



Dot Catcher

Bjorn Jongenelen
Quinten Van Ginderen

Digital System Design

Inhoudstafel

Inleiding	3
Doel	3
Uitwerking	4
Hardware	4
Software	4
Problemen en oplossingen	5
Game Logic	5
Willekeurigheid	5
LED matrix	6
Werking	6
Gameloop	6
Werking	6
Buttons	7
Werking	7
Collision	7
Werking	7
Timer	8
Wat leerden we?	8
Conclusie	8
Is het doel bereikt?	8
Uitbreidingen	8

Inleiding

Voor dit vak hebben we leren werken met het Basys 3 Board. Dit gaf ons mogelijkheden om te werken met andere technologieën die nog niet zo gekend zijn.

Als opgave hebben we het project Dot Catcher gekozen: dit is een spel waar we verschillende componenten samengevoegd hebben om zo een creatieve en ludieke opgave te maken.

Voor deze opgave hebben we gebruik gemaakt van de verschillende componenten op het bord en externe hardware onder andere het zevensegmentendisplay, LED matrix en drukknoppen.

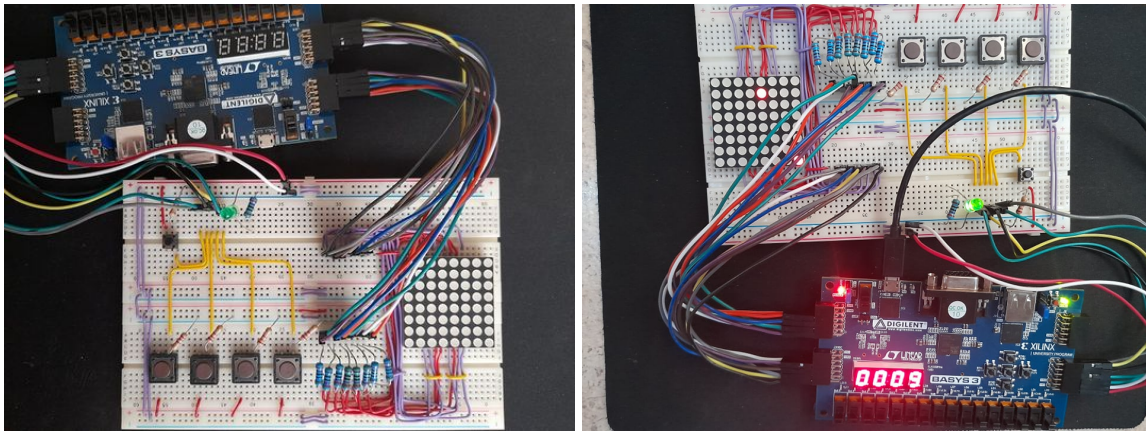
Doel

De bedoeling van dit spel is op de LED matrix één led op een willekeurige plaats te doen branden. Onderaan de matrix schijnt er een tweede led: dit is de speler die we kunnen besturen met behulp van de drukknoppen. Om het spel te winnen moet je tien keer met de speler de willekeurige led raken. Wanneer het spel begint start ook een timer waar je kan zien hoe lang je er over doet.

Hierdoor willen we FPGA aantrekkelijker maken voor mensen met minder kennis of die hiermee net beginnen.

Uitwerking

Hardware



De fysieke opbouw is relatief simpel. Het bestaat uit:

- vier knoppen
- de LED matrix
- de bedrading.

We gebruiken het zevensegmentendisplay als timer voor het project.

Verder staat er op het breadboard nog een groene led. Deze geeft aan of er spanning aanwezig is op de voedingslijnen van het bordje.

Software

De softwarematige uitwerking was iets ingewikkelder.

In de software maakten we modules aan voor het aansturen van

- de zevensegmentendisplay
- de matrix
- de knoppen.

Er zijn ook testbenches opgesteld voor de display en de knoppen.

Deze zijn te bekijken in Vivado® als je het project downloadt. In onze top block zit de game-logica en de verbindingen voor alle onderliggende componenten.

Zoals de Timer, knoppen en LED matrix driver.

Problemen en oplossingen

Tijdens het project zijn we verschillende problemen en errors tegengekomen. Sommige waren hiervan snel op te lossen zoals een error of bug in het programma, andere minder. We ondervonden ook verschillende problemen met de hardware (bijvoorbeeld een knop die hoog werd wanneer het niet werd gevraagd).

Game Logic

Tijdens het testen van ons spel was het al snel duidelijk dat er een kleine kans was dat het target na aanraking berekent was om op dezelfde plaats te blijven staan. Als de speler dit target pakt, zal het target simpelweg verdwijnen. Als dit gebeurt, dien je zo snel mogelijk op de middelste knop op de FPGA te duwen om een nieuw target te krijgen.

Eerst dachten we dat het een bug was waarbij het target buiten de matrix viel. Maar toen we met een led de collision aan het debuggen waren, bleek dat het target op de speler bleef staan, ongeacht de spelers beweging na de aanraking.

Willekeurigheid

Er is geen native random generator in de VHDL libraries. Dus moesten we een andere manier vinden.

Om steeds zo willekeurig mogelijk een nieuw target op te laten lichten moest er logica geschreven worden. Nu, wat is er meer willekeurig dan een persoon? Iedereen zal het spelletje anders spelen. Dus is de willekeur op de beweging van de speler gebaseerd.

