Universitatea Politehnica din Bucuresti
Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei

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 obținerea de 9 puncte acumulate de către un jucător în urma lovirii mingii de border-ul special din spatele paletei adversarului. Am încorporat 2 moduri diferite de joc : single player, multiplayer(2 jucători). Tastele prin care se selectează modul de joc sunt butoanele de la tastatură "1" și "2".

Multiplayer

1. Se apasă pe butonul "2" de la tastatură.

2. După ce a apărut paleta în partea de sus a ecranului, prin apăsarea tastei SPACE jocul poate începe.

3. După ce un jucător înscrie scorul este actualizat pe afisazul cu 7 segmente, cifra de pe afișaj notată cu HEX0 corespunzând jucătorului numărul 2, iar cifra de pe afișaj notată cu HEX3 corespunde jucătorului cu numărul 1 și totodată în mod automat după fiecare punct înscris jocul intră în pauză și așteaptă apăsarea tastei space pentru a fi reluat.

4. În orice moment se poate apăsa pe tasta ESC pentru a se ieși din joc și a începe un altul.

SinglePlayer(VS computer)

1. Se apasă pe butonul "1" de la tastatură, sau se apasă direct pe tasta “SPACE”.

2. Paleta apare automat în partea de sus a ecranului, iar PLAYER2 va fi o simulare a inteligenței artificiale care încearcă să urmărească deplasarea mingii și să contracareze încercările de a înscrie ale PLAYER1 .

3. După ce un jucător înscrie, scorul este actualizat pe afișajul cu 7 segmente, cifra de pe afișaj notată cu HEX0 corespunzând jucătorului numărul 1, iar cifra de pe afișaj notată cu HEX3 corespunzând jucătorului cu numărul 2 și totodată, în mod automat, după fiecare punct înscris jocul intră în pauză și așteaptă apăsarea tastei SPACE pentru a fi reluat.

4. În orice moment se poate apăsa pe tasta ESC pentru a se ieși din joc și a începe un altul.

Tastele folosite

PLAYER1:

A -> mișcarea paletei către stânga

D -> mișcarea paletei către dreapta

PLAYER2:

