

# STACK GAME – FPGA IMPLEMENTATION

**Petr Mutina**

Bachelor Degree Programme (3), FEEC BUT

E-mail: xmutin01@stud.feec.vutbr.cz

Supervised by: Marek Bohrn

E-mail: bohrn@feec.vutbr.cz

**Abstract:** The main goal of this work was to design simplified version of Stack game in VHDL language and implementation to FPGA development kit Spartan-3. There is standard computer keyboard connected via PS/2 interface as input and monitor with VGA interface as output. It was necessary to design 21bit VGA convertor for this project.

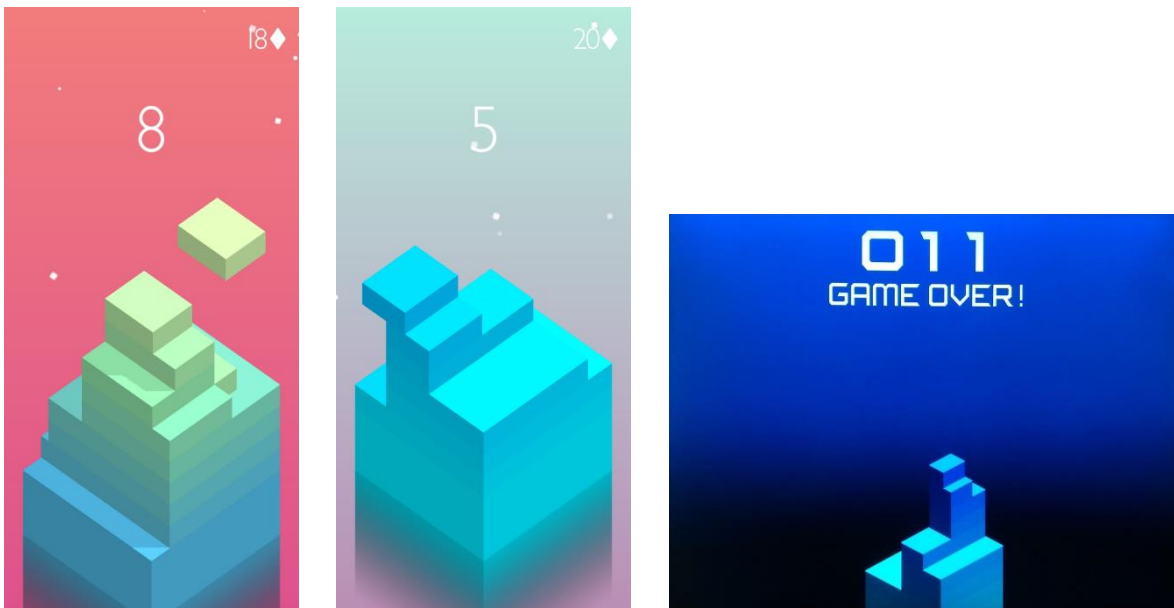
**Keywords:** Stack game, VHDL, FPGA

## 1 ÚVOD

Mezi nejpopulárnější hry vždy patřily ty, které vynikaly jednoduchým grafickým zpracováním a prostými pravidly. Proto vznikl nápad vytvořit funkční popis zjednodušení hry Stack v jazyce VHDL a následně jej implementovat do vývojového kitu Spartan-3 (FPGA). Princip hry spočívá v postupném skládání deskových bloků na sebe, kdy je úkolem dosáhnout co největší výšky. Jako vstupní periferie pro ovládání slouží standardní počítačová klávesnice připojená přes rozhraní PS/2. Dále je k vývojovému prostředku připojen přes 21bitový převodník monitor s VGA rozhráním. Byl kladen důraz na jednoduchost a poutavost grafického zpracování, hra bude pravděpodobně využívána při prezentacích školy (dny otevřených dveří a podobně).

Autorem originální hry (viz obrázek 1 vlevo a uprostřed) je Ketchapp, ta je volně dostupná na:

- **Google Play:** <https://play.google.com/store/apps/details?id=com.ketchapp.stack>,
- **App Store:** <https://apps.apple.com/app/stack/id1080487957>.