Bijvoorbeeld: telkens als de speler een stap naar rechts zet, verplaatst het volgende target zich twee stappen naar onder en één naar links.

Dit nieuwe target verplaatst zich elke keer de speler zich dus verplaatst. En komt tevoorschijn zodra de speler het huidige target pakt.

LED matrix

Op de matrix mochten telkens maar maximaal twee leds oplichten. Als de matrix echter op 400 KHz werkt, lijken er meerdere lampjes op te lichten omdat de signalen niet kunnen volgen (mogelijks door capacitantie op de bedrading).

Door het signaal nog eens door acht te delen draaide de matrix nog op 50 KHz, waardoor er slechts twee leds goed oplichten.

Werking

De matrix vereist twee signalen (X en Y). Deze worden aangemaakt in de gameloop en naar de matrix driver doorgelust. De matrix kan dus maar maximaal één lampje per cycle aanzetten, anders branden er ook leds die op dat moment niet moeten branden. Door deze handelingen snel achter elkaar te herhalen lijkt het alsof er gelijktijdig meerdere leds branden. Op deze manier branden onze twee leds mooi samen. En kan je zien waar de speler en het doelwit zich bevinden.

Gameloop

De game loop draait op 50 KHz apart van de processen voor het bepalen van posities. Dit wil zeggen dat zolang de vorige processen hun logica niet voltooid hebben, de gameloop de code blijft uitvoeren zolang er niets verandert. Het incrementeren van scores bijvoorbeeld. Hier zijn we achter gekomen toen we het scoresysteem aan het implementeren waren. De score werd na één target al meer dan tien. Dit probleem hebben we opgelost met een debouncer.

Werking

Om ons spel sequentieel te laten lopen moeten hebben we een kloksignaal nodig. Een clock divider gaat dit kloksignaal maken aan de hand van de 100MHz klok die in de FPGA zit.

Telkens deze divider tot 255 heeft geteld zal de gamestate geïncrementeerd worden.

De gameloop is een proces dat de gamestate mee krijgt. We hebben acht gamestates. In de eerste gamestate wordt de speler op de LED matrix gezet. Daarna worden alle pixels gereset. En vervolgens komt het target op de LED matrix.

De stappen er tussen zijn om de LED matrix te laten rusten en de pixels terug donker te laten worden.

Dan in de zevende stap wordt gecontroleerd of je speler al dan niet het target raakt. Indien dat zo is zal je score met één punt verhogen en zal het target verspringen.

Buttons

De knoppen gebruikt op het spelbord zijn gewone drukknoppen. Deze zijn echter een redelijk grote bron van storing: het lijkt soms alsof er meerdere keren op de knopjes gedrukt wordt bij een slecht contact.

Eén van de knoppen is vervangen moeten worden tijdens het ontwikkelen omdat deze een te grote storing gaf op het signaal.

Ze worden in de software gedebounced en kunnen verder ontstoord worden met eventueel een condensator schakeling.

Werking

In de software worden de knoppen niet alleen gebruikt om de speler over het veld te verplaatsen maar ook om het volgende target zo willekeurig mogelijk te plaatsen.

Telkens je op een knop duwt zal je speler één plekje verplaatsen. Echter zal het target er minimaal twee verplaatsen. Eén stapje in de Y richting en één stapje in de X richting.

Afhankelijk van hoe je naar je target gaat, kan het ook zijn dat het nieuwe target zich twee posities verplaatst in de X of Y as.

Collision

In Github is het makkelijk te zien dat er in de reference files oude bestanden zitten met code die we eerder hadden geschreven.

Deze files zullen in Vivado® geen fatale foutmeldingen geven maar werken niet zoals verwacht. Deze files zijn er alleen nog voor referenties en snippets uit te halen. In de folder 'clock driven LED matrix' staat de code die we nadien gebruikt hebben om onze finale code op te baseren. Echter in deze files werkte de collision niet en lichtte steeds de tweede led rechtsonder op zonder duidelijke verklaring.

De meest voor de hand liggende oplossing was herbeginnen maar baseren op deze code.

Werking

In de gameloop wordt ook gecontroleerd of de speler- en targetpositie dezelfde zijn. Als dat zo is, krijgt het huidige target de positie van het nieuwe target en zal je score met één verhogen. Als het target op de speler gezet wordt, dien je zo snel mogelijk je target te resetten.

Timer

De timer wordt weergegeven op de ingebouwde zevensegmentdisplay van de FPGA. Hij start met timen van zodra de lampjes op de LED matrix branden. En stopt als er tien targets geraakt zijn.

De timer kan gereset worden met de kleine zwarte knop op het breadboard.

Wat leerden we?

- een LED matrix aansturen op een FPGA.
- het maken van een spel loop in VHDL waar alles parallel loopt.
- het opbouwen van een pseudo willekeurige generator in VHDL.
- het bekijken van resultaten van een testbench.
- het aanmaken van een teller gebaseerd kloksignaal.

Conclusie

Is het doel bereikt?

Wij menen van wel: een simpel spelletje op een geavanceerd toestel.

Als het op een opendeurdag tentoongesteld zou worden, zijn we ervan overtuigd dat er mensen naar zouden kijken en het uitproberen.

Uitbreidingen

Met de multispeler zouden we het persoonlijk nog leuker gevonden hebben, omdat het nu een spelletje is voor één persoon. Mogelijks zouden we controllers kunnen 3D-printen en ze met een kabel naar de FPGA kunnen verbinden.