



Laboratorní cvičení z předmětu PO

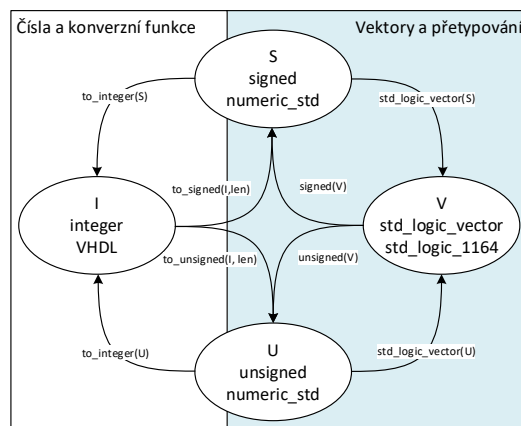
Čítače, děličky hodin, buzení displejů



1. Z kurzu předmětu na elearning.tul.cz stáhněte a rozbalte projekt **LAB03.zip**.
2. Dvakrát klikněte na soubor **LAB03.xpr** uvnitř dekomprimované složky. Otevře se prostředí Vivado.

Knihovna **numeric_std** a čítač

1. Knihovna **numeric_std** definuje datové typy **unsigned** a **signed** a umožňuje s nimi realizovat aritmetické operace. V dnešním cvičení si vyzkoušíme sčítání. Konverze a přetypování mezi typy **(un)signed**, **std_logic_vector** a **integer** definuje následující diagram:

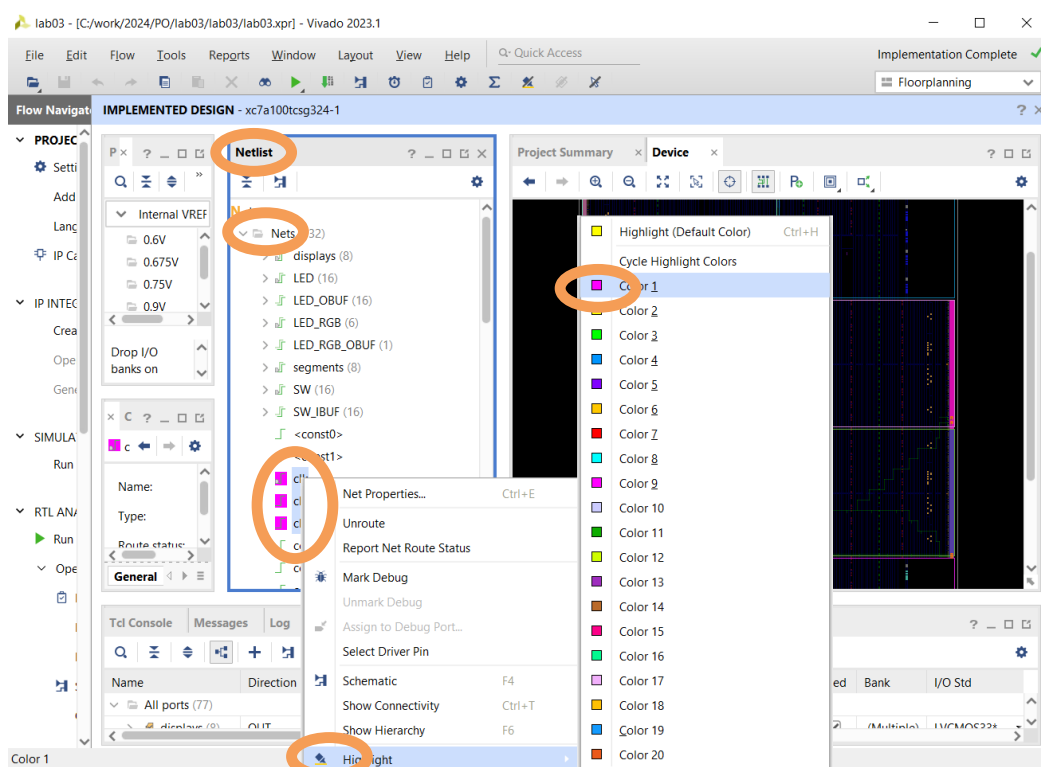


2. Otevřete soubor **top.vhd** a vytvořte v něm synchronní cyklický čítač s předvolbou
 - V deklarační části architektury deklarujte proměnnou **counter** datového typu **unsigned** s šířkou 16 bitů
 - Vytvořte proces, ve kterém implementujete čítač:
 - Čítá se na náběžnou hranu hodinového signálu **clk**
 - Je-li aktivní signál **rstn** (=0), pak se nastaví na hodnotu 0
 - Čítač čítá od 0 do hodnoty **SW**
 - Čítání se po dosažení SW cyklicky opakuje
 - V těle architektury pomocí paralelních příkazů:
 - Po dosažení hodnoty SW rozsviďte **LED_RGB(0)**, jinak ji ponechte vypnutou
 - Hodnotu čítače zobrazte na **LED**
3. Vytvořte **bitstream** a ověřte na vývojové desce



