

SVEUČILIŠTE U SPLITU

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

PROJEKTIRANJE DIGITALNIH SUSTAVA

SEMINARSKI RAD

BREAKOUT GAME

BRUNO GRBAVAC, JELENA MATIĆ

Split, Srpanj 2024.

# SADRŽAJ

1. UVOD.....	2
2. BREAKOUT.....	3
3. VGA .....	4
4. IMPLEMENTACIJA.....	6
5. PROBLEMI I RIJEŠENJA.....	9
6. ZAKLJUČAK I POBOLJŠANJA.....	10

## 1. UVOD

U ovom seminarskom radu ćemo prikazati razvoj i implementaciju klasične arkadne igre "Breakout" pomoću programskog jezika Verilog na FPGA pločici, koristeći VGA za prikaz igre. Verilog je jedan od najraširenijih hardverskih opisnih jezika (HDL), koji omogućava modeliranje, simulaciju i sintezu digitalnih sustava. Cilj ovog rada je detaljno opisati korake potrebne za dizajn, simulaciju i implementaciju ove igre na digitalnom sustavu, uključujući implementaciju VGA prikaza.

Igra Breakout se sastoji od pomičnog reketa, lopte i zida od cigli, s ciljem razbijanja svih cigli pomoću lopte koju igrač kontrolira pločicu. Implementirana igra sadrži tri razine, s rastućom težinom i brzinom, čime se doprinosi dinamičnosti i izazovnosti igre. Razvijena logika igre upravlja pokretima lopte, kolizijama s pločicom i ciglama, te povećanjem težine igre na višim razinama.

Korištenje VGA standarda omogućava vizualizaciju igre na zaslonu, što zahtijeva precizno razumijevanje vremenskih intervala i sinkronizacije signala, što je ključan aspekt digitalnog dizajna. Implementacija VGA prikaza igri dodaje novu dimenziju, omogućujući korisniku interakciju putem grafičkog sučelja.

U ovom radu će biti objašnjeni pravila igre, arhitektura dizajna, implementacija u Verilog-u, te testiranje i verifikacija funkcionalnosti igre. Dodatno će biti obrađeni izazovi i rješenja koja su se pojavila tijekom razvoja, pružajući tako sveobuhvatan uvid u praktičnu primjenu teorijskih znanja iz područja projektiranja digitalnih sustava.

## 2. BREAKOUT

Breakout je klasična arkadna igra u kojoj igrač kontrolira pločicu na dnu ekrana kako bi odbijao lopticu prema blokovima smještenim na vrhu. Cilj igre je uništiti sve blokove koristeći lopticu i pločicu kako bi se prešlo na sljedeću razinu. Igra započinje s jednim životom, a igrač gubi život ako loptica padne ispod pločice.

Kako igra napreduje, razina težine se povećava, uključujući brže kretanje. Igrač može pomicati pločicu lijevo i desno koristeći tipke na ploči kako bi precizno usmjerio lopticu prema blokovima.

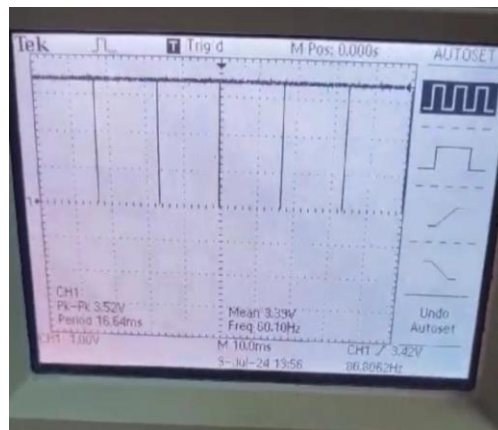
Cilj igre je prijeći poništiti sve blokove na svim levelima. Breakout je izazovna igra koja testira igračeve reflekse, preciznost i strategijsko razmišljanje dok pokušava preživjeti što duže i dobiti.