J -> mișcarea paletei către stânga

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

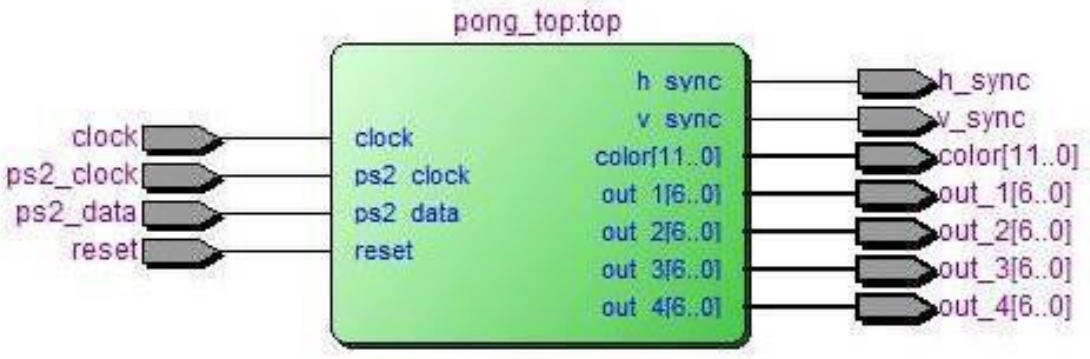
SPACE -> START/PAUSE

ESC -> Exit current game

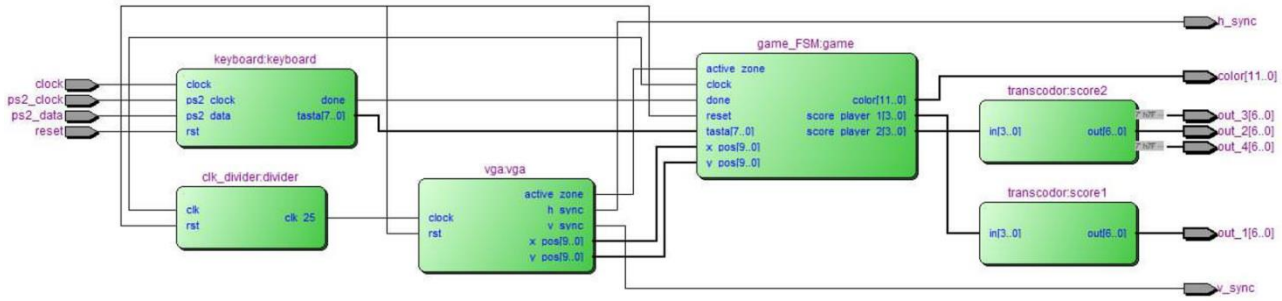
KEY1 de pee FPGA este RESET

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 biți reprezintă culoarea roșu, următorii 4 verde și ultimii patru albastru , aceștia sunt legați la portul VGA)

