



Електротехнички факултет, Универзитет у Београду

Документација пројектног задатка из
основа рачунарске технике 2 - ФПГА пројекат
Београд, 3.2.2019.

LEVEL

Предметни професори:

доц. др Марија Пунт
доц. др Жарко Станисављевић

Предметни асистенти:

ас. мс Филип Хаџић
Данко Миладиновић
Јелица Цинцовић

Студенти:

Душан Стијовић, 2017/0145
Лука Шпехар, 2017/0141
Стефан Теслић, 2017/0124

Садржај:

1. Опис пројектног задатка	4
2. Реализација пројекта	4
2.1 VGA I PS/2 протоколи	4
2.2 Комуникација са екстерном меморијом(СДРАМ)	5
2.3 Комуникација између две плочице (I/O пинови)	6
3. Следећи кораци у развоју пројекта	7
4. Списак реализованих датотека	8

1. Опис пројектног задатка

LEVEL представља мултиплејер тип игрице у којој два играча одмеравају вештину погађања противника и избегавања пројектила испаљених од стране противника.

Сваки играч контролише свој свемирски брод преко тастатуре и могуће је видети тренутно стање игре на засебним мониторима.

Играчи играју на одвојеним уређајима, односно сваки играч има своју тастатуру и свој монитор који зависе од једне ФПГА плочице.

Игра се реализује као међусобна комуникација две одвојене плочице.

Сваки играч испаљује пројектиле из свог свемирског брода у смеру свог противника са циљем да га погоди и тиме буде ближи победи.

Када пројектил погоди играча, тај играч губи један живот. Након 3 изгубљена живота, играч губи једно срце и три живота му се регенеришу.

Када противник играча изгуби сва срца и све животе, партија се завршава и постоји могућност поновне игре.

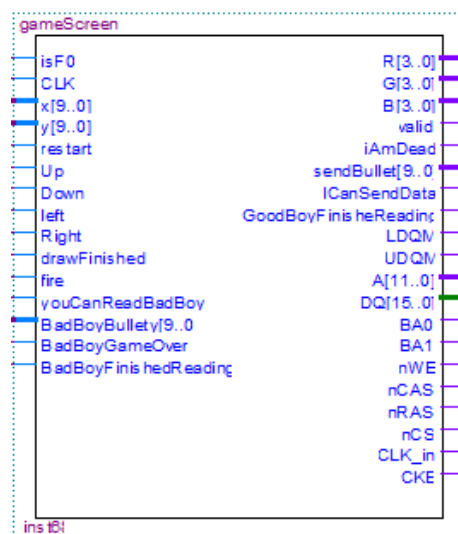
Сваки свемирски брод је слика учитана из СДРАМ меморије која се налази на плочици. Одатле се исчитавају подаци за веран приказ играча.

2. Реализација пројекта

Пројекат је рализован кроз четири протокола:

1. PS/2 протокол
2. VGA протокол
3. Комуникација са екстерном меморијом (СДРАМ)
4. Комуникација између две плочице (I/O пинови)

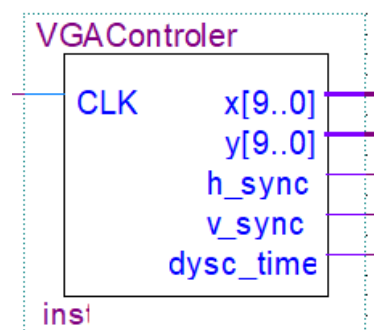
Највећи део логики се извршава у модулу gameScreen.



2.1 VGA I PS/2 протоколи

Имплементација VGA и PS/2 протокола је детаљно одрађена на лабораторијским вежбама те није потребно детаљно објаснити начин функционисања самих протокола.

VGA протокол функционише тако што је модулу VGAController прослеђен клок плочице и унутар тог модула постоје регистри који представљају X и Y координате на основу чега се касније врше одређена

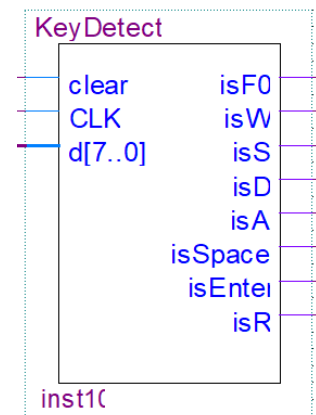
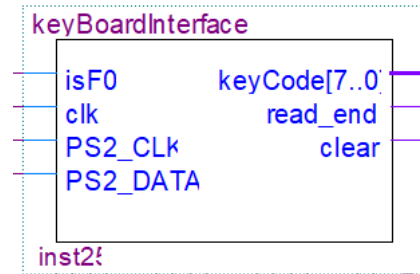


поређења у циљу приказивања различитих садржаја на екрану.