**Obrázek 1:** Ukázka originální hry (vlevo a uprostřed) a výsledného designu (vpravo)

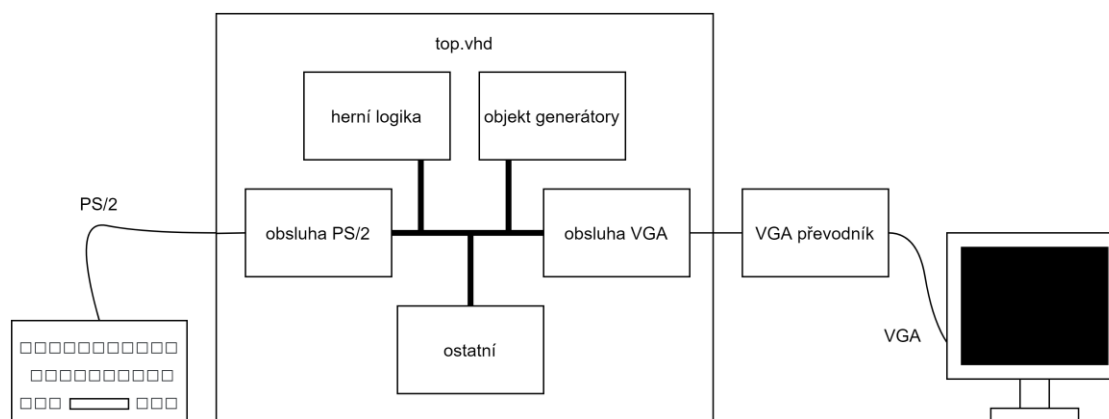
## 2 HARDWARE

K dispozici byly dvě varianty vývojového kitu Spartan-3. S menším (XC3S200) a větším (XC3S1000) FPGA. I přes velkou náročnost návrhu na plochu čipu se díky výrazné optimalizaci (design byl zmenšen více než dvojnásobně) povedlo hru úspěšně implementovat, a to i do menší verze přípravku. Vznikly proto dva návrhy, jejich hardwarové požadavky jsou uvedeny v tabulce 1 v závěru. V obou případech došlo ke splnění podmínek statické časové analýzy, frekvence  $f_{\max}$  je větší než frekvence použitého krystalu 50 MHz [1].

Dále byl navržen 21bitový VGA převodník. Ten sestává ze dvou vodičů pro synchronizační signály a ze tří pasivních rezistorových sítí. Volbou vhodné velikosti odporu těchto rezistorů je umožněno generovat až  $2^7 \cdot 2^7 \cdot 2^7$ , tedy 2 097 152 barev. Převodník dále disponuje piezoměničem a 8bitovým audio převodníkem. Ty však, stejně jako realizace rozšiřující desky, nejsou součástí této práce.

## 3 POPIS HRY V JAZYCE VHDL

Při návrhu byl kladen důraz na to, aby byl design plně synchronní, splňoval statickou časovou analýzu a aby dodržoval základní pravidla návrhu (neobsahuje například nevhodně vytvořené latches plynoucí z neúplných podmínek a podobně). Zjednodušené blokové schéma je zobrazeno na obrázku 2. V případě potřeby byla správná funkčnost některých bloků ověřována pomocí simulací. Následně byl celý design testován, aby došlo k odstranění všech nedostatků.



