

# VHDL – projekt

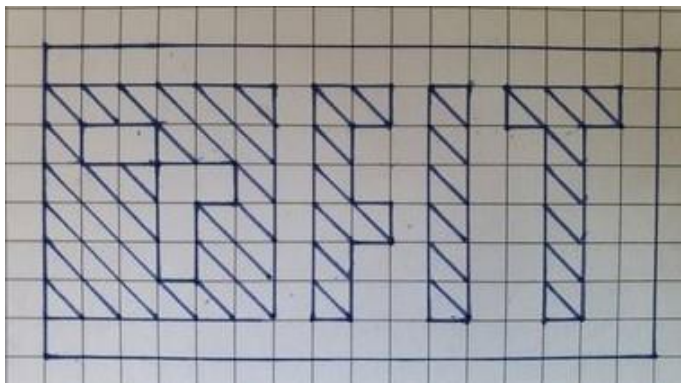
**Jméno:** Michal Zapletal

**Login:** xzaple41

**Datum:** 7.5.2022

## POPIS:

- FSM
  - Řídí celý obvod pomocí 6 stavů
    1. RESET – vyčkává na zapnutí
    2. INITIALIZE – načte ROM do entit sloupců
    3. RIGHT\_DIR – 3x obrázek doprava
    4. LEFT\_DIR – 3x obrázek doleva
    5. ANIMATION – načítá animaci
    6. RELOADING – čeká 2 sekundy než se vrátí do stavu RESET
- COUNTRY
  - 1 „DISPLAY RELOAD“
    1. Nastavuje frekvenci obnovení displeje
    2. Aktivní ve všech stavech
  - 2 „MOVE COLUMNS“
    1. Časová perioda, po které se se posouvá displej
    2. Aktivní ve stavu 2 a 3
  - 3 „CHESSBOARD“
    1. Časová perioda, která je potřeba pro načtení šachovnice
    2. Aktivní ve stavu 5
  - 4 „TIMER“
    1. Odpovídá zhruba 2 vteřinám
    2. Časová perioda, po kterou je zobrazena šachovnice
    3. Aktivní ve stavu 5
  - 5 „MOVE COLUMN COUNTER“
    1. Počítá, kolik proběhlo posuvů
    2. Aktivní ve stavu 6
- ROM
  - Uložení orbázku
  - Výsledný obrázek:



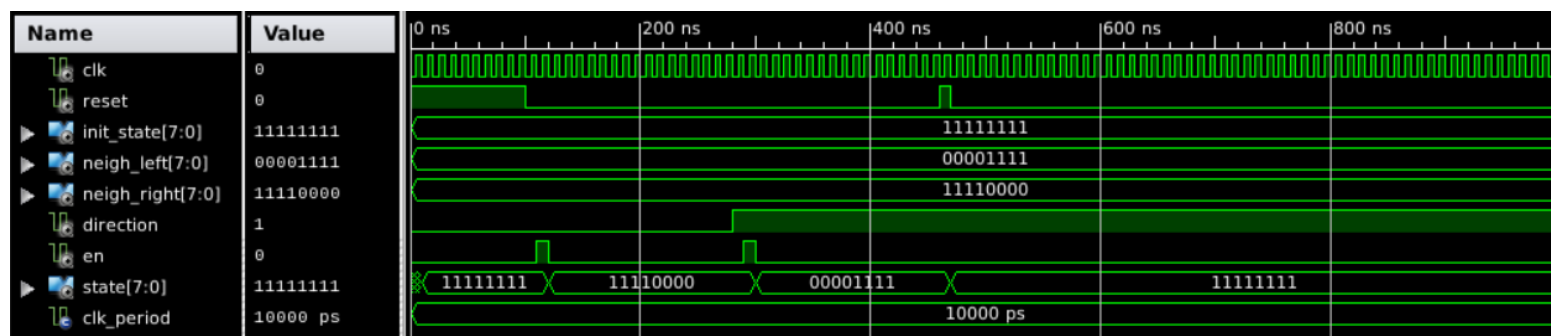
#### - SLOUPEC

- Při signálu RESET ukládá hodnotu STATE\_INIT, tu potom posílá na výstup
- Pokud bude aktivováno EN, tak ukládá hodnotu z pravého nebo levého souseda, tu potom posílá na výstup STATE