Пинови потребни за сам испис садржаја на екран јесу пинови за боју (VGA_R, VGA_G, VGA_B) и пинови за хоризонталну и вертикалну синхронизацију (VGA_VS, VGA_HS).

PS/2 протокол функционише тако што секвенцијално шаље податке самој плочици. Реализација читања података је урађена кроз секвенцијални бројач и 10 једнобитних регистара, док се подаци који су спремни за слање стављају у осмобитни регистар. Регистар се брише када на пријему детектујемо брејк код (break code). Тиме смо спремили регистар пристиглих података за наредно читање.

Постоји модул *KeyDetect* који има улогу да провери који тастер је притиснут. У случају да је, на пример, isF0 активно, у *keyBoardInterface* се брише прихватни регистар података. Остали тастери су управљачки сигнали за инкрементирање/декрементирање играчевих координата на екрану.

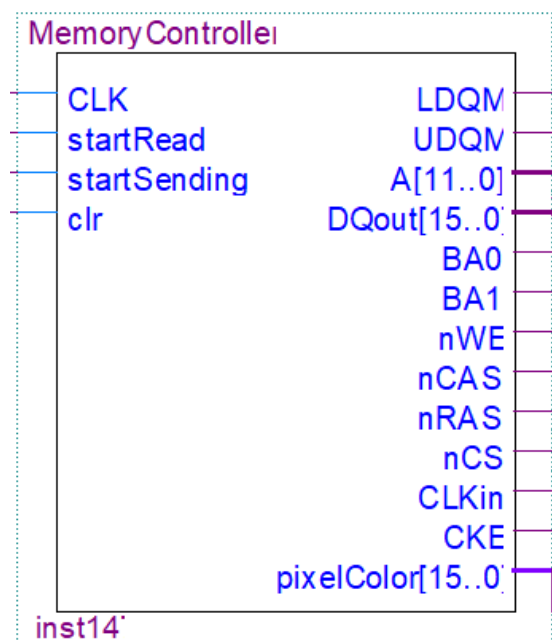


2.2 Комуникација са екстерном меморијом(СДРАМ)

Комуникација између СДРАМ-а и саме игрице је реализована у два важна стадијума. Прво што је потребно јесте да се омогући иницијализација меморије чиме спречавамо квар или неисправан рад меморије.

- 1) Први корак јесте да се пропусти напон кроз плочицу. Потребно је одржавати СКЕ на високом потенцијалу, DQN такође и да на свим другим улазима буде NOP команда.
- 2) Одржавање стабилног напона, клока и NOP инструкција на инпутима за минималних 200µs.
- 3) Пустити тзв. PRECHARGE команду за све банке (PRE или PREA, разлика је у томе што је за PRE потребно мануелно извршити причарџ, док PREA аутоматски одрађује посао за нас)
- 4) Када се све банке налазе у idle стању, потребно је пустити 8 команди за аутоматско ажурирање.
- 5) Последњи корак је да се постави тзв. mode register који нам говори на који начин користимо нашу меморију. За level је потребно искључиво стање читања података.

Ако су сви горенаведени кораци исправно



iii Поља означена са “NE” су поља која преносе клок или су пинови са напоном или уземљењем те нису употребљиви

- У случају када метак стигне до ивице екрана, Y координата метка се ставља у регистар за чекање и уједно се шаље сигнал “*sendBullet*” док се на пријемној

NE	recieveStatus
NE	sendStatus
sendActiveBit0	finishedReading
sendActiveBit1	hasFinishedReading
recieveActiveBit0	recieveActiveBit1
NE	NE
-	-
sendBullet	recieveBullet
R0	R1
NE	R2
NE	R3
R4	R5
R6	R7
R8	R9
NE	NE
S0	S1
S2	S3
S4	S5
S6	S7
S8	S9

PLOCA A/
PLAYER_1

sendStatus	recieveActiveBit0
recieveStatus	recieveActiveBit1
hasFinishedReading	-
finishedReading	
sendActiveBit0	sendActiveBit1
NE	NE
recieveBullet	sendBullet
S0	S1
S2	S3
S4	NE
S5	NE
S6	S7
S8	S9
R0	R1
NE	NE
R2	R3
R4	R5
R6	R7
R8	NE
R9	NE

PLOCA B/
PLAYER_2

страни на улазном пину генерише сигнал “*recieveBullet*” који прослеђује информацију пријемнику да може да почне да чита податке дате сигнаlima *R[9..0]*, док се на предајној страни (на излазу из регистра координате) шаљу сигнали *S[9..0]*.