### 3. VGA

VGA (Video Graphics Array) je standard koji se koristi za prikaz slika na računalnim monitorima i ekranima. Osnovna VGA rezolucija je 640x480 piksela, koja je i korištena za ovaj projekt, iako podržava i druge rezolucije poput 800x600, 1024x768 i više, što ga čini fleksibilnim za različite primjene. Spartan-3E FPGA (Field Programmable Gate Array) je model FPGA-a koji je korišten za implementaciju projekta i VGA sučelja.

VGA koristi analogni signal za prijenos informacija o bojama svakog piksela na ekranu. Za RGB (crvena, zelena, plava) prikaz, koriste se tri analogna signala koji predstavljaju intenzitete ovih boja za svaki piksel. Ovo omogućava bogatstvo boja i detaljno prikazivanje slika. Spartan-3E FPGA se može programirati da generira VGA signale koristeći svoje resurse za logiku i memoriju. Implementacija VGA sučelja na FPGA-u zahtijeva precizno upravljanje sinkronizacijom signala i generiranje odgovarajućih RGB vrijednosti za svaki piksel na ekranu.

VGA standard obično koristi frekvenciju osvježavanja od 60 Hz, što znači da se slika na ekranu obnavlja 60 puta u sekundi. Ovo osigurava glatko prikazivanje pokreta i smanjuje efekte treperenja slike. Budući da je ovo dinamična igra, te uključuje stalno kretanje loptice za određeni broj piksela (definirano kao brzina loptice, za level 1, to je 6 piksela), loptica je prebrza (360 piksela u sekundi) te se nije normalno definirala na ekranu. Implementiran je brojač koji je usporavao ažuriranje prikaza loptice na svako peti prikaz ekrana, tako da je osvježavanje loptice 12 Hz, a ne 60 Hz. To znači da loptica neće prijeći u jednoj sekundi 360 piksela, nego  $360/5 = 72$  piksela i po x i po y koordinati.



Slika 1.

Na slici 1. potvrđeno je vrijeme trajanja VSYNC, što je 16,68 sekundi, kao i u datasheetu [[https://www.epanorama.net/documents/pc/vga\\_timing.html](https://www.epanorama.net/documents/pc/vga_timing.html)]. VGA koristi horizontalnu i vertikalnu sinkronizaciju kako bi monitor znao kada počinje i završava svaki red i svaka linija na ekranu. Horizontalna sinkronizacija (HSYNC) signalizira početak i kraj svakog reda na ekranu, dok vertikalna sinkronizacija (VSYNC) označava početak i kraj svakog okvira na ekranu. Ovi signali omogućuju precizno skeniranje ekrana i sinkronizaciju s grafikom koja se prikazuje.

Pločica u igrici radi na svaki pozitivni brid inputa LIJEVO i DESNO, u toj implementaciji nije korišten brojač nego na svaki pritisak gumba, optica ide 6 piksela lijevo ili desno po x osi ovisno koji gumb je pritisnut, za sva tri levela je ista brzina.

VGA standard je bio široko prihvaćen zbog svoje kompatibilnosti s različitim računalnim sustavima i monitorima. Iako su moderniji standardi poput HDMI-a i DisplayPort-a postali popularniji za visokokvalitetni grafički prikaz, VGA ostaje prisutan u mnogim uređajima, često kao jeftinija opcija ili za podršku starijim uređajima.

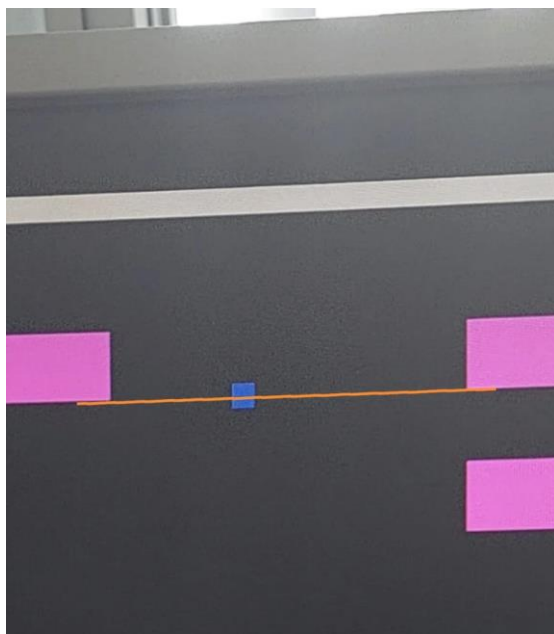
Ukratko, VGA je igrao ključnu ulogu u razvoju računalnih grafičkih tehnologija i i dalje je važan za određene primjene unatoč napretku novijih tehnologija.

