

IVH Project - Grafický čítač na FPGA a ZYNQu

Autor: Nurdaulet Turar (xturarn00)

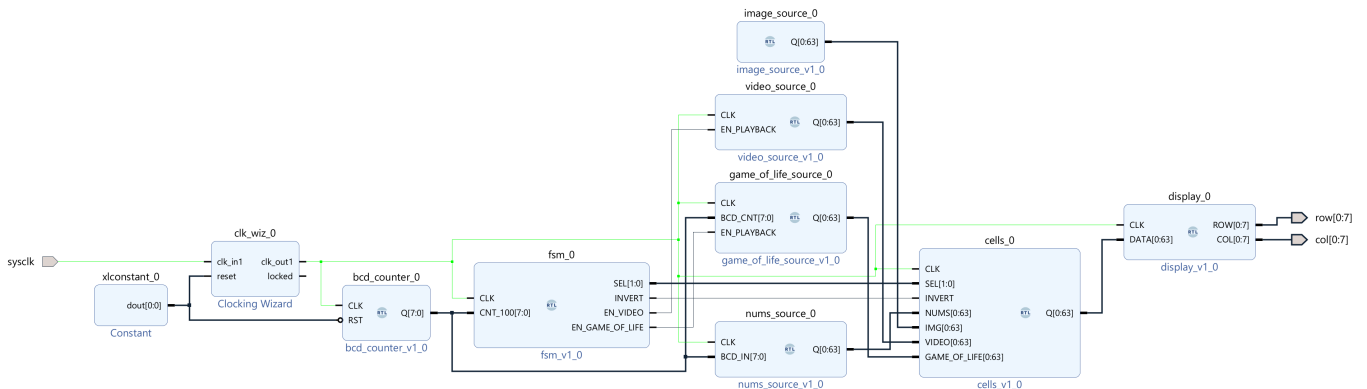
Datum: 10.5.2024

Obsah

- Blokový návrh
 - Zobrazení
 - Zdroje obrázků
 - Běh programu
- Simulace komponent
- Video ukázka na ZYNQu
- Poznámky

Blokový návrh

Do ZYNQ bude připojen maticový LED displej 8x8. Projekt provádí ovládání tohoto displeje. Vstupy displeje na blokovém návrhu (obrázek 1) uvedené jako **row[0:7]** a **col[0:7]**.



Obrázek 1: Blokový návrh

Zobrazení

Maticový displej umožňuje rozsvícení jenom 1 sloupce současně. Kvůli tomu, zobrazení funguje tak, že každý sloupec se zobrazí o 2ms, přepne se na další o 2ms a tak dále. Naše oči blikání na takové frekvenci neumí rozpoznat, což využijeme pro naše účely. Řízení displeje a tento

algoritmus je implementován v bloku **display**. Na vstupu má 64 bity, které reprezentují, jaké LED-ky je třeba rozsvítit.

Zdroje obrázků

Implementace obsahuje 4 "source" bloky s 64bitovými výstupy, které jsou připojené do maticového displeje:

- **nums_source**: Počítá sekundy od 0 do 99.
- **image_source**: Na výstupu je konstantní hodnota - obrázek.
- **video_source**: Při spuštění začne postupně cyklicky nahrávat na výstup obrázky z paměti.
- **game_of_life_source**: Při spuštění bude simulovat [Conway's game of life](#) v reálném čase s určenou v kódu počáteční mřížkou.

Jejich výstupy jsou připojeny do bloku **cells**, kde se provádí výběr obrázku na zobrazení a případně inverze obrázku.

Běh programu

Výběr obrázku je řízen blokem **fsm**:

- Prvních 30 sekund běhu programu se zobrazuje čítač (nums_source).
- Od 21 do 30 sekundy je obrázek invertován.
- Dalších 5 sekund (31-35s) se zobrazuje obrázek (image_source).
- Dalších 10s: video_source.
- Od 45 do 99s: game_of_life_source.
- Po překročení 99s se čítač zresetuje na 0 a bude opakovat stavy uvedené výše, pokud se HW nevypne.