#### TESTBENCH:

#### - SLOUPEC

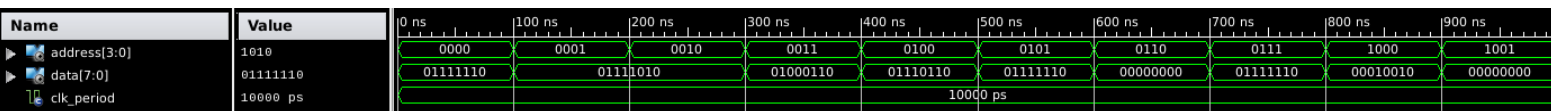
- RESET – při signálu v 1 načítá hodnotu ze STATE\_INIT
- INIT\_STATE, NEIGH\_LEFT, NEIGH\_RIGHT předem inicializované hodnoty
- DIRECTION – nepodařilo se mi implantovat DIRECTION\_T, proto je použita 0 = RIGHT a 1 = LEFT
- EN – nahrává hodnotu do STATE podle DIRECTION z NEIGH\_RIGHT nebo NEIGH\_LEFT
- STATE – výstup



- Na simulaci je vidět použití EN při různých DIRECTION a znovupoužití RESET.
- Sám SLOUPEC ale nemění okolní sloupce, pouze zná jejich hodnoty
- Samotná rotace probíhá na úrovni sloupců jen z půlky. Při spuštění EN se načte na výstup hodnota ze souseda, ale nedokážou následně samy provést rotaci a tak rotace probíhá mimo SLOUPEC.
- Jednodušeji to znamená, že sloupce se samy o sobě dokáží pohybovat bez vnějšího zásahu o jedno doprava nebo doleva, ale ne o víc.

#### - ROM

- DATA – výstup, vektor
- ADDRESS – vstup, adresa vektoru



- Nesynchronní podle CLK, ihned po zadání adresy se objeví příslušná data spadající pod adresu

#### - FSM a Celé FPGA

- Nedokázal jsem spustit simulaci, nenačítal se časový průběh, ať jsem se sebevíc snažil
- Možná byl nějaký problém s generickým parametry. Nešly mi testbenche pouze pokud v nich byl nějaký generický parametr

- **Popis běhu**

- Čeká se na přiřazení do RESET = 0
- Inicializuje se displej z paměti ROM
- Proces pohybu probíhá tak, že entita SLOUPEC při povolení pohybu (EN = 1), tak na výstup načte souseda z příslušné strany. Tohle dělá v cyklu samotný sloupec. K tomu však je ještě potřeba aby vždycky potom byl posunut samotný vstupní vektor do SLOUPEC, takže probíhá ještě posun NE V ENTITĚ SLOUPEC. Poté zase na výstup načte sloupec souseda atd. Takhle se to opakuje v cyklu
- Otáčení na druhou stranu
- Načtení animace (šachovnice)
- Čekání zhruba 2 sekundy kdy je zobrazena šachovnice
- Přechod zpět na začátek

- **Zvolení hodnot proměnných**

- Obnovovací frekvence (COUNTER 1)
  1. Hodnota vycházela ze z CLK frekvence 25MHz, počtu sloupců a následnou frekvenci, kterou jsem si zvolil
  2.  $POČET\ CLK\ CYKLŮ = (CLK\ FREKVENCE / POČET\ SLOUPCŮ)$   
 $POČET\ CLK\ CYKLŮ = 1\ 562\ 500$
  3. To je počet CLK cyklů, které jsou potřeba, aby se displej obnovil za jednu sekundu, to však nestačí, proto to musí vydělit ještě námi zvolenou frekvencí.
  4. 200 už bylo podle mě příliš, tak jsem zvolil 200Hz
  5.  $VÝSLEDEK = 1\ 562\ 500 / 200$   
 $VÝSLEDEK = 15\ 625\ CLK\ cyklů\ (1\ 1110\ 1000\ 0100)$
  6. To odpovídá zhruba 13 bitovém CLK counteru
  7. (chtěl jsem zvolit frekvenci 100Hz, ale na kameře to blikalo)
- Frekvence posunu
  1. Hodnota vychází z CLK a námi zvolené frekvence. Používáme k tomu counter do n
  2. Např. pro counter na 25 bitů je výsledná frekvence 0,74Hz
  3. Osobně jsem zvolil counter na 23 bitů
  4.  $VÝSLEDEK = 25\ 000\ 000 / 2^{23}$   
 $VÝSLEDEK = 2,98Hz$

- **VIDEO**

- <https://youtu.be/Tb3PS72Zin0>