## 4. IMPLEMENTACIJA

Budući da je bilo par problema tijekom izrade ovog projekta, napravljene su dvije verzije igre, v1 i v2, sa svojim prednostima i manama.

V1, početna ideja igre se sastojala od 8 blokova raspoređenih u gornjem dijelu ekrana, loptice i pločice i okolnih rubova. Igrač pritiskom na tipke LIJEVO i DESNO pomjera pločicu da se loptica odbije od nju i na taj način pokušava poništiti sve blokove. Budući da se pločica pomiče samo po x osi, implementirana je kao 10-bitni registar dok je loptica implementirana kao dva 10bitna registra, za x i za y smjer. Za početnu inicijalizaciju postavili smo početne koordinate i pločice i loptice. Također, potrebno je i implementirati smjer loptice, u oba smjera, to su dva jednobitni registra. Obijanje pločice o rubove i o pločicu implementirani su kao preklapanja x i y koordinate loptice s koordinatama rubova, tj. pločice. Pri preklopu s bočnim rubovima, loptica mijenja svoj smjer po x osi, a pri preklopu s gornjim rubom i pločicom mijenja se smjer po y osi. Na sličan način je implementiran i sudar s blokovima. Svaki od osam blokova je registar koji u sebi sadrži informaciju je li poništen ili nije. Ukoliko je poništen loptica se neće odbiti od njega, a ukoliko nije poništen, loptici će se promijeniti smjer po x ili po y osi, ovisi u kojem ga je predjelu „udarila“ te promijeniti stanje registra u „poništen“.

Nakon te implementacije naišli smo na problem levela. Implementirana su 3 levela, svaki sa svojom brzinom loptice, prvi - 3, drugi - 6 i treći – 12 piksela. Pošto koordinate gdje bi mogao nastati sudar dobivamo računajući višekratnike broja brzine loptice, svaki level je trebao imati posebne uvjete sudara. Ukoliko bi svi uvjeti sudara bili jednaki, loptica bi nam previše „upadala“ u blok prilikom sudara prikazano na slici 1.

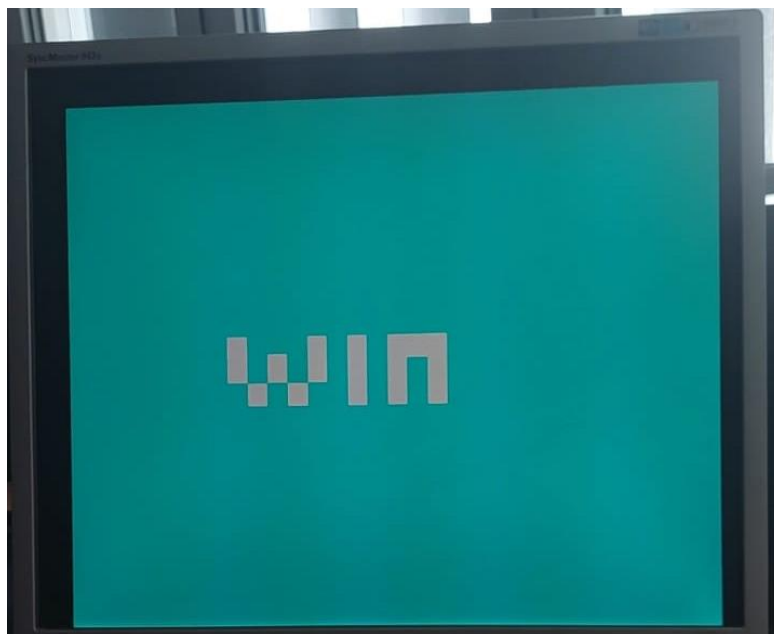


Slika 1.

Sinteza tog koda je trajala predugo te smo došli do ideje da smanjimo broj blokova kako bi mogli implementirati i sintetizirati projekt bez problema.

