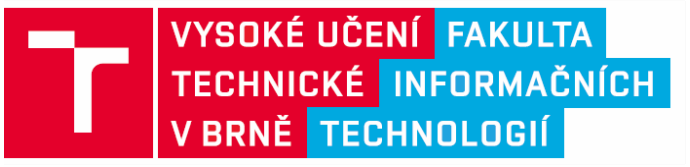
Vysoké učení technické v Brně

Fakulta informačních technologií



IMP - Mikroprocesorové a vestavěné systémy

Hra Life (celulární automat) na maticovém displeji

xkanda00 - Rastislav Kanda

Obsah

1 Popis projektu

Úlohou bolo vytvoriť jednoduchú aplikáciu, ktorá bude na externe pripojenom maticovom displeji LED simulovať priebeh života buniek (tzv. hru Life). Počiatočný stav celulárneho automatu (populáciu) vhodne špecifikovať v kóde, implementovať možnosť krokovania algoritmu, pozastavenia a opatovného spustenia s rozumnou frekvenciou obnovovania stavov. Taktiež bolo potreba ošetriť cyklické okrajové podmienky celulárneho automatu a ako z hlavných súčastí taktiež kontrolu susedstva každej bunky.

1.1 Popis hry

Hra života je dvojstavový, dvojrozmerný celulárny automat, ktorý svojím chovaním pripomína vývoj spoločenstva živých organizmov. Odohráva sa na matici buniek, ktorej stav predurčuje podobu hry v nasledujúcom kroku. Uživateľ určí počiatočný stav a naďalej už beží podľa predpísaných pravidiel:

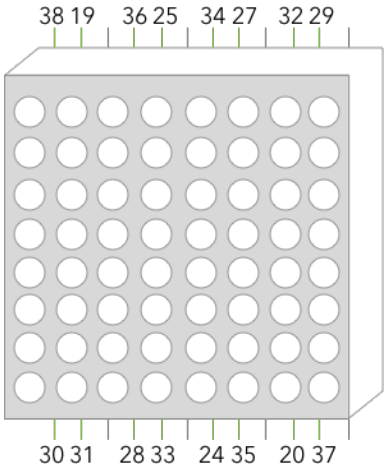
1. Každá živá bunka s menej než dvoma živými susedmi zomrie.
2. Každá živá bunka s dvoma alebo troma živými susedmi zostáva žiť.
3. Každá živá bunka s viac ako troma živými susedmi zomrie.
4. Každá mŕtva bunka s práve tromi živými susedmi oživne.

Pre výpočet nového stavu bunky je dôležité previesť kontolu susedstva buniek (tzv. 9-okolie). Susedstvo je tvorené danou bunkou a jej bezprostredne priľahlými bunkami v horizontálnom, vertikálnom a diagonálnom smere.

2 Externé schémy zapojenia

Maticový displej BM-10EG88MD je potreba zapojiť na FITkit 2.0 na piny MCU JP9 tak ako je to znázornené na obrázku.

Pri zapojení som zvolil kombináciu pinov ktoré rozsvecujú zelenú farbu RG LED displeja. Usporiadanie zapojenia bolo zvolené v poradí kedy piny na JP9 symbolizujú riadky a stĺpce poporadí zdola nahor a teda najspodnejšie piny (č. 19,20) symbolizujú 1. riadok a 1. stĺpec na displeji až po najvrchnejšie piny (č. 37,38), ktoré symbolizujú 8. riadok a 8.stĺpec. Piny 21,22,23,26 boli vynechané. Pin č.26 neskôr spôsoboval z neznámych príčin problémy a tak som sa rozhodol pre najjednoduchšie riešenie a teda premapovať ho na port 24.



Obr. 1: Schéma zapojenia jednotlivých portov z pinov JP9

3 Popis implementácie

Pri implementácii boli prevzaté kódy FPGA z demonstračného projektu Klávesnica v plnom znení. Zdrojové kódy boli použité pre správnu implementáciu ovládania klávesnice.

Operácia modulo bola prevzatá zo Stackoverflow[1], pretože bola potreba ošetriť kontrolu susedov na záporné čísla pre správne fungovanie algoritmu.

3.1 Rozsvietenie displeja

Prvou časťou bolo potreba rozsvietiť displej čo v preklade znamená aby na jednotlivé porty displeja začať prúdiť prúd. To zaisťujú funkcie `setCols()` a `setRows()` ktoré nastavujú na jednotlivé piny úroveň logická 1 alebo logická 0. Je ale potreba aby sa tento postup opakoval vo vhodnom intervale pretože pri jednom nastavení ľubovoľného portu na log.1 a iného portu na log.0 dióda len blikne. Pri vhodne zvolenom intervale cyklu sa displej javí akoby svietil, aj ked iba veľmi rýchlo bliká.

Funkcie `setCols()` a `setRows()` sú volané v hlavnom nekonečnom cykle programu ktoré rozsviecujú displej podľa globálneho dvojdimenzionálneho poľa, ktoré je nositeľom stavu celulárneho automatu.