Obrázek 2: Blokové schéma návrhu

### 3.1 OBSLUHA PS/2

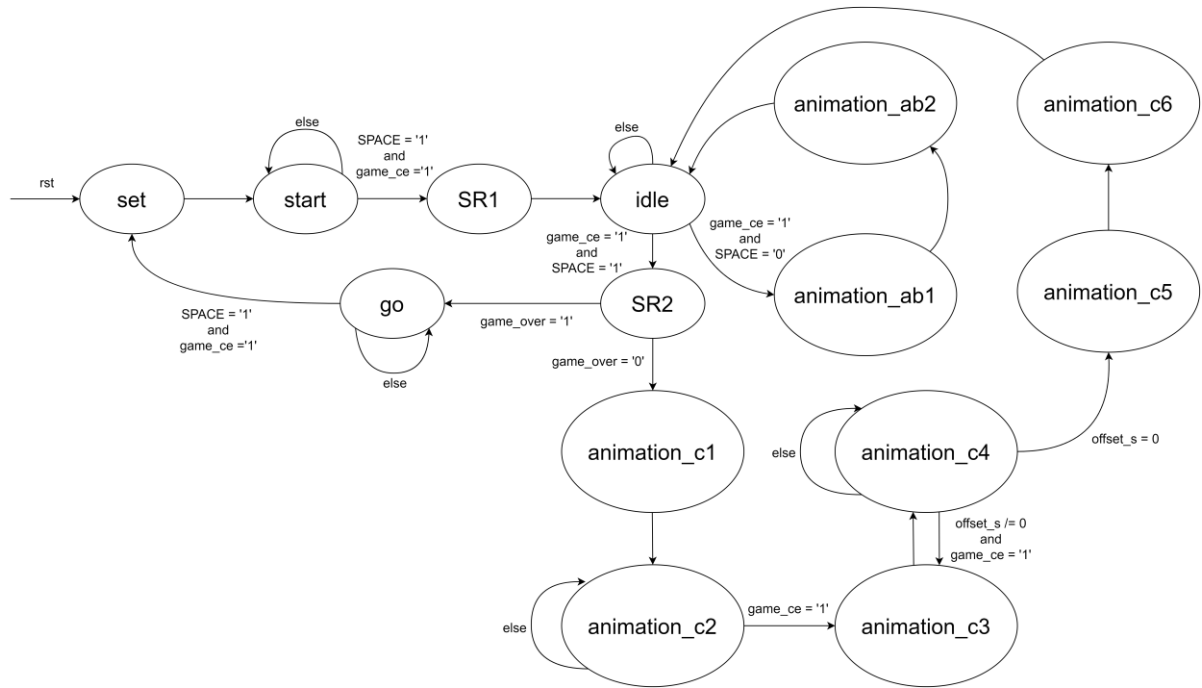
Tento blok, jak již název napovídá, obsluhuje rozhraní PS/2. Počínaje ošetřením zákmitů na sběrnici a vyčítáním datových rámců až po watchdog, který zajišťuje restart bloku v případě poruchy či neočekávaného odpojení/připojení klávesnice. Hra je ovládána pouze klávesou mezerník. V případě absence klávesnice je možné využít tlačítka osazeného přímo na desce plošného spoje kitu. Dále je zde pro případ nouze možnost provést softwarový i hardwarový reset designu.

### 3.2 OBSLUHA VGA

Zde jsou generovány synchronizační pulzy nejen pro VGA rozhraní, ale i pro herní logiku. Aby bylo dosaženo lepší hratelnosti, bylo zvoleno rozlišení 800 na 600 px při 72 Hz – pixel clock (doba zobrazování jednoho pixelu) je shodný jako perioda použitého oscilátoru [2]. Kvůli tomuhle (společně s malou kapacitou paměti FPGA) je nutné generovat informaci o barevném odstínu každého zobrazovaného pixelu pouze pomocí kombinační sítě. Aby byla splněna statická časová analýza, je tato síť vhodně rozdělena na více částí pomocí klopných obvodů. Přidáním určitého množství dalších klopných obvodů je zde dosaženo stejného zpoždění synchronizačních pulzů a signálů určujících barvu (pipelining).

### 3.3 HERNÍ LOGIKA

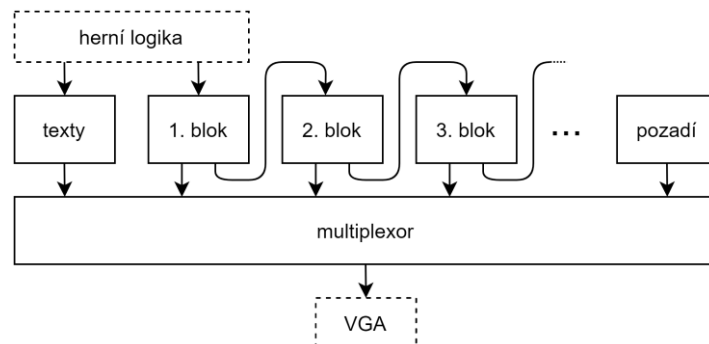
Samotná hra se může nacházet ve třech stavech – START, HRA a KONEC. Dále je potřeba generovat nové bloky, vypisovat texty, vykreslovat animace, vyhodnocovat pozice bloků a podobně. O tohle vše se stará herní logika. Jedná se o jeden velký stavový automat (viz obrázek 3), který dále ovládá příslušné obvody jako posuvné registry či například BCD čítače v případě počítání herního skóre. Svou činnost vykonává pouze v době, kdy není vykreslován obraz (vertikální synchronizace).



