



## Laboratorní cvičení z předmětu HSC

### LAB 06 – Model Composer



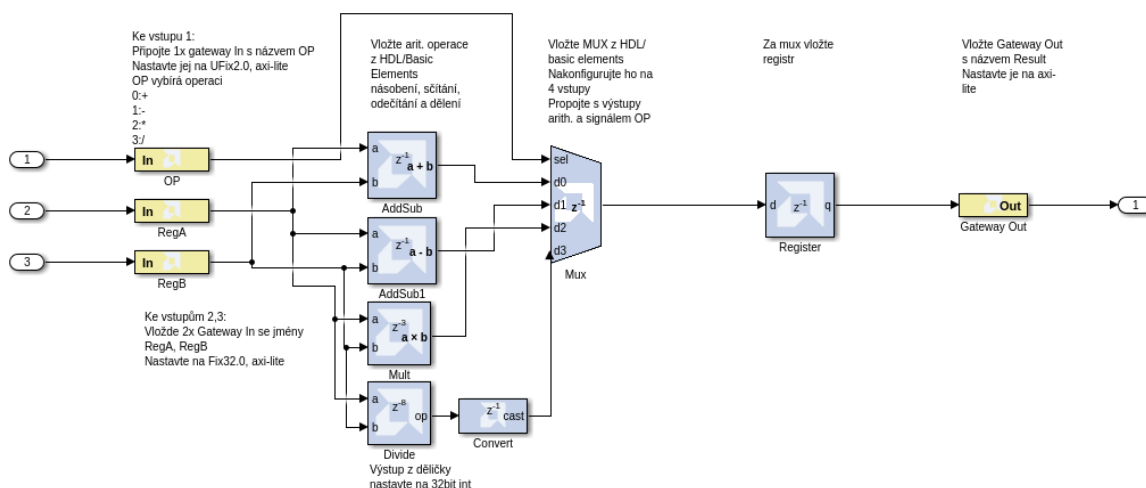
## Sestavení základního systému

Nejprve se seznámíme se základním systémem. Budeme pracovat s deskou **Avnet Zedboard**. Systém obsahuje instanci procesorového systému Zynq PS, jádro **axi\_video** a potřebnou sběrnicovou infrastrukturu. Připravené IP jádro **axi\_video** z operační paměti čte obraz a zobrazuje ho na analogovém video výstupu. Činí tak v rozlišení 10240×480 @60 Hz. Modul zobrazování běží asynchronně vůči systému na 25 MHz. V **Model Composer** vytvoříme jednoduché IP jádro, které bude výstupní digitální signál realtime upravovat.

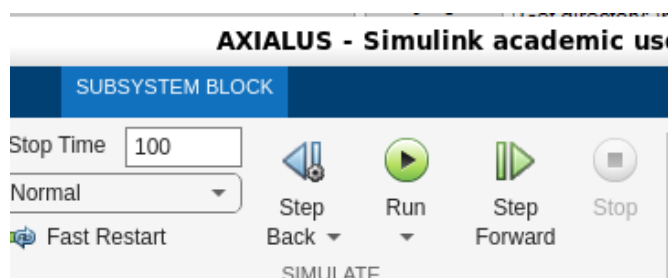
- 1) Stáhněte **lab06.zip** a rozbalte jej, nebo si do domovského adresáře zkopírujte adresář **/srv/HSC/lab06**
- 2) Nastavte prostředí vašeho bash na Matlab 2022a a Vivado 2024.1 pomocí příkazu
- 3) `source /srv/HSC/env/xilinx.sh`
  - a) V případě, že máte modifikovaný `.bashrc`, prosím odstraňte z něj řádky s `LM_LICENSE_FILE`, cestou k matlabu a nástrojům Xilinx.
- 4) Spusťte **Vivado** a otevřete projekt ve složce `lab06_vivado`.
- 5) Otevřete block design **Flow** → **Open Block Design** → **design\_1.bd** a seznámte se se strukturou návrhu

## Seznámení s Model Composer

- 1) Spusťte **Model Composer** (`model_composer&`)
- 2) Změňte složku na **lab06/lab06\_mc**
- 3) Otevřete připravený **Simulink** projekt **AXIALU.slx**
- 4) Otevřete **subsystem AXIALU** a doplňte dle instrukcí obvod, který bude realizovat primitivní ALU jednotku na sběrnici AXI-Lite