* out_1 (reprezintă scorul PLAYER1 și e legat la HEX0 de pe afișajul cu 7 segmente) * out_2 (reprezintă scorul PLAYER2 și e legat la HEX3 de pe afișajul 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 in the cente
ball_y <= screen_height >> 1;
paddle2_x <= screen_width >> 1; // paddle2 up in
paddle2_y <= border_size << 2;
paddle1_x <= screen_width >> 1; // paddle 1 posi
paddle1_y <= screen_height - (border_size << 2);
state <= STATE_PLAYER_SELECT; // next state chos
score_player_1 <= 4'd0;//reset player scores
score_player_2 <= 4'd0;
speed_counter <= 6'd0;
computer_counter <= 6'd0;
player_mode <= 1'b0;
ball_speed <= 6'd5;
computer_speed <= computer_speed_default;
red_paddle <= 1'b1;
green_paddle <= 1'b0;
blue_paddle <= 1'b0;
```

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

setată să apară în centrul ecranului). De asemenea, paleta jucătorului 2 nu va apărea

decât în momentul în care este selectat modul multiplayer Se observă totuși că :

- Pentru paleta jucatorului 1 au fost lăsate 13 coloane libere între aceasta și border (paddle1_y <= screen_height - (border_size << 2) linia de cod amintită însemnând faptul că poziția pe y a paletei o să fie screen_height - border*4 =480- 24 = 456. Poziția pe x pentru ambele padduri este identică, fiind situată în centrul ecranului(320 = screen_width/2). Poziția pe y a paletei jucătorului 2 este paddle2_y <= border_size << 2, adică 24 ceea ce înseamnă exact că la poziția pentru paddle 1 o diferență de 24 linii (ceea ce simbolizează sfârșitul lui feature_size + 13 linii lăsate negre).

Totodată, în această stare sunt resetate scorurile playerilor, counterul ce este folosit pentru a determina viteza cu care se deplasează mingea și counterul pentru computer care este folosit în același mod.

- Tranziția în starea următoare nu are la bază vreo condiție și este implicită.

2. STATE_PLAYER_SELECT :

Această stare nu necesită o explicație amănunțita, deoarece este foarte clar care este funcționalitatea: ea generează semnalul prin care se alege modul de joc în functie de inputul primit de la tastatură. Totodată, în această stare este setată direcția pe axa x și y (ball_dx ,ball_dy) odată ce a fost selectat modul de joc.

3. STATE_GAME :

Aceasta este starea care conține logica efectivă a jocului, tranziția în aceasta făcându-se din starea anterioară numai la apăsarea tastei SPACE.

Logica acestei stări este următoarea , dacă este apăsat butonul SPACE se trece imediat în starea de pauză dacă este apăsat butonul ESC se trece în starea de reset.

Dacă este apăsată tasta A se verifică o condiție astfel încât să nu se depășească zona destinată jocului (acel feature_size + o jumătate de paletă + latura pătratului din care e compusă mingea). Această condiție este logică deoarece numărul de pixeli cu care se mișcă paleta este egal cu ball_width și pentru o deplasare în stânga înseamnă să scad ball_width pixeli din poziția curentă a padului și pentru a nu ieși din ecran trebuie să țin cont de feature size și jumătate din lungimea unui pad.

