuvný registr

1. Otevřete obvod **shift\_reg.vhd**
2. Vytvořte popis 8bit posuvného registru s paralelní předvolbou a sériovým vstupem. Pro popis využijte příkaz `process` a postupujte obdobně jako v předcházejícím cvičení:

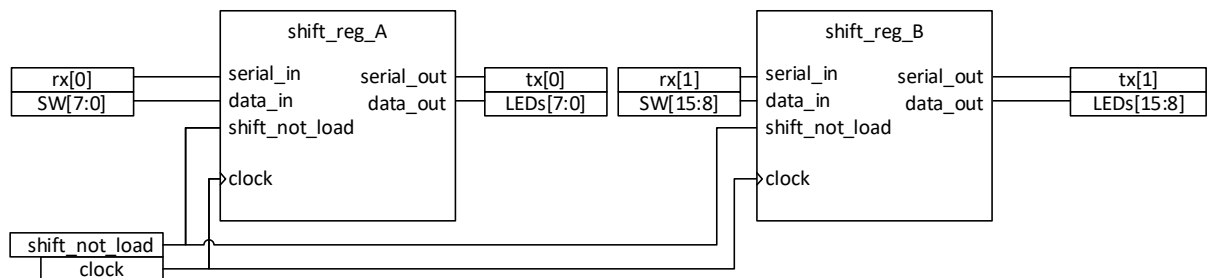


- a. Je-li vstup **shift\_not\_load** v log.0, pak se do registrů nahrají **data\_in**. Je-li v log.1, pak se do registru [7] nahraje vstup **serial\_in**; hodnota registru [7] se nahraje do registru [6], registr [6] → registr [5], atd. až po operaci registr [1] → registr [0]
    - i. Funkci lze popsat pomocí jednoho výrazu s použitím výběru (**range**) a slučování **&** na pravé straně.
  - b. Výstup **serial\_out** je připojen na registr [0]. Nedávejte ho do procesu – vytvořili byste další registr.
  - c. Výstupy obvodu (před multiplexory) jsou zapojeny na výstup **data\_out**.
3. Otevřete simulační soubor **shift\_reg\_tb.vhd**.
  4. Upravte simulaci tak, abyste ověřili chování obvodu **shift\_reg**:
    - a. Nahrajte nějakou hodnotu do registru
    - b. Vysouvejte ji a měňte hodnotu vstupu **serial\_in** tak, aby se na výstupu **data\_out** objevila žádaná hodnota 0xC9.

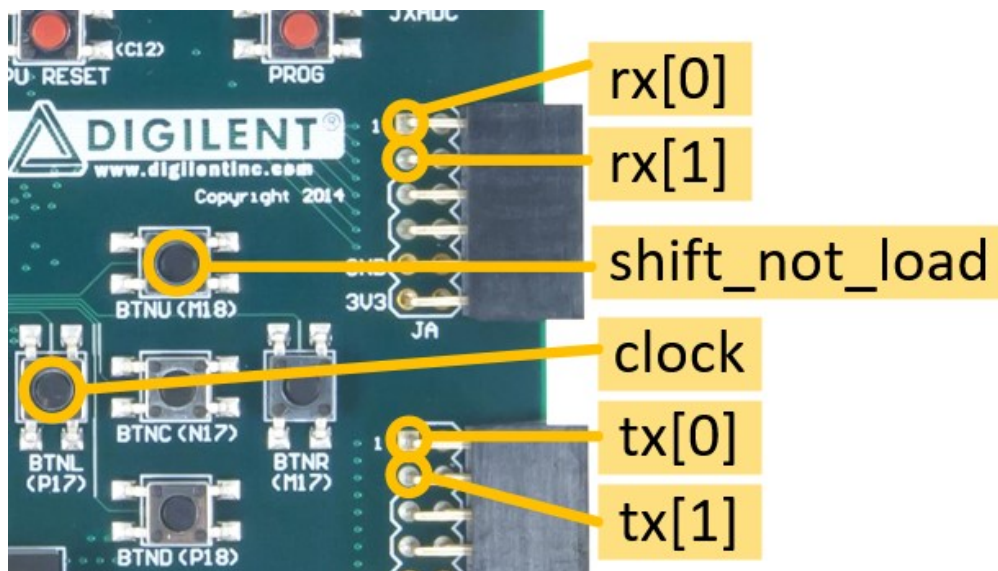


## Sériová komunikace

1. Otevřete soubor **rxtx.vhd** a nastavte jej jako hlavní obvod (set as top)
2. Vytvořte dvě **instance** obvodu **shift\_reg** a propojte je se vstupy a výstupy obvodu dle schéma:



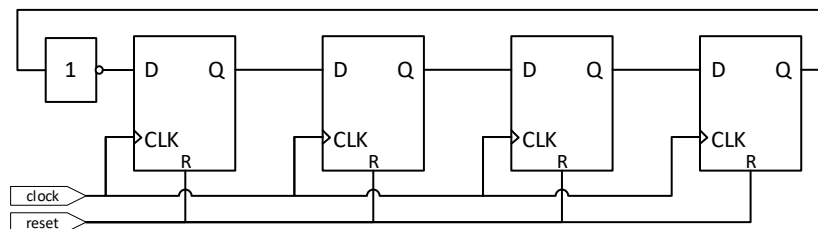
3. Vytvořte bitstream a nahrajte jej do vývojové desky.
4. Pomocí přiložených vodičů propojte signály **rx** (receive) a **tx** (transmit) tak, abyste mohli vzájemně přenášet data mezi posuvnými registry. **LED** a **SW** jsou na obvyklém místě:



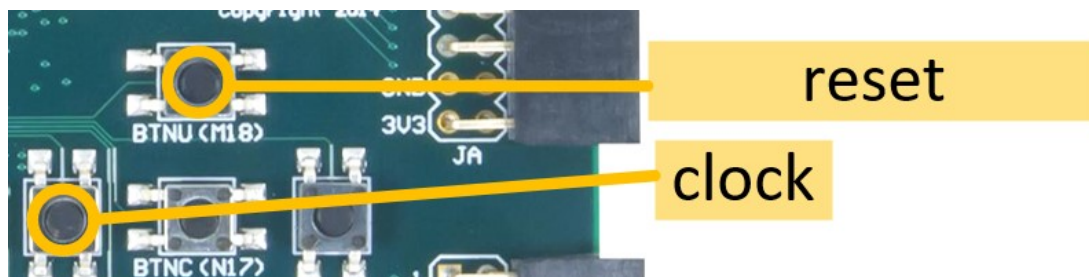
PS: Chcete-li komunikovat se sousedy, budete muset propojit i zem (GND). Hodinový signál poslat mezi deskami není triviální, takže se spokojíme s ruční synchronizací.



## Johnsonův čítač

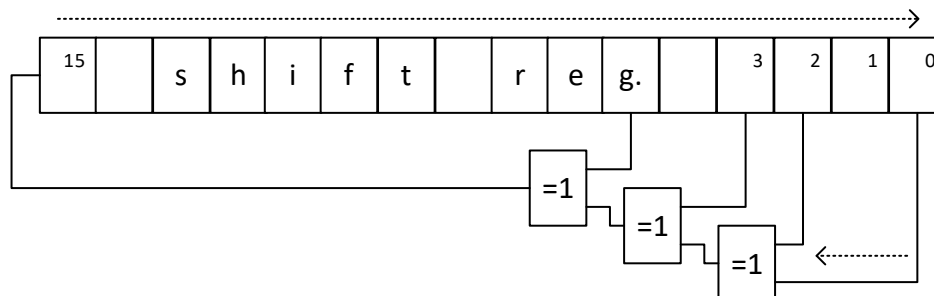


1. Otevřete soubor **johnson\_count.vhd** a nastavte jej jako hlavní soubor (set as top).
2. Navrhněte čtyřbitový Johnsonův čítač
  - a. Synchronní vstup **reset** vynuluje registry. Vstup je aktivní v log. 1.
  - b. Na výstup **q\_johnson** přiveďte výstup z registrů čítače.
  - c. Zaznamenejte si všechny stavy Johnsonova čítače.
    - i. Začněte hodnotou 0000 a pokračujte, dokud nedojde k opakování. Každý stav označte vzestupně v binárním váhovém kódu.
    - ii. Ze záznamu vytvořte dekodér, který bude kódovat stav čítače do binárního váhového kódu
  - d. Upravte délku výstupu **q\_binary** na správnou hodnotu a připojte na něj výstup dekodéru.
3. Vytvořte bistream a funkci čítače a dekodéru ověřte na vývojové desce. Johnsonův kód je zapojen na diodách vlevo, binární váhový kód na LED vpravo. Ovládání je následující:

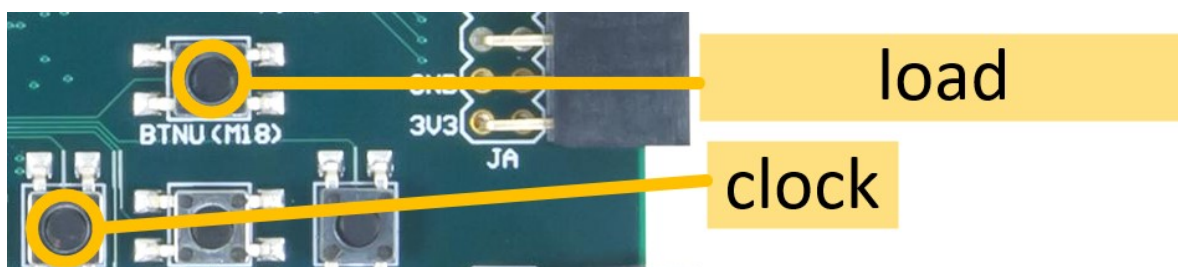




## Posuvný registr s lineární zpětnou vazbou (LFSR)



1. Přepněte na soubor **LFSR.vhd** a nastavte jej jako hlavní soubor (set as top)
2. Navrhněte obvod 16 bit lineárního zpětnovazebného posuvného registru.
  - a. Jádrem LFSR tvoří posuvný registr. Namísto externího sériového vstupu však vytvoříme zpětnou vazbu tak, jak zobrazuje zjednodušené schéma výše.
  - b. Vstup **load** (není na schématu zobrazen) řídí chod obvodu následovně: Je-li aktivní, pak se do registru nahraje hodnota na vstupu **SW**. Není-li aktivní (v log. 0), pak se registr posouvá a hodnota 15. registru je nahrazena hodnotou ze zpětné vazby.
  - c. Stav registru je zobrazován na **LED**.
3. Vytvořte bitstream a nahrajte jej do vývojové desky. Ovládání LFSR:



4. Jak se obvod chová? Co se stane, když nahrajeme do registru samé 0?