Simulace komponent

bcd_counter

Je sestaven ze 3 generických (generic) za sebou připojených čítačů:

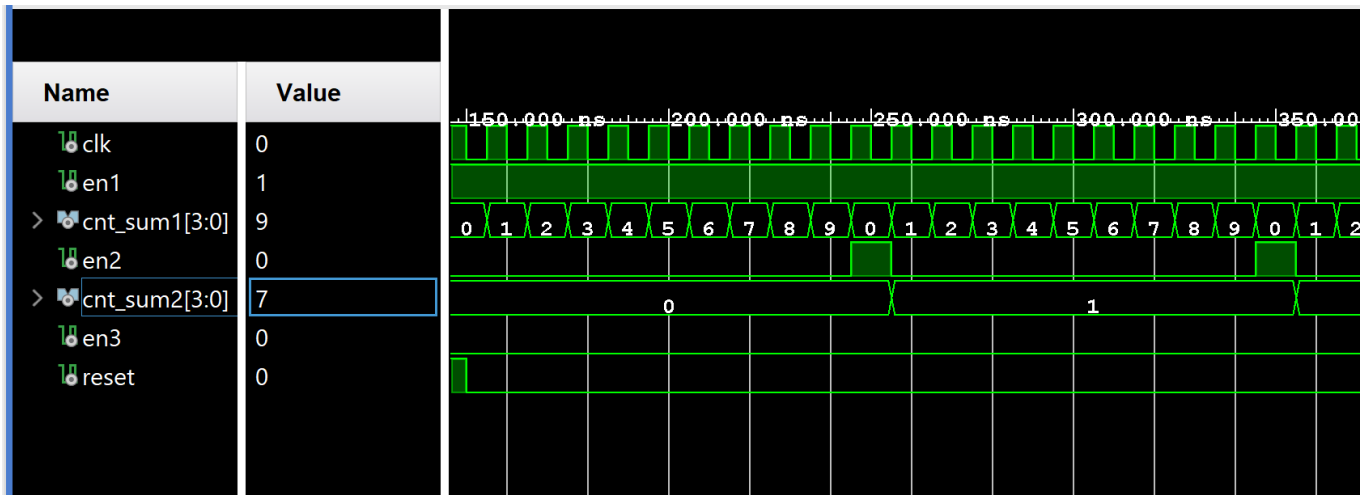
1. Počítá náběžné hrany -> výstup EN=1 jednou za sekundu.
2. Počítá sekundy
 - výstup Q je počet napočtených sekund (0 až 9)

- výstup EN_OUT nastaví při přetečení (-> je třeba navýšit počet desetín)

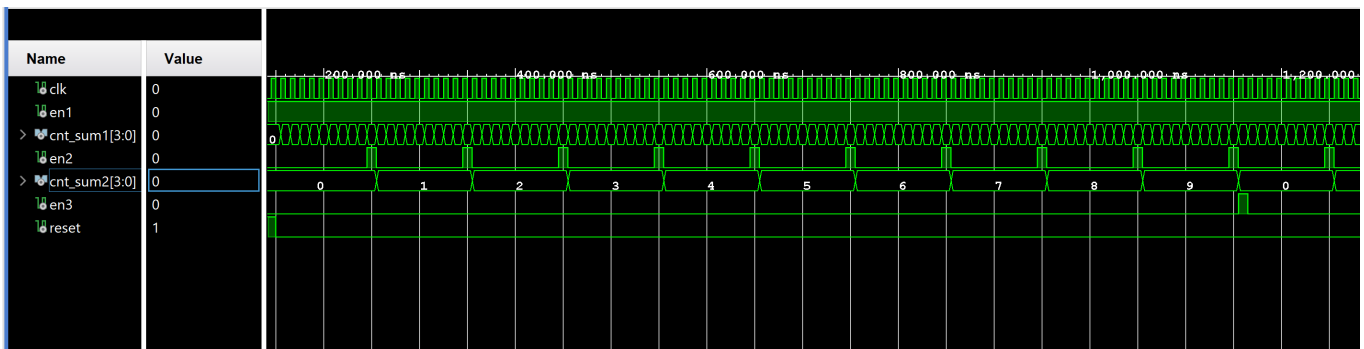
3. Počítá desetiny sekundy.