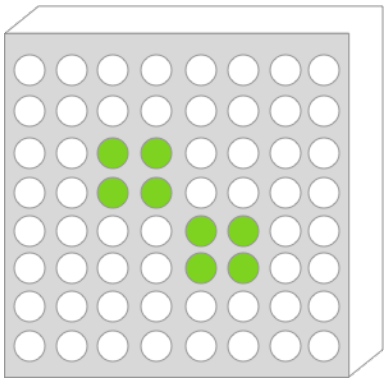
3.2 Algoritmus hry

Najdôležitejšou časťou algoritmu sú 2 globálne dvojdimenzionálne polia, kde jedno je nositeľom hlavných informácií (`matrix[8][8]`) a podľa neho sa rozsvecuje displej a druhé je pomocné(`temp[8][8]`) do ktorého sa prevádzajú výpočty a pri ukončení priechodu algoritmu sa nakopíruje do hlavného.

Pre každú bunku je v algoritme spočítaný počet živých susedov podľa ktorého sa neskôr rozhoduje či danú bunku zhasnúť alebo rozsvietiť. Susedia sa kontrolujú v tzv. 9-okolí ako už bolo spomínané a počet susedov sa ukladá do lokálnej premennej `neigh`. Cyklické okrajové podmienky sú ošetrené pomocou delenia modulo. To bolo potreba rozsíriť o zaobchádzanie so zápornými hodnotami aby nedošlo k indexovaniu mimo pole a taktiež aby v prípade nultého indexu nám vrátilo index č. 7 atď.

Následne je podľa počtu susedov a vyššie uvedených pravidiel rozhodnuté či sa daná bunka má alebo nemá rozsvietiť alebo zhasnúť. Rozhodovanie je implementované ako postupnosť podmienok v cykle.

Frekvencia volania algoritmu je 350 iterácii zmeny riadkov v hlavnom cykle.



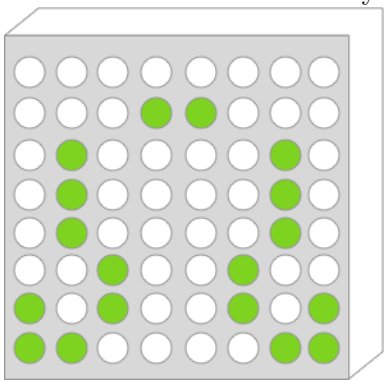
Obr. 2: Počiatočný stav globálnej matice

[1] https://stackoverflow.com/questions/4003232/how-to-code-a-modulo-operator-in-c-c-obj-c-that-handles-negative-numbers

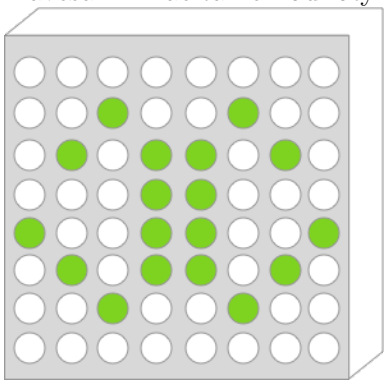
3.3 Popis ovládania

Na ovládanie hry slúži klávesnica FITkitu. Popis ovládania:

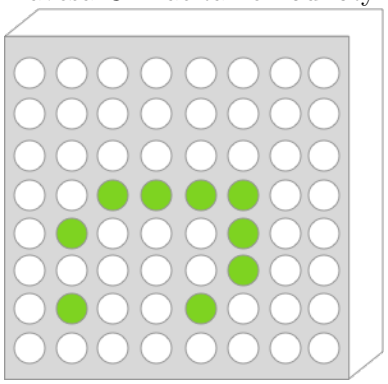
- klávesa 1: Zastavenie/opätovné spustenie behu algoritmu - "*DEBUG*"mód
- klávesa 0: Krok algoritmu
- klávesa A: Načítanie hodnoty z prednastavenej matice



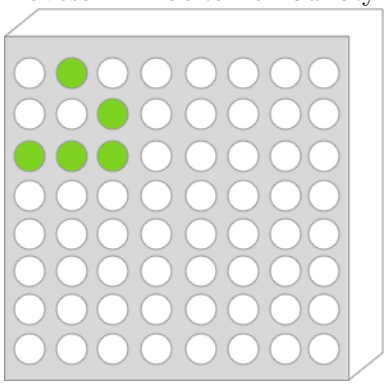
- klávesa B: Načítanie hodnoty z prednastavenej matice



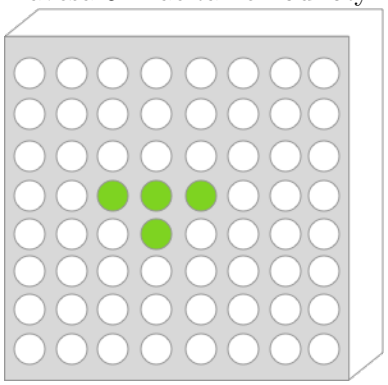
- klávesa C: Načítanie hodnoty z prednastavenej matice



- klávesa D: Načítanie hodnoty z prednastavenej matice



- klávesa 5: Načítanie hodnoty z prednastavenej matice



4 Záverečné zhrnutie

Riešenie projektu prebiehalo v poriadku keďže som projekt začal riešiť z odstupom a mal dostatok času na riešenie. Najväčšou výzvou bolo zozačiatku rozsvietiť maticový displej ale po rozsvietení prebiehalo všetko v poriadku. V rámci projektu boli implementované všetky body ktoré bolo potreba splniť, ako aj cyklické ošetrenie okrajov a krokovanie a zastavenie algoritmu.