La fel se întâmplă și în momentul în care se dorește o deplasare dreaptă a padului doar că de data această poziția pe x a padului trebuie să fie mai mică sau egală cu screen_witdh – aceleași valori de care am ținut cont în partea stânga. Aceleași lucruri se întâmplă identic și pentru player2 cu singură diferență că este verificat dacă semnalul care spune modul jocului este activ.

* De observat este că : nu folosesc un counter pentru a permite vizualizarea pe ecran a mișcării mingii respectiv a padului ci doar în momentul în care un frame complet este realizat(ceea ce înseamnă actualizarea tuturor celor 640x480 de pixeli) se poate actualiza poziția paddului și a bilei, ceea ce înseamnă echivalentul unui counter care numără până la 640x480. Acest lucru face ca mișcarea padului și a bilei să fie vizibile pentru ochiul uman.

Avem un counter care numără până la viteza bilei pe care am setat-o. În momentul în care acest counter atinge acea valoare este resetat și se verifică în primul rând dacă direcția bilei este

la dreaptă pe axă x (ball_dx = 1). Dacă direcția este la dreaptă se verifică condiția de overflow (dacă bilă este în screen_width-feature_size-ball_width) astfel încât bilă să nu iasă din suprafața de joc și se adaugă la poziția pe x a bilei încă un ball_width atâta timp cât această condiție este îndeplinită.Altfel dacă bilă merge la stânga se verifică o condiție de overflow (feature_size + ball_width) și se scade din poziția curentă de pe axă x a bilei ball_width.Dacă bilă nu mai poate merge la stânga sau la dreapta (adică ambele condiții de overflow nu sunt îndeplinite) înseamnă că bilă lovește feature-ul și atunci direcția x (ball_dx) este setată la 0 sau la 1 în funcție de ce margine a ecranului va lovi (stânga sau dreapta).

Dacă direcția pe y este setată înseamnă că bilă merge în jos. În momentul în care bilă merge în jos sunt 3 posibilități :

- fie bila lovește peretele lateral (feature-ul) mergând pe diagonală ori în stânga ori în dreaptă în funcție de direcția pe x.
- fie bila lovește padul
- fie jucătorul 1 nu plasează padul în locația corectă pentru a lovi mingea, ceea ce înseamnă că este un punct pentru jucatorul 2.

Condiția pentru ca bila să lovească padul este următoarea: ball_x >= paddle1_x - (paddle_width >> 1)) && (ball_x <= paddle1_x + (paddle_width >> 1)) && (ball_y == paddle1_y - ball_width). Condiția este evidentă, deoarece presupune exact suprapunerea poziției mingii pe x,y cu jumătatea superioară sau inferioară a padului(poziția pe x a padului si y reprezinta niste puncte de aceea este pusa aceasta conditie, practic este conditia care este folosită pentru afișarea padului la care am adăugat condiția ca poziția bilei pe axa x,y să coincidă cu suprafața padului). Dacă bila lovește padul viteza cu care se deplasează, mingea este încrementată, pentru a face aceasta se verifică dacă vitea mingii este >1 și se scade 1 din valoarea curentă. Cu cât viteza este mai mică cu atât counter-ul va număra până la o valoare mai mica, ceea ce va face iluzia deplasării mai rapide a mingii pe ecran.

