Proiect FPGA

PONG

Ionescu Valentin-Alexandru

"Pong (comercializat sub numele de PONG) este unul dintre primele jocuri video arcade;

Este un joc de sport de tenis, care are o grafică simplă bidimensională.In timp ce alte jocuri video arcade, cum ar fi Computer Space, au venit înainte, Pong a fost unul dintre primele jocuri video pentru a ajunge la popularitatea mainstream.

Scopul este de a învinge adversarul într-un joc de tenis de masă simulat câștigând un scor mai mare.

Jocul a fost inițial produs de Atari Incorporated (Atari), care l-a lansat în 1972." Wikipedia

Introducere: Conceptul și regulile jocului

Scopul principal al jocului este obtinerea de 9 puncte acumulate de catre un jucator in urma lovirii mingii de border-ul special din spatele paletei adversarului. Am incorporat 2 moduri diferite de joc : single player , multiplayer(2 jucatori). Tastele prin care se selecteaza modul de joc sunt butoanele de la tastatura "1" si "2".

Multiplayer

1.Se apasa pe butonul "2" de la tastatura.

2. Dupa ce a aparut paleta in partea de sus a ecranului, prin apasarea tastei SPACE jocul poate incepe.

3. Dupa ce un jucator inscrie scorul este actualizat pe afisazul cu 7 segmente , cifra de pe afisaj notata cu HEX0 corespunzand jucatorului numarul 2, iar cifra de pe afisaj notata cu HEX3 corespunde jucatorului cu numarul 1 si totodata in mod automat dupa fiecare punct inscris jocul intra in pauza si asteapta apasarea tastei space pentru a fi reluat.

4. In orice moment se poate apasa pe tasta ESC pentru a se iesi din joc si a incepe un altul.

SinglePlayer(VS computer)

1.Se apasa pe butonul "1" de la tastatura sau se apasa direct pe tasta SPACE.

2.Paleta apare automat in partea de sus a ecranului, iar PLAYER2 va fi o simulare a inteligentei artificiale care incearca sa urmareasca deplasarea mingii si sa contracareze incercarile de a inscrie ale PLAYER1 .

3. Dupa ce un jucator inscrie scorul este actualizat pe afisazul cu 7 segmente , cifra de pe afisaj notata cu HEX0 corespunzand jucatorului numarul 1, iar cifra de pe afisaj notata cu HEX3 corespunde jucatorului cu numarul 2 si totodata in mod automat dupa fiecare punct inscris jocul intra in pauza si asteapta apasarea tastei SPACE pentru a fi reluat.

4. In orice moment se poate apasa pe tasta ESC pentru a se iesi din joc si a incepe un altul.

Tastele folosite

PLAYER1:

A -> miscarea paletei catre stanga

D -> miscarea paletei catre dreapta

PLAYER2:

J -> miscarea paletei catre stanga

L -> miscarea paletei catre dreapta

Taste de control

1 -> SinglePlayer (vs Computer)

2 -> Multiplayer

R -> schimbarea culorii paddului pentru player1 in rosu

G -> schimbarea culorii paddului pentru player1 in verde

B -> schimbarea culorii paddului pentru player1 in albastru

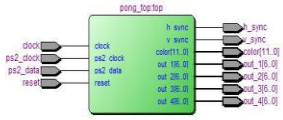
SPACE -> START/PAUSE

ESC -> Exit current game

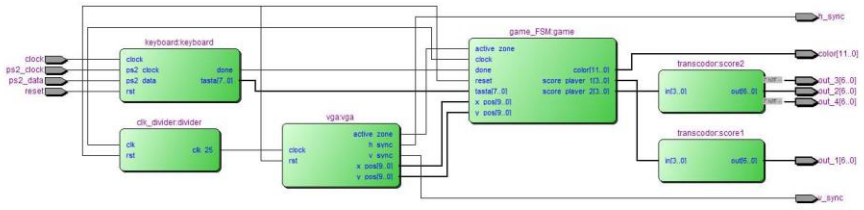
KEY1 de pee FPGA este RESET

Top-level block diagram

Modulul de top



Aceasta reprezinta schema interconectarii blocurilor :



Inputs:

* clock

* reset

* ps2_data

* ps2_clock

Outputs:

* h_sync,

* v_sync,

* color (primii 4 biti reprezinta culaorea rosu urmatorii 4 verde si ultimii patru albastru , acestia sunt legati la portul VGA)

* out_1 (reprezinta scorul PLAYER1 si e legat la HEX0 de pe afisajul cu 7 segmente)

* out_2 (reprezinta scorul PLAYER2 si e legat la HEX3 de pe afisajul cu 7 segmente)

* out_3 (mereu high stinge segmentul HEX1)

* out_4 (mereu high stinge segmentul HEX2)

Module:

* clk_divider – divizor de ceas , primeste ca input un ceas de 50Mhz output un ceas de 25Mhz (pentru rezolutie)

* keyboard –modul care se ocupa cu citirea datelor de la tastatura si verificarea corectitudinii acestora , totdata genereaza un semnal de done doar in momentul in care datele de la tastatura sunt diferite de F0 (break code)

* game _FSM – automatul care controleaza jocul efectiv si contine logica de joc, contine 6 stari din fiecare stare se poate reveni la starea de reset

* vga – modulul vga care asigura sincronizarea pentru ecranul monitorului si care transmite daca este in zona activa pozitia pe x si pe y

* pong - modulul de top in care sunt instantiate si conectate toate aceste module

game_FSM

Acest modul reprezinta nucleul jocului pong. Contine automatul care controleaza flow-ul jocului si totodata logica jocului. El transmite ca

si output scorurile pentru ambii jucatori si culorile ce trebuie afisate pe ecranul monitorului.

Starile automatului

1. STATE_RESET :

```
ball_x <= screen_width >>1;

ball_y <= screen_height >> 1;

paddle2_x <= screen_width >> 1;

paddle2_y <= border_size << 2;

paddle1_x <= screen_width >> 1;

paddle1_y <= screen_height - (border_size << 2);

state <= STATE_PLAYER_SELECT;
```

```
score_player_1 <= 4'd0;

score_player_2 <= 4'd0;

speed_counter <= 6'd0;

computer_counter <= 6'd0;
```

Starea de reset , starea in care sunt date pozitiile pe x,y pentru palete, pozitiile pe x,y pentru minge (mingea este

setata sa apara in centrul ecranului). De asemenea paleta jucatorului 2 nu va aparea decat in momentul in care este selectat modul multiplayer

Se observa totusi ca :

- Pentru paleta jucatorului 1 au fost lasate 13 coloane libere intre aceasta si border (paddle1_y <= screen_height - (border_size << 2) linia de cod amintita insemnand faptul ca pozitia pe y a paletei o sa fie screen_height - border*4 =480- 24 = 456. Pozitia pe x pentru ambele padduri este identica fiind situata in centrul ecranului(320 = screen_width/2). Pozitia pe y a paletei jucatorului 2 este paddle2_y <= border_size << 2 adica 24 ceea ce inseamna exact ca la pozitia pentru paddle 1 o diferenta de 24 linii (ceea ce simbolizeaza sfarsitul lui feature_size + 13 linii lasate negre).

Totodata in aceasta stare sunt resetate scorurile playerilor , counterul ce este folosit pentru a determina viteza cu care se deplaseaza mingea si counterul pentru computer care este folosit in acelasi mod.

- Tranzitia in starea urmatoare nu are la baza vreo conditie si este implicita.

2. STATE_PLAYER_SELECT :

Aceasta stare nu necesita o explicatie amanuntita deoarece este foarte clar care este functionalitatea: ea genereaza semnalul

prin care se alege modul de joc in functie de inputul primit de la tastatura. Totodata in aceasta stare este setata directia pe axa x si y (ball_dx ,ball_dy) odata ce a fost selectat modul de joc.

3. STATE_GAME :

Aceasta este starea care contine logica efectiva a jocului , tranzitia in aceasta facandu-se din starea anterioara numai la apasarea tastei SPACE.

Logica acestei stari este urmatoarea , daca este apasat butonul SPACE se trece imediat in starea de pauza daca este apasat butonul ESC se trece in starea de reset.

Daca este apasata tasta A se verifica o conditie astfel incat sa nu se depaseasca zona destinata jocului (acel feature_size + o jumatate de paleta + latura patratului din care e compusa mingea). Aceasta conditie este logica deoarece numarul de pixeli cu care se misca paleta este egal cu ball_width si pentru o deplasare in stanga inseamna sa scad ball_width pixeli din pozitia curenta a padului si pentru a nu iesi din ecran trebuie sa tin cont de feature size si jumatate din lungimea unui pad.

La fel se intampla si in momentul in care se doreste o deplasare dreapta a padului doar ca de data aceasta pozitia pe x a padului trebuie sa fie mai mica sau egala cu screen_witdh – aceleasi valori de care am tinut cont in partea stanga. Aceleasi lucruri se intampla identic si pentru player2 cu singura diferenta ca este verificat daca semnalul care spune modul jocului este activ.

* De observat este ca : nu folosesc un counter pentru a permite vizualizarea pe ecran a miscarii mingii respectiv a padului ci doar in momentul in care un frame complet este realizat(ceea ce inseamna actualizarea tuturor celor 640x480 de pixeli) se poate actualiza pozitia paddului si a bilei, ceea ce inseamna echivalentul unui counter care numara pana la 640x480. Acest lucru face ca miscarea padului si a bilei sa fie vizibile pentru ochiul uman.

Avem un counter care numara pana la viteza bilei pe care am setat-o. In momentul in care acest counter atinge acea valoare este resetat si se verifica in primul rand daca directia bilei este

la dreapta pe axa x (ball_dx = 1). Daca directia este la dreapta se verifica conditia de overflow (daca bila este in screen_width - feature_size - ball_width) astfel incat bila sa nu iasa din suprafata de joc si se adauga la pozitia pe x a bilei inca un ball_width atata timp cat aceasta conditie este indeplinita.Altfel daca bila merge la stanga se verifica o conditie de overflow (feature_size + ball_width) si se scade din pozitia curenta de pe axa x a bilei ball_width.Daca bila nu mai poate merge la stanga sau la dreapta (adica ambele conditii de overflow nu sunt indeplinite) inseamna ca bila loveste feature-ul si atunci directia x (ball_dx) este setata la 0 sau la 1 in functie de ce margine a ecranului va lovi (stanga sau dreapta).

Daca directia pe y este setata inseamna ca bila merge in jos. In momentul in care bila merge in jos sunt 3 posibilitati :

- fie bila loveste peretele lateral (feature-ul) mergand pe diagonala ori in stanga ori in dreapta in functie de directia pe x.

- fie bila loveste padul

- fie jucatorul 1 nu plaseaza padul in locatia corecta pentru a lovi mingea ceea ce inseamna ca este un punct pentru jucatorul 2.

Conditia pentru ca bila sa loveasca padul este urmatoarea ball_x >= paddle1_x - (paddle_width >> 1)) && (ball_x <= paddle1_x + (paddle_width >> 1)) && (ball_y == paddle1_y - ball_width). Conditia este evidenta deoarece presupune exact suprapunerea pozitiei mingii pe x,y cu jumatatea superioara sau inferioara a padului(pozitia pe x a padului si y reprezinta niste puncte de aceea este pusa aceasta conditie, practic este conditia care este folosita pentru afisarea padului la care am adaugat conditia ca pozitia bilei pe axa x,y sa coincida cu suprafata padului). Daca bila loveste padul viteza cu care se deplaseaza mingea este incrementata, pentru a face aceasta se verifica daca vitea mingii este >1 si se scade 1 din valoarea curenta. Cu cat viteza este mai mica cu atat counter-ul va numara pana la o valorea mai mica ceea ce va face iluzia deplasarii mai rapide a mingii pe ecran.

Daca directia mingii este in jos si a lovit feature-ul (marginea) inseamna ca la pozitia actuala pe y a mingii adaug inca o latura a mingii(indiferent de directia pe x fie ca este stanga sau dreapta mingea isi continua deplasarea in jos.Pentru aceasta este pusa conditia ball_y <= screen_height - feature_size - border_size.

A 3-a posibiliatate este cea in care mingea a lovit suprafata de sub padul jucatorului 1 ceea ce inseamna punct pentru jucatorul 2.In cazul in care aceasta se intampla automatul va trece in starea STATE_PLAYER2_SCORE pozitia mingii va fi setata in centrul ecranului viteza va fi resetata la valoarea prestabilita , pozitia padurilor va fi resetata, iar din aceasta stare se trece in starea game la apasarea tastei SPACE.

In mod identic se intampla si cu paddul2 si exista acelasi 3 posibilitati pentru aceasta directie.

Daca speed_counter nu a ajuns la valoarea prestabilita pentru viteza mingii acesta este incrementat in continuare. Daca player_mode este 0 ceea ce inseamna single_player este incercata simularea inteligentei artificiale. In mod similar exista un counter pentru computer pentru a determina viteza si totodata nivelul de dificultate pentru computer. Computerul se va misca doar in momentul in care se actualizeaza un frame complet si acest counter ajunge la valoarea prestabilita.In momentul in care counter computerului ajunge la acea valoare este verificata conditia de overflow pentru ca padul sa nu iasa din zona de joc in functie de pozitia pe x si pe y a mingii iar, counterul e resetat.

Daca mingea merge la stanga pozitia de pe x a mingii este evident decrementata si atunci si pozitia padului de pe x a lui player 2 este decrementata. Simtetric se intampla pentru o depalsare la dreapta.

Daca counterul nu a ajuns la acea valoare va numara in continuare.

4. STATE_PLAYER1_SCORE :

Starea este evidenta , se poate iesi din aceasta stare apasand pe space pentru a relua jocul sau ESC pentru a-l reseta.

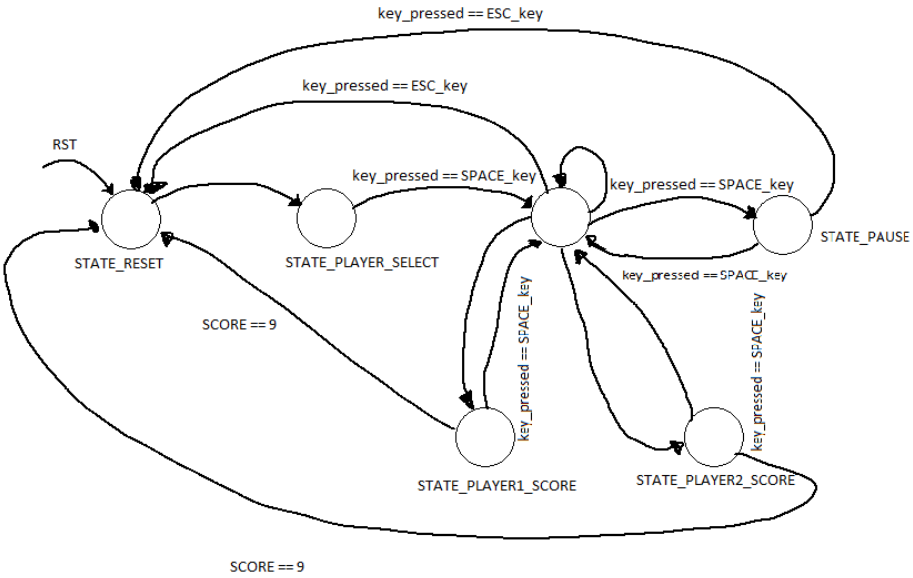
5. STATE_PLAYER2_SCORE :

Starea este evidenta , se poate iesi din aceasta stare apasand pe space pentru a relua jocul sau ESC pentru a-l reseta.

6. STATE_PAUSE :

In starea de pauza se poate intra doar din starea de joc. Din aceasta stare se paote iesi apasand pe SPACE pentru a relua jocul sau ESC pentru reset.

Diagrama starilor automatului :



Transmitrea culorilor catre ecranul monitorului

Transmiterea culorilor este facuta tot de acest modul intr-un proces always sensibil la frontul pozitiv al ceasului, separat de cel al automatului.

Border

Daca este in zona activa si daca pozitia pe x este <= cu border_size (inseamna margina din partea stanga) sau daca pozitia pe x este >= screen_width - border_size (marginea din partea dreapta) sau daca pozitia pe y <= border_size (marginea de sus) sau pozitia pe y >= screen_height - border_size (marginea de jos) atunci culoarea transmisa monitorului este alb.

Feature

Condiitile sunt simetrice pentru feature culoarea fiind roz.

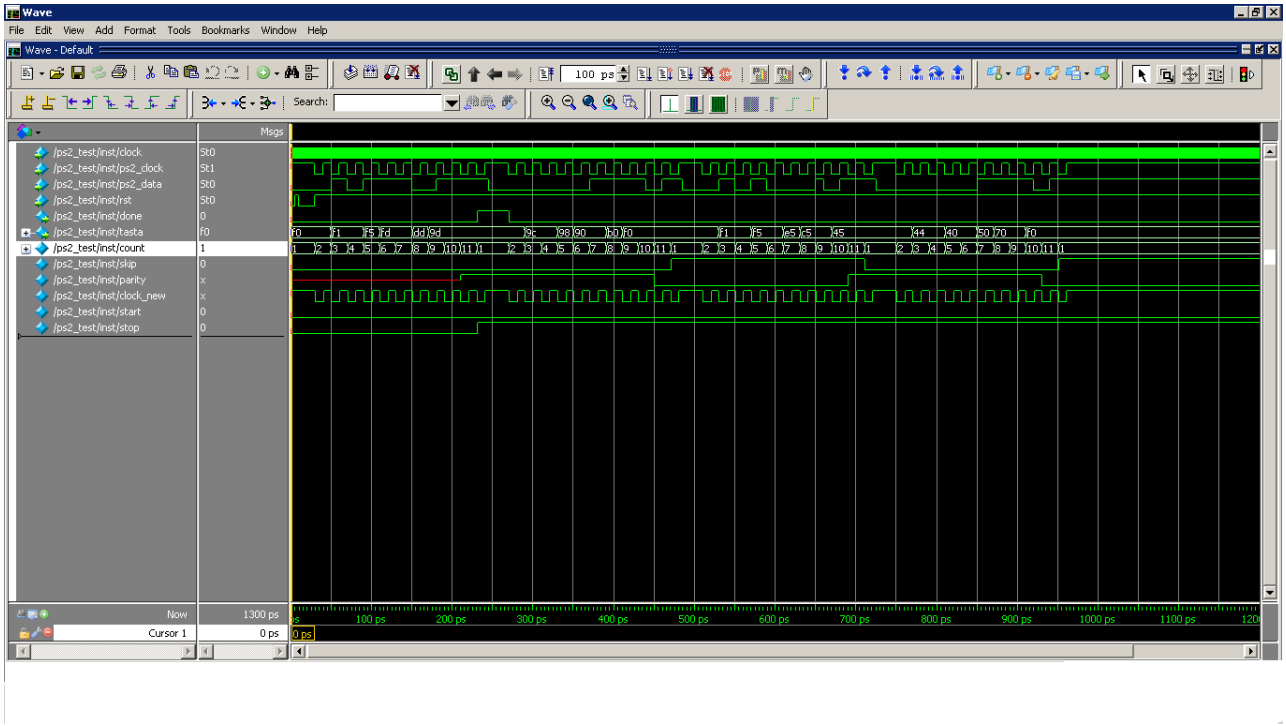
Paddle-uri

Pentru a desena paddle-ul pentru player1 tinem cont ca pozitia pe x si y reprezinta un singur pixel si alegem sa desenam un dreptunghi astfel : daca pozitia pe x se afla intre paddle_x si + sau - paddle_width (ceea ce inseamna ca desenam la stanga si la dreapta pozitiei pe x) si pozitia pe y se afla paddle_y + sau - paddle_height (desenam in susul si in josul pozitiei pe y a paddleului) atunci transmitem monitorului culoarea rosu. Aceasta logica este simtetrica pentru ambele paddle-uri dar si pentru minge avand ball_x si ball_y.

In rest culoarea transmisa este negru , inclusiv daca nu suntem in zona activa.

Modulul keyboard

Acest modul este responsabil cu datele transmise de la tastatura si cu generarea semnalului done care reprezinta faptul ca un frame complet a fost primit si datele transmise sunt corecte (verificari ale bitilor de start/stop si paritate). De asemenea in interior este geenrat un semnal de skip (daca datele transmise sunt F0 - break code adica tasta a fost apasata si dupa a fost ridicata). Semnalul de done nu este activ decat daca frame-ul este corect(dpdv al datelor) si datele transmise sunt diferite de break code. Pentru acest modul am un testbench si voi atasa mai jos o imagine cu formele de unda obtinute. De asemenea in folderul PS2 din acest repo exista un modul de test mai intituiv (afisarea codurilor transmise de tastatura pe segmente).



Din forma de unda se observa un comportament adecvat , de asemenea se observa ca sunt transmise date pe langa F0 dar simularea fiind foarte rapida iar datele fiind transmise la un interval foarte rapid (microsecunde poate chiar nanosecunde) acestea se pierd deoarece semnalul de done nu se actualizeaza suficient de rapid , dar datorita faptului ca nu este posibil ca datele sa fie transmise asa rapid de catre un om , pierderile tind catre 0.

Modulul VGA

Aceste reprezinta modululde sincronizare. Pentru acest modulul nu am avut nevoie de un testbench deoarece functionalitatea modulului poate fi vazuta imediat pe ecranul unui monitor. Mentionez ca folosind modulul de aici doar schimband anumiti parametri si ceasul in functie de rezolutie se poate crea orice rezolutie.

General timing

Screen refresh rate	60 Hz
Vertical refresh	31.46875 kHz
Pixel freq.	25.175 MHz

Horizontal timing (line)

Polarity of horizontal sync pulse is negative.

Scanline part	Pixels	Time [µs]
Visible area	640	25.422045680238
Front porch	16	0.63555114200596
Sync pulse	96	3.8133068520357
Back porch	48	1.9066534260179
Whole line	800	31.777557100298

Vertical timing (frame)

Polarity of vertical sync pulse is negative.

Frame part	Lines	Time [ms]
Visible area	480	15.253227408143
Front porch	10	0.31777557100298
Sync pulse	2	0.063555114200596
Back porch	33	1.0486593843098
Whole frame	525	16.683217477656

Modulul clk_divider

Acesta este un divizor de ceas folosit pentru a diviza cu doi frecventa ceasului de 50Mhz de pe placa DE1 pentru a-l da ca input pt modulul de vga care necesita un ceas de 25Mhz pentru a reda imaginea.

Modulul pong

Modulul de top in care sunt conectate toate celelalte module .

COMENTARII

- in modul multiplayer jucatorii nu pot apasa simultan pe taste deoarece se pierd datele transmise -> pentru rezolvare ar trebui ca datele primite de la tastatura sa fie salvate intr-un FIFO si apoi date ca input la automatul care contine logica jocului

- imbunatatiri care pot fi aduse :

- adaugarea pe ecran a scorurilor

- un logo pe ecran (se poate face o stare intreaga iar tranzitia se poate face dupa un anumit interval de timp)

- un set de reguli care sa apara pe ecran de asemena si instructiuni de folosire

- sunete la lovirea de perete si de pad a mingii

- un mesaj pentru castigator sau un mesaj de gameover

- credite afisate pe ecran