- 5) Vyzkoušejte funkci jednotky v připravené simulaci pomocí tlačítka **Run**

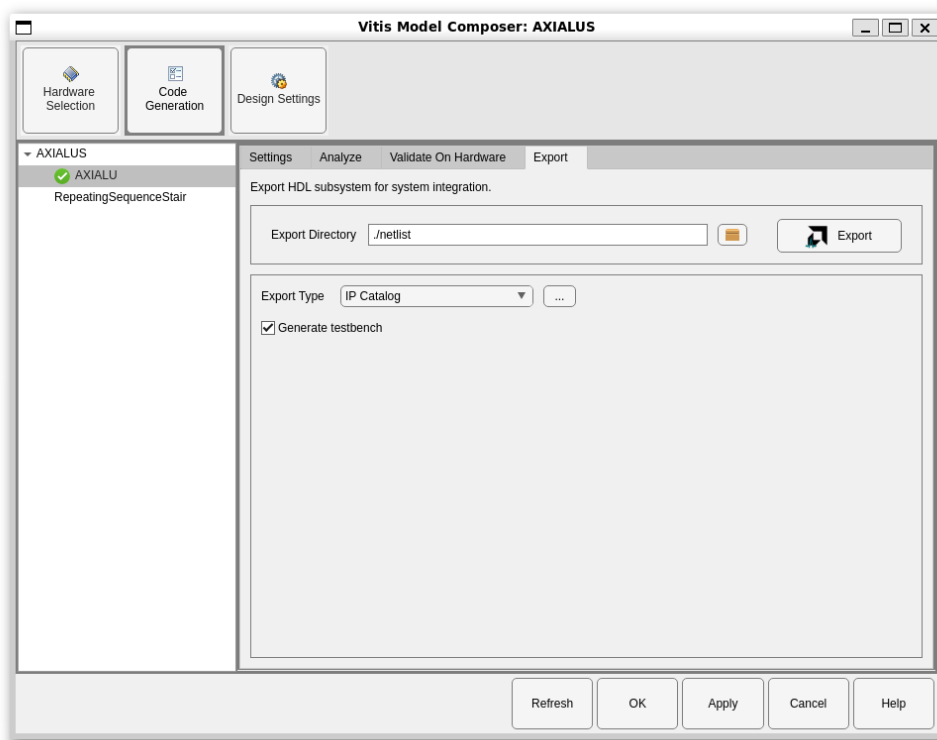


- 6) Pro export IP jádra otevřete bloček **Vitis Model Composer Hub**

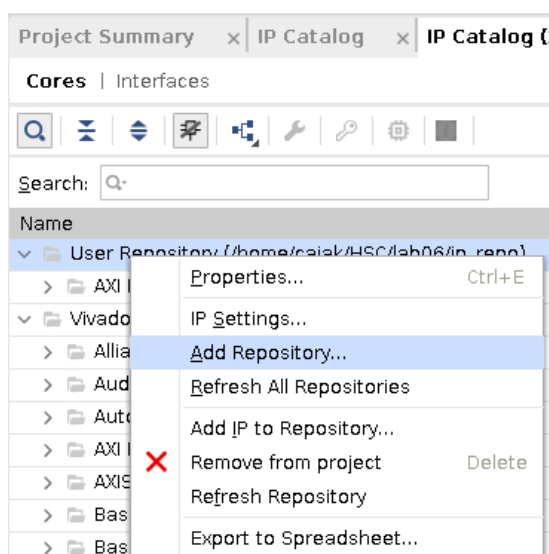


Vitis Model Composer Hub

- 7) V záložce **Export** obrazovky **Code Generation** klikněte na **Export**



- 8) Vraťte se do **Vivado** a přidejte cestu k IP repozitáři (**Window**→**IP Catalog**), pravý tlačítkem myši v okně **IP Catalog** zvolte **Add Repository**:





- a) Přidejte repozitář **/lab06/lab06\_mc/netlist**
- 9) Vložte IP jádro do obvodu a propojte se zbytkem systému.

### Model Composer realtime video jádro

- 1) Otevřete soubor **videortp.slx** a jeho subsystém **video\_rtp**
- 2) Vytvořte HW, který umožní:
  - a) **Konverzi** z barevného prostoru RGB do šedotónového prostoru, dle vzorce

$$Y = 0.3R + 0.59G + 0.11B$$

- b) **Derivaci** šedotónového obrazu v ose X šedotónového, tj. diferenci mezi dvojicí sousedních pixelů
  - c) **Prahování** s volitelným prahem
  - d) Prostý **průchod** obrazu
  - e) Volitelně můžete vytvořit funkci gamma korekce, derivace v ose Y, obarvení šedotónového obrazu nebo jinou jednoduchou operaci.
- 3) Poznámky k vypracování:
  - a) Pro ovládání/nastavení využijte bloček **gateway\_in** s názvem **op\_select** a jeho přejmenované (a přenastavené) kopie. Je nastaven tak, aby byl přístupný přes rozhraní AXI-lite.
  - b) Budete-li využívat aritmetické funkce, pamatujte na to, že počet registrů v datové cestě se musí shodovat s počtem registrů umístěných mezi vstupy **HSync**, **VSync** a výstupy **f\_HSync** a **f\_VSync**. S výhodou lze využít bloček **delay**.
  - c) Signál **active** indikuje aktivní oblast obrazu. Mimo aktivní oblast generujte na výstup vždy 0,0,0.
- 4) Po dokončení model vyexportujte jako IP jádro a vložte ho do blokového schéma ve **Vivado**.
  - a) Propojte jej přímo s výstupy jednotky **axi\_video**.
  - b) Odstranění dolních bitů RGB signálu ponechejte na výstupu vašeho IP jádra.
  - c) Do hodinového vstupu připojte signál **FCLK\_CLK1** z výstupu bloku Zynq PS (65 MHz).
- 5) Vytvořte HDL wrapper **bd** souboru (**Sources**, pravé tlačítko nad **.bd** souborem, **Create HDL Wrapper**), vygenerujte blokový design (**Flow Navigator**→**IP INTEGRATOR**→**Generate Block Design**), vygenerujte bitstream (**Flow**→**Generate bitstream**) a exportujete projekt společně s vytvořeným bitstreamem (**File**→**Export**→**Export HW**).
- 6) Vyzkoušejte funkci obvodu