Dacă direcția mingii este în jos și a lovit feature-ul (marginea) înseamnă că la poziția actuală pe y a mingii adaug înca o latură a mingii (indiferent de direcția pe x fie că este stânga sau dreapta, mingea își continuă deplasarea in jos). Pentru aceasta este pusă condiția ball_y <= screen_height - feature_size - border_size.

A 3-a posibilitate este cea în care mingea a lovit suprafața de sub padul jucătorului 1, ceea ce înseamnă punct pentru jucătorul 2. În cazul în care aceasta se întâmplă, automatul va trece în starea STATE_PLAYER2_SCORE, poziția mingii va fi setată în centrul ecranului, viteza va fi resetată la valoarea prestabilită, poziția padurilor va fi resetată, iar din această stare se trece în starea game la apăsarea tastei SPACE.

În mod identic se întamplă și cu paddul2 și există aceleași 3 posibilități pentru această directie.

Dacă speed_counter nu a ajuns la valoarea prestabilită pentru viteza mingii acesta este încrementat în continuare. Dacă player_mode este 0 ceea ce înseamnă single_player este încercata simularea inteligenței artificiale. În mod similar există un counter pentru computer pentru a determina viteza și totodată nivelul de dificultate pentru computer. Computerul se va mișca doar în momentul în care se actualizează un frame complet și acest counter ajunge la valoarea prestabilită. În momentul în care counterul computerului ajunge la acea valoare este verificată condiția de overflow pentru ca padul să nu iasă din zona de joc în funcție de poziția pe x și pe y a mingii iar, counterul e resetat.

Dacă mingea merge la stânga poziția de pe x a mingii este evident decrementată și atunci și poziția padului de pe x a lui player 2 este decrementată. Simtetric se întamplă pentru o depalsare la dreapta.

Dacă counterul nu a ajuns la acea valoare va număra în continuare.

4. STATE_PLAYER1_SCORE :

Starea este evidentă, se poate ieși din această stare apăsând pe space pentru a relua jocul sau ESC pentru a-l reseta.

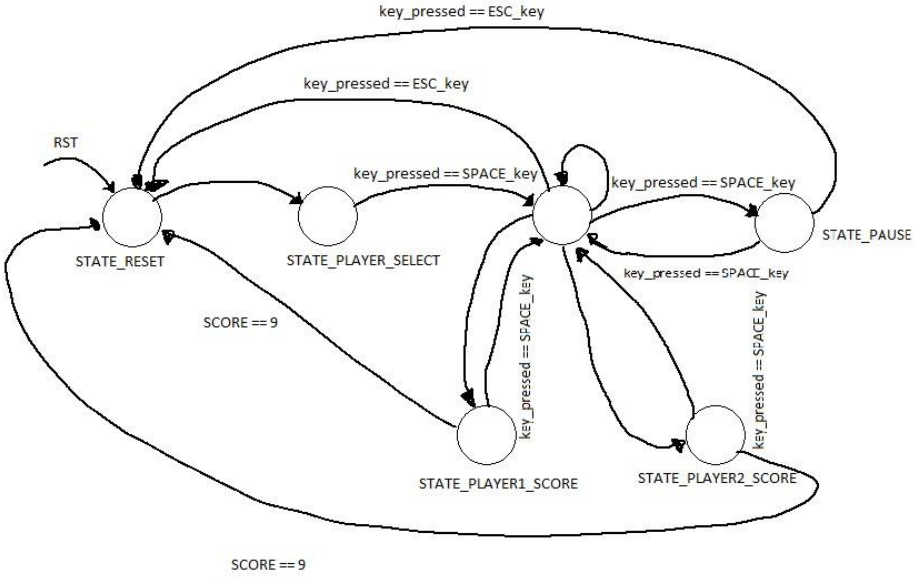
5. STATE_PLAYER2_SCORE :

Starea este evidentă, se poate ieși din această stare apăsând pe space pentru a relua jocul sau ESC pentru a-l reseta.

6. STATE_PAUSE :

În starea de pauză se poate intra doar din starea de joc. Din această stare se poate ieși apăsând pe SPACE pentru a relua jocul sau ESC pentru reset.

Diagrama stărilor automatului :



Transmitrea culorilor către ecranul monitorului

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

Border

Daca este în zona activă și dacă poziția pe x este <= cu border_size (înseamnă marginea din partea stângă) sau dacă poziția pe x este >= screen_width - border_size (marginea din partea dreaptă) sau dacă poziția pe y <= border_size (marginea de sus) sau poziția pe y >= screen_height - border_size (marginea de jos) atunci culoarea transmisă monitorului este alb.