-При успешном читању се генерише сигнал *finishedReading*, а на пријемној страни се прихвата сигнал *hasFinishedReading* који говори пошљаоцу да може да очисти регистар за слање података и да се стави у спремно стање прихватања наредног пројектила. Битови *sendActiveBit0/sendActiveBit1* и *recieveActiveBit0/recieveActiveBit1* служе за пренос два синхронизациона бита који се користе за синхронизацију почетка исцртавања почетног екрана и за почетак игре.

Почетак игре је када оба играча притисну тастер “enter”.

Могућност поновног играња (рестартовања) је реализована исто

преко жичица. Наиме, рестарт од стране неког играча је немогућа све док није претходно добијен победник. Када се рунда завршила било који играч може да притисне тастер Р или дугме 0 на плочици како би се игра рестартовала.

3. Следећи кораци у развоју пројекта

Пројекат има потенцијал да се прошири у разним смеровима.

Једна могућност је да се пројекат прошити са још две плочице где постоје још два пара монитор-тастатура за још два додатна играча. Два играча који су у истом тиму могу да деле пројекцију једног екрана и да, наравно, виде својег партнера на свом екрану. Принцип игре остаје исти, противници са друге стране (других монитора) имају циљ да поразе два противника са другог екрана (илити других екрана).

Следећи могући корак у развоју пројекта је подршка за миш и тастатуру. У зависности од периферије која је укључена на плочицу, мења се и сам гејмплеј. Неки играчи могу да користе миш док други могу да користе тастатуру, зависи од тога шта коме више одговара.

Постоји могућност за имплементацију више различитих пројектила и њихова селекција.

Ако би се, на пример, имплементирала 4 додатна пројектила поред стандардног. Корисници би бирали пројектиле притиском на одређени број на тастатури који је

аналоган редном броју пројектила на екрану. У случају да се игра са мишем, корисник може да искористи могућност скривања како би одабрао жељени пројектил.

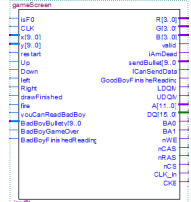
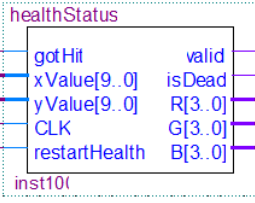
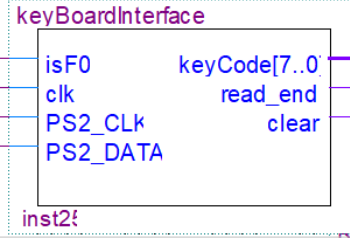
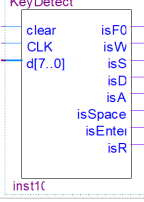
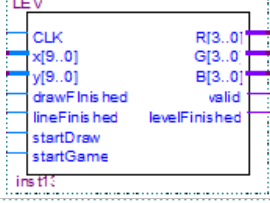
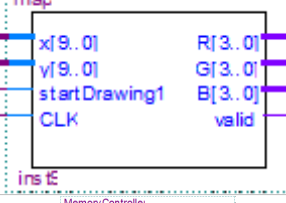
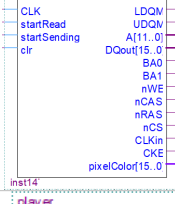
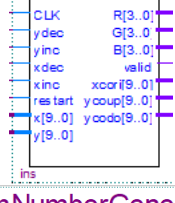
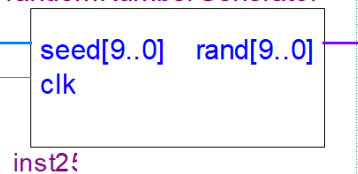
Нове верзије пројектила би имали различито понашање. На пример, један од меткова би могао да има могућност експлозије и да оштети већински погођену мету и уједно може да оштети играче који су у близини погођеног играча. Следећи пројектил би, на пример, могао да има могућност дисперзије по мапи - један пројектил постане 3-4 нова пројектила. Могуће је додати, такође, пројектил који се понаша као “трака енергије” који се испљује држањем дугмета за пуцање у трајању до пуштања дугмета или док се не истроши резерва тог пројектила (муниција).

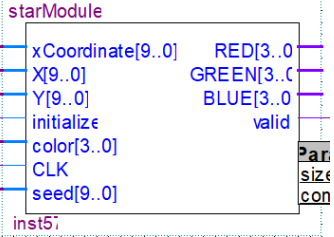
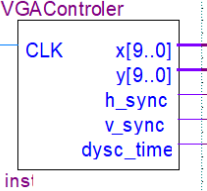
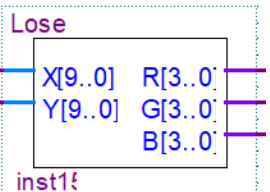
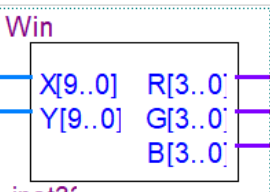
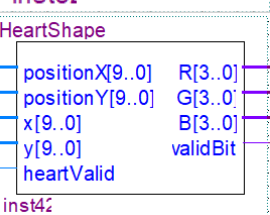
Једно од могућих додатака игрици је могућност имплементације звучних ефеката коришћењем Ардуино компоненте за звук.

Постоји још много могућих додатака, све се своди на то колико је пројектант креативан.

4. Списак реализованих датотека

Назив Датотеке	Изглед Модула	Опис функционалности
bulletBadBoy.bdf		Модул садржи логику за кретање пројектила од тзв. “лошег играча”. У пројекту за првог играча пројектил се креће од краја десне ивице екрана ка левој ивици.
bulletGoodBoy.bdf		Модул садржи логику за кретање пројектила од тзв. “доброг играча”. У пројекту за првог играча пројектил се креће од десне ивице играча до краја десне ивице екрана.
bulletShapeGoodBoy.bdf		Модул садржи изглед пројектила за другог играча. Првенствено се састоји од компаратора.
bulletShapeBadBoy.bdf		Модул садржи изглед пројектила за првог играча. Првенствено се састоји од компаратора.

gameScreen.bdf		Логика игре. Садржи велики број регистара, како и осталих модула као што су играч, пројектили, статус живота и слично.
healthStatus.bdf		Статус живота играча. Садржи регистре који воде евиденцију о животима и броју срца играча. Садржи такође модуле за срце.
KeyboardInterface.bdf		Модул који служи за интерпретирање пристиглих података од тастатуре. Прослеђује их KeyDetect-у који даље интерпретира податке. Садржи секвенцијални бројач и регистре.
KeyDetect.bdf		Модул је задужен за декодовање пристиглих података и прослеђивање адекватних сигнала даљој логици игре.
LEV.bdf		Почетни екран. Модул се састоји првенствено од компаратора. У другом плану садржи регистре уз помоћ којих се постиже ефекат анимације.
map.bdf		Модул садржи логику исцртавања мапе - исцртавања звезда. Састоји се од више модула starModule.
MemoryController.bdf		Интерфејс за комуникацију са СДРАМ меморијом. Садржи управљачку и операциону јединицу поред прихватних регистра података из меморије.
Player.bdf		Модул који садржи регистре и компараторе који се даље користе за померање играча по екрану.
randomNumberGenerator.bdf		Модул садржи један шифт регистар и хор коло уз помоћ којег се постиже генерисање случајних бројева.

StarModule.bdf		Модул садржи регистре за почетне позиције звездица и информације о бојама звездица. Садржи такође генератор случајних бројева уз помоћ којег се добија ефекат померања играча у свемиру.
VGAController.bdf		Модул уз помоћ којег се остварује исцртавање по екрану.
Win.bdf		Модул се састоји од компаратора и испишује #1 на екрану у ситуацији победе играча.
Lose.bdf		Модул се састоји од компаратора и испишује LOSE на екрану у ситуацији ако је играч изгубио.
HeartShape.bdf		Модул садржи податке о позицији срца које треба да се исцрта како и његов изглед.