Výstupem jsou 2 BCD hodnoty - reprezentující 0 až 99 sekund.

Na obrázcích 2 a 3 je uvedena simulace **counter_tb.vhd**, kde jsou definovány 2 za sebou připojené čítače, které počítají od 0 do 9. Výstupy jsou **cnt_sum1[3:0]** (sekundy) a **cnt_sum2[3:0]** (10 * sekundy). Když BCD čítání dosáhne 99s, nastaví se na 0s a pokračuje v počítání, což je uvedeno na obrázku 3.



Obrázek 2: Ukázka sčítání



Obrázek 3: Ukázka sčítání při přetečení

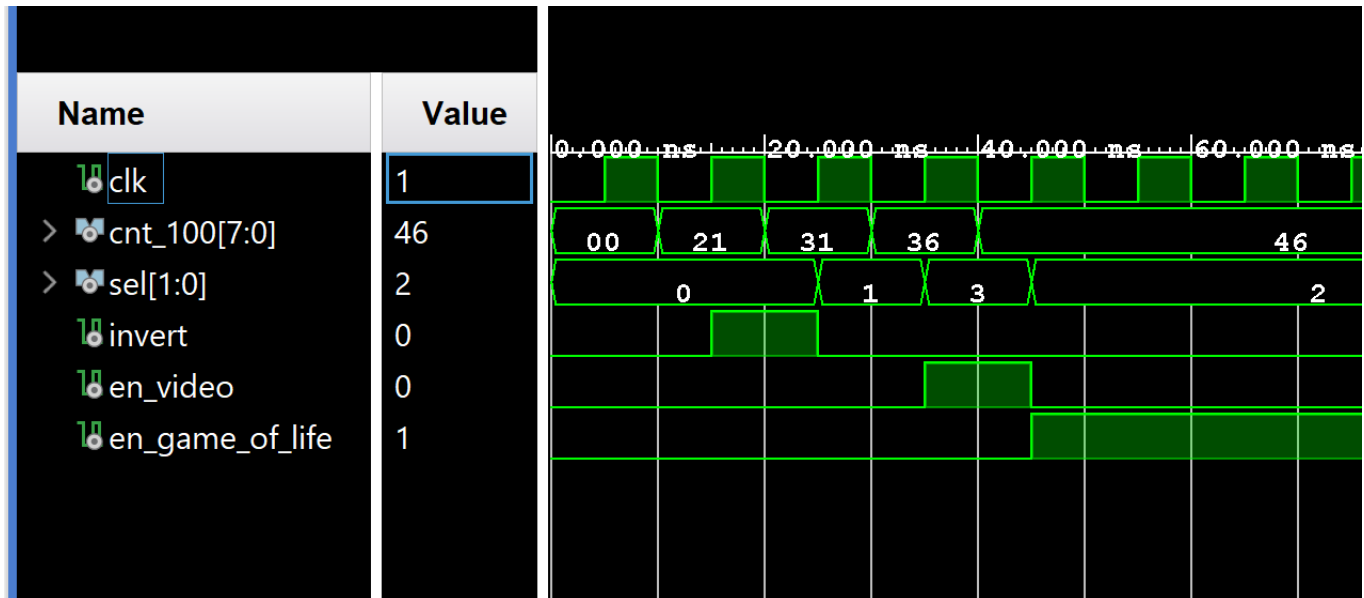
fsm

Výběr obrázku je řízen blokem **fsm**:

- Prvních 30 sekund běhu programu se zobrazuje čítač (nums_source).
- Od 21 do 30 sekundy je obrázek invertován.
- Dalších 5 sekund (31-35s) se zobrazuje obrázek (image_source).
- Dalších 10s: video_source.
- Od 46 do 99s: game_of_life_source.

- Po překročení 99s se čítač zresetuje na 0 a bude opakovat stavy uvedené výše, pokud se HW nevypne.