V2, verzija u kojoj su implementirana tri levela igrice (ali tri bloka za poništavanje) sa svojim posebnim brzinama loptice na 6, 9 i 12 piksela. Leveli su implementirani pomoću tri trobitna registara čije početno stanje iznosi 3'b000. Parna stanja registra označuju da blok postoji, tj. da nije poništen i da je vidljiv na ekranu, a sudarom oni postaju naparni što znači da su poništeni. Leveli su definirani dvobitnim registrom `br_winova`, gdje igricu završavamo kad je `br_winova` jednak 3, što znaci da smo prešli sva tri levela i na ekranu se ispiše win, prikazano na slici 2.





Slika 2.

Igrač gubi igru ukoliko mu loptica pobjegne ispod područja pločice. Igra se može početi ispočetka gdje će igrač biti vraćen na prvi level pritiskom na gumb RESET.

Crtanje svih prikaza na ekranu i njihovo bojanje je provedeno računajući željeni položaj po pikselima. Izračunato je u kojim rasponima piksela su rubovi i blokovi i ostatak i na taj način su crtani. Crtanje je pozivano u if petljama, i provedeno kada su određeni uvjeti petlje zadovoljeni, kao npr. pobjeda ili poraz ili game mode.

## 5. PROBLEMI I RIJEŠENJA

Prvotni problem koji se odnosio na prikaz slike putem VGA na monitoru- korišteni taktni signal na pločici je 50 MHz, što je znatno više od preporučenih 25 MHz potrebnih za VGA protokol. Ova visoka frekvencija je činila sinkronizaciju i praćenje signala izazovnim, stoga smo morali prilagoditi signal tako da smo na svaki pozitivni brid negirali varijablu clk25, čime smo dobili stabilan signal od 25 MHz. Ovaj signal koristili smo za sve potrebne operacije u vezi s uspostavom i prikazom slike preko VGA sučelja.

Nadalje, spomenuti problem s sintezom v1, sa tri levela i predugom sintezom je riješena s v2 i smanjenjem broja blokova. Moguće unaprijeđenje projekta bi bilo istestirati do kojeg broja blokova i levela možemo doći, a da nam sinteza ne traje predugo.

Problem kojeg smo imali i riješili dodavanjem debouncera je taj što svaki stisak tipki LIJEVO i DESNO bi odveo pločicu s kraja na kraj ekrana prebrzo. To je zato što je su pozitivni bridovi prečesto i svaki stisak gumba se očitava kao više stiskova. Nakon dodatka debouncera, riješen je i taj problem.

Problem prebrzog preolaska loptice opisan je u poglavlju iznad gdje je izračunato i novo vrijeme osvježavanja VGA signala.

## 6. ZAKLJUČAK I POBOLJŠANJA

Ovaj projekt je imao za cilj pobliže se upoznati s radom FPGA pločica, pisanjem koda u Verilogu, shvatiti logiku igre i usvojiti određena znanja o sveopćem povezivanju svih navedenih stavki kako bi se dobila funkcionalna verzija igre. Rad na ovom projektu se uveliko provodio na temelju puno pokušaja, odstranjivanja pogreški, i kreiranju logike same igre. Implementacija napisanog koda na pločicu nam je davala prikaz napretka u projektu. Iz svake greške se mogao donijeti koristan zaključak i naučilo se nešto novo te se na osnovu njih točno znalo koji dio koda je nedovoljno dobar. Takav način testiranja koda je zanimljiv i nimalo jednoličan.

Budući da se radi o kompleksnoj igri, možemo uvesti još dosta poboljšanja u budućnosti:

- Dodavanje više blokova,
- Napraviti dvije vrste bloka gdje jednoj vrsti trebaju 2 sudara za poništenje,
- Dodavanje više levela,
- Mogućnost promjene kuta odbijanja loptice od pločicu ukoliko se sudaramo s određenom brzinom
- Uvođenje drugog igrača i zapis bodova,
- Ideja slične igrice da se blokovi slažu jedan na drugoga bez praznina te da se moraju po redu poništavati, bilo bi lakše implementirati, moguće bi bilo ubaciti dosta blokova, samo da su različite boje,...