Děličky hodin

1. Otevřete implementovaný design z předchozí úlohy (**Flow** → **Open Implemented Design**)
2. Přepněte na floorplanning (**Layout** → **Floorplanning**)
3. V podokně **Netlist** rozbalte složku **Nets** a zvolte vodiče **clk**, **clk_IBUF** a **clk_IBUF_BUFG**, klikněte **pravým** tlačítkem myši a zvolte **Highlight** a vyberte nějakou barvu:



4. Prozkoumejte:
 - Jak jsou vedeny hodinové signály v FPGA.
 - Všimněte si dedikovaných vodičů vedoucích ze „středu“ obvodu.
 - Vstup CE registru FDRE, pomocí kterého je strukturní omezení FPGA pro generování hodin uvnitř obvodu řešeno.
5. Upravte proces čítače tak, abyste vytvořili kaskádu dvou čítačů – děličkou a čítačem.
 - Dělička periodicky čítá a každých 10 ms „povolí“ jedno přičtení hlavnímu (původnímu) čítači a sama se nastaví opět do 0.
 - Frekvence hodinového signálu **clk** je 10 ns. Do jaké hodnoty musí dělička čítat?

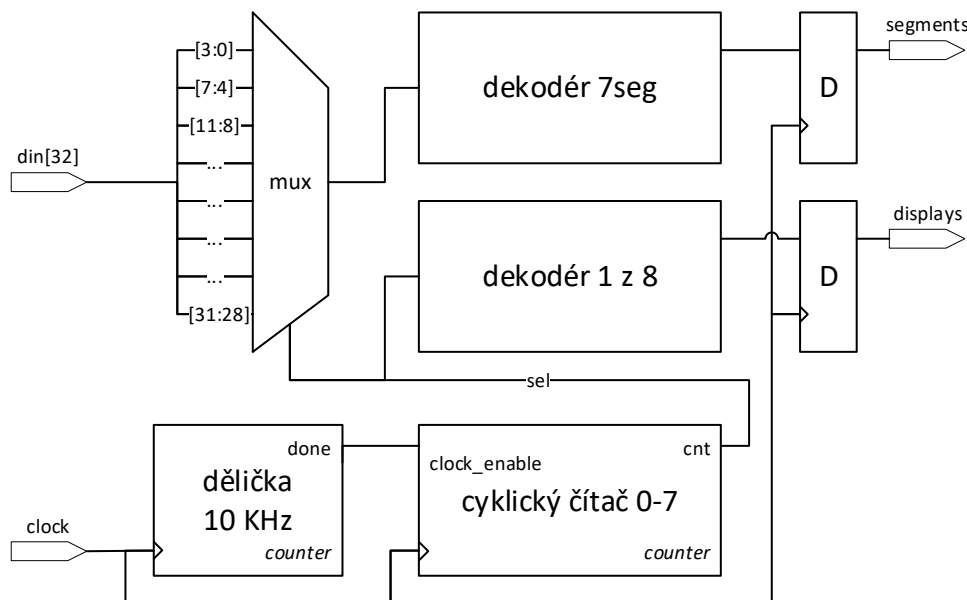


- Řešte s pomocí proměnné typu integer s omezením rozsahu:

```
variable div : integer range low to high := 0;
```

Budič 8x7seg displeje

1. Otevřete soubor **ssegdriver.vhd**
2. S pomocí znalosti čítačů a dekodérů z předešlých cvičení sestavte budič sedmisegmentových displejů.



3. Doporučený postup:
 - a. **Čítač a děličku** vytvořte pomocí jednoho procesu. **Děličku** vytvořte pomocí pomocné proměnné v procesu, **sel** vytvořte jako signál typu **unsigned**.
 - b. Pro oba dekodéry vytvořte procesy implementující registr i dekodér současně.
 - c. Multiplexor vstupních dat realizujte pomocí přiřazovacího příkazu. Rozsah vybírejte s pomocí konverze **sel** na **integer**.
4. Otevřete soubor **top.vhd**:
 - a. Upravte kód předchozích úloh tak, aby čítač byl 32bitový, ale stále čítal s periodou 1ms.
 - b. LED zapojte pouze na horních 16 bitů výstupu čítače



- c. Horní limit čítání posuňte na horních 16 bitů a zprava doplňte 0.
 - d. Vložte instanci budiče sedmisegmentových displejů a připojte jej na výstup upraveného čítače.
5. Vygenerujte bitstream a ověřte funkci obvodu

Domácí úloha

V souboru **debounce.vhd** popište obvod, který detekuje stisk tlačítka (vstup **din**) a signalizuje jej na výstupu **pressed** po dobu jednoho taktu hodinového signálu **clk**. Údaje o frekvenci jsou do předávány generickým parametrem **C_CLK_FREQ_HZ**. Generický parametr **C_REQ_SCAN_RATE_HZ** udává žádanou frekvenci zkoumání vstupu.

Funkci obvodu otestujte pomocí souboru **debounce_tb.vhd**, který budete muset doplnit o testovací buzení vstupu **din**. Simulaci spustíte ve Vivado pomocí **Flow** → **Simulation** → **Run Behavioral Simulation**