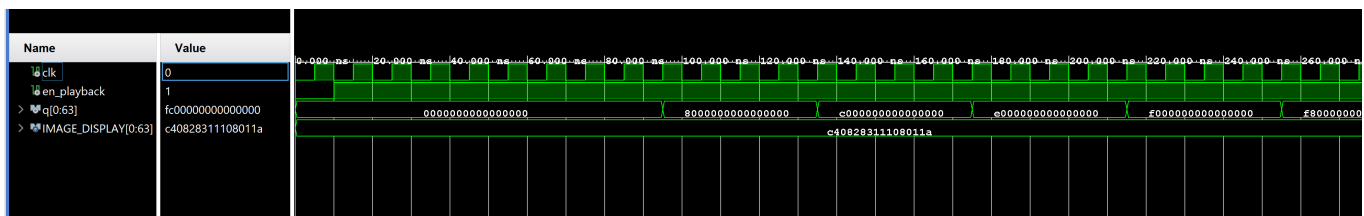
Na obrázku 4 je uvedena simulace testbenche **fsm_tb.vhd**.



Obrázek 4: Ukázka fsm

video_source (ROM Paměť)

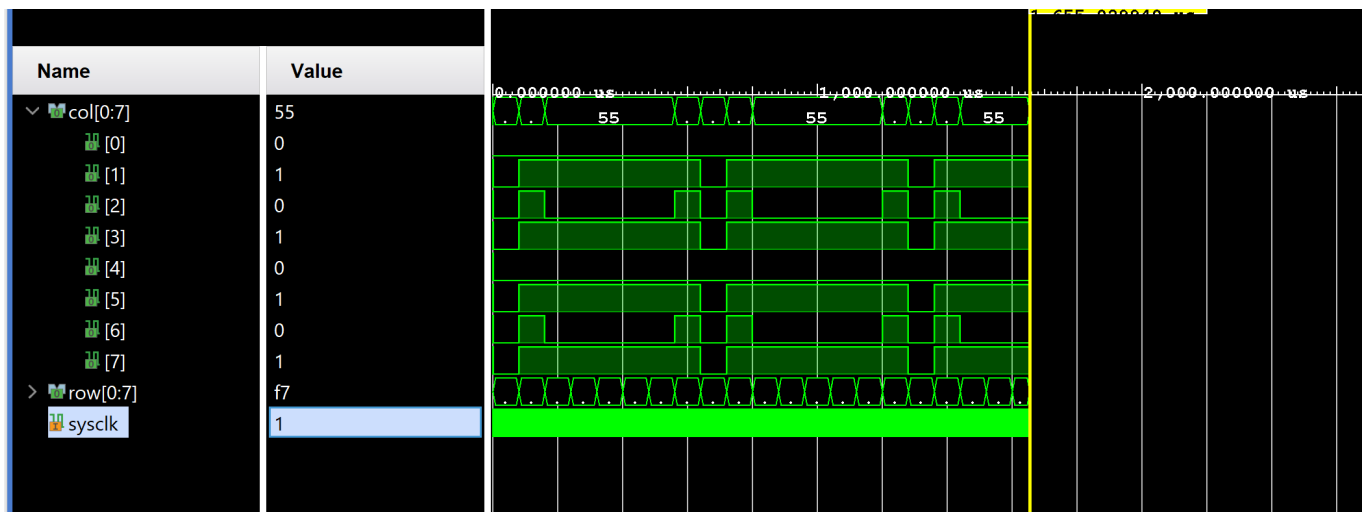
Na obrázku 5 je uvedena simulace testbench **video_source_tb.vhd**. Zobrazovač je nastaven zapisovat další obrázek po každých 4 taktech.



Obrázek 5: Ukázka video_source

Celý systém

Simulace (obrázek č. 6) byla provedena pomocí souborů "design_1_wrapper.vhd" a nastavení "force clock" pro vstup sysclk. Simulace trvá velmi dlouho, a proto je uvedeno jenom cca. 5ms simulace.



Obrázek 6: Ukázka simulace celého systému

Video ukázka na ZYNQu

[zde](#)

Poznámky

- Simulace game of life v některých případech funguje špatně. Opravit, aby to fungovalo přesně jak má, jsem ještě nezvládl.
- Testbench čítače netestuje sám blok bcd_counter, ale jenom generický (generic) čítač. Je zlepšením čítače z druhého podúkolu.
- Kromě uvedených v obrázcích, v projektu jsou další testbench soubory testující další komponenty systému.