Obrázek 3: Stavový automat herní logiky

### 3.4 OBJEKT GENERÁTORY

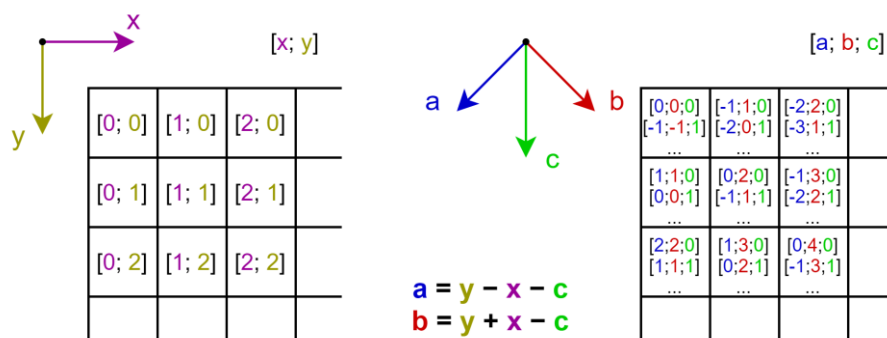
V této části dochází k výpočtu toho, jakou barvu bude mít zobrazovaný pixel. Každý objekt generátor (pro jednotlivé deskové bloky, pozadí a podobně) obdrží informaci o tom, jaká část obrazu se bude vykreslovat, a na základě toho odešle příslušná data do prioritního multiplexoru. Ten vyhodnotí, který objekt má přednost, a přiřadí na VGA rozhraní příslušný barevný odstín. Deskové bloky si mohou předávat příslušná data mezi sebou, herní logika tak určuje pouze parametry bloku prvního (viz obrázek 4). Tím je dosaženo velké úspory prostředků FPGA, zároveň lze jednoduše vytvářet animace.



Obrázek 4: Zjednodušené blokové schéma zapojení objekt generátorů

Pro vykreslování trojrozměrného prostoru byla vytvořena transformace souřadnicového systému X-Y do A-B-C (viz obrázek 5). Pomocí této transformace lze jednotlivé deskové bloky popsat ně-

kolika nerovnicemi. Dále jsou zde uloženy textury s nápisy a jednotlivými číslicemi. Pro jejich tvorbu bylo použito písmo Frostbite Narrow [3].



**Obrázek 5:** Transformace souřadnicového systému X-Y na A-B-C

### 3.5 OSTATNÍ

Tento blok reprezentuje další podpůrné části designu jako například děličky frekvence, generátor synchronního resetu a podobně.

## 4 ZÁVĚR

Byl splněn zadaný cíl vytvořit funkční popis zjednodušení hry Stack v jazyce VHDL, který byl následně úspěšně implementován do vývojového kitu Spartan-3 (viz obrázek 1). Výsledný design vyniká jednoduchým ovládáním a minimalistickým a poutavým grafickým zpracováním. Díky optimalizaci (například přidání offsetů k souřadnicím a indexace 64 předpřipravených 21bitových barev umožnily zmenšit šířku sběrnic) se povedlo návrh zmenšit do takové míry, že jej bylo následně po drobné úpravě (odebrání jednoho deskového bloku) možné nahrát i do menší verze FPGA (viz tabulka 1). Byla splněna statická časová analýza. Dále byl úspěšně navržen 21bitový VGA převodník.

**Tabulka 1:** Hardwarové požadavky výsledného designu pro jednotlivé verze vývojového kitu

prostředek	XC3S200			XC3S1000		
	použito [-]	dostupné [-]	použito [%]	použito [-]	dostupné [-]	použito [%]
slice	1918	1920	99	2959	7680	38
LUT	3522	3840	91	4095	15360	26
klopný obvod	947	3840	24	2462	15360	16
RAM	9	12	75	9	24	37
$f_{\max}$ [MHz]	52,743			60,529		

## REFERENCE

- [1] Xilinx. Spartan-3 FPGA Starter Kit Board User Guide. [elektronický dokument]. UG130 (v1.2). 20. 6. 2008 [cit. 12. 3. 2020].
- [2] SECONS. VESA Signal 800 x 600 @ 72 Hz timing. In: TinyVGA.com: VGA Microcontroller projects. [online]. 2008 [cit. 12. 3. 2020]. Dostupné z: <http://tinyvga.com/vga-timing/800x600@72Hz>
- [3] Isaac K. Frostbite. In: dafont.com. [online]. 28. 9. 2018. [cit. 12. 3. 2020]. Dostupné z: <https://www.dafont.com/frostbite.font>