Feature

Condițiile sunt simetrice pentru feature, culoarea fiind roz.

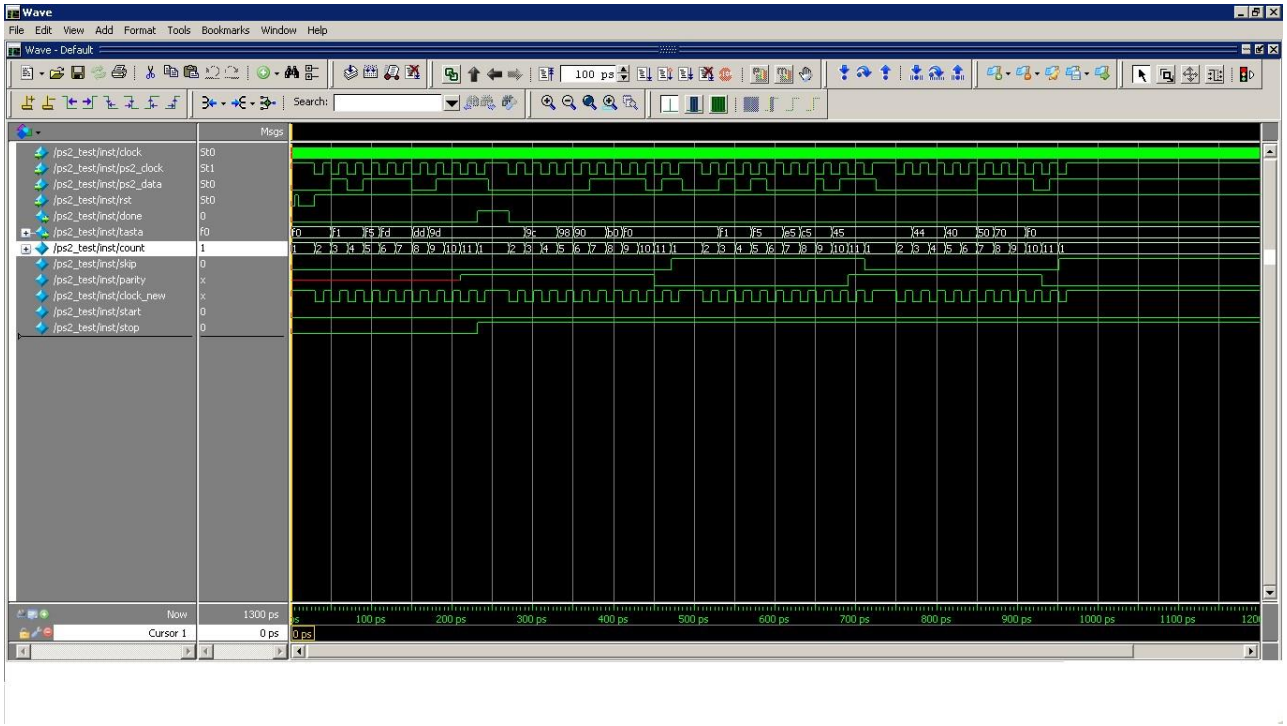
Paddle-uri

Pentru a desena paddle-ul pentru player1 ținem cont că poziția pe x si y reprezintă un singur pixel și alegem să desenăm un dreptunghi astfel : dacă poziția pe x se află între paddle_x și + sau - paddle_width (ceea ce înseamnă că desenăm la stânga și la dreapta poziției pe x) și poziția pe y se află paddle_y + sau - paddle_height (desenăm în susul și în josul poziției pe y a paddleului), atunci transmitem monitorului culoarea roșu. Această logică este simtetrică pentru ambele paddle-uri dar și pentru minge avand ball_x si ball_y.

În rest culoarea transmisă este negru, inclusiv dacă nu suntem în zona activă.

Modulul keyboard

Acest modul este responsabil cu datele transmise de la tastatură și cu generarea semnalului done, care reprezintă faptul că un frame complet a fost primit și datele transmise sunt corecte (verificări ale biților de start/stop și paritate). De asemenea, în interior este generat un semnal de skip (dacă datele transmise sunt F0 - break code, adică tasta a fost apăsată și după a fost ridicată). Semnalul de done nu este activ decât dacă frame-ul este correct (dpdv al datelor) și datele transmise sunt diferite de break code. Pentru acest modul am un testbench și voi atașa mai jos o imagine cu formele de undă obținute. De asemenea, în folderul PS2 din acest repo există un modul de test mai intituiv (afișarea codurilor transmise de tastatură pe segmente).



Din forma de undă se observă un comportament adecvat , de asemenea se observă că sunt transmise date pe lângă F0, dar simularea fiind foarte rapidă iar datele fiind transmise la un interval foarte rapid (microsecunde, poate chiar nanosecunde) acestea se pierd, deoarece semnalul de done nu se actualizează suficient de rapid, dar datorită faptului că nu este posibil ca datele să fie transmise așa rapid de către un om, pierderile tind catre 0.

Modulul VGA

Acesta reprezintă modulul de sincronizare. Pentru acest modul nu am avut nevoie de un testbench deoarece funcționalitatea modulului poate fi văzută imediat pe ecranul unui monitor dar am decis totusi sa fac un testbench. Menționez că folosind modulul de aici doar schimbând anumiți parametri și ceasul în funcție de rezoluție se poate crea orice rezoluție.

General timing

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

Horizontal timing (line)

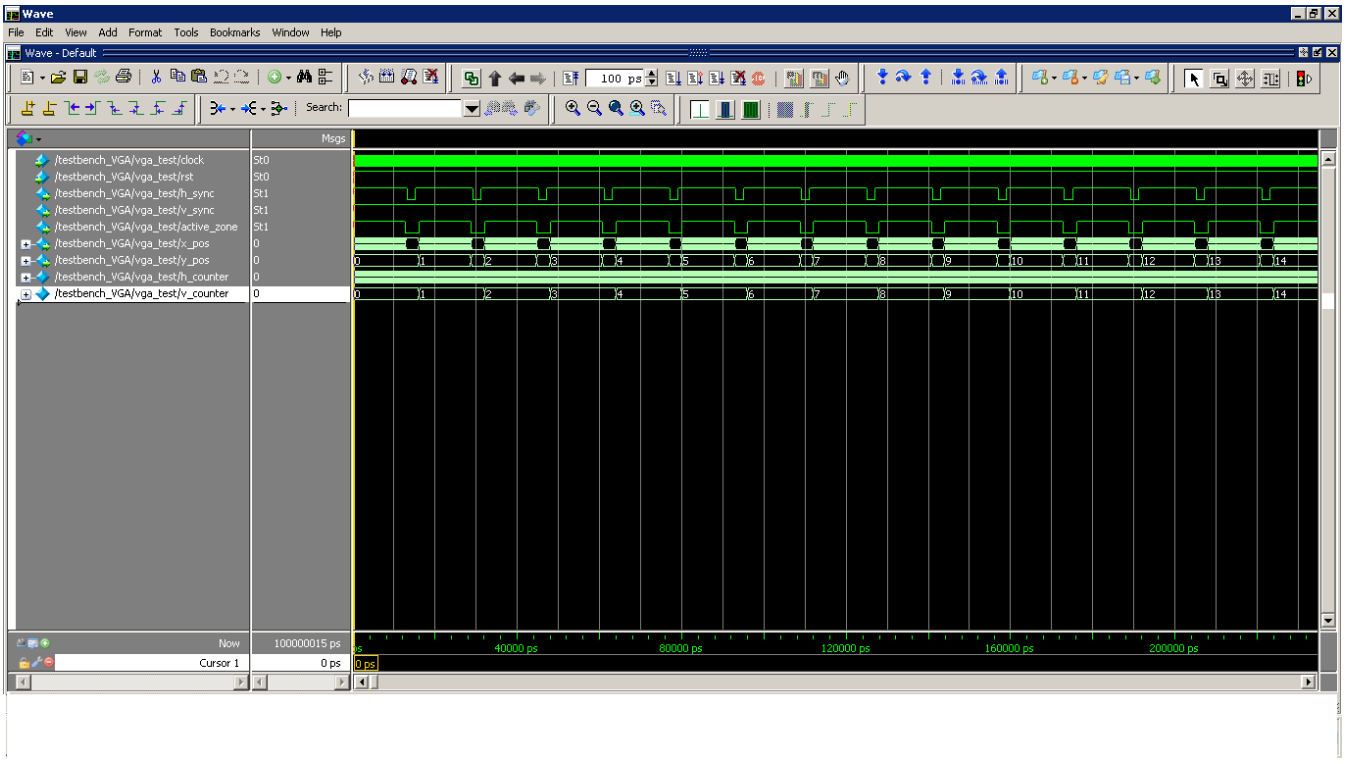
Polarity of horizontal sync pulse is negative.

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 frecvența ceasului de 50Mhz de pe placa DE1 pentru a-l da ca input pentru modulul de vga care necesită un ceas de 25Mhz pentru a reda imaginea.

Modulul pong

Modulul de top în care sunt conectate toate celelalte module.

COMENTARII

- în modul multiplayer, jucătorii nu pot apăsa simultan pe taste, deoarece se pierd datele transmise -> pentru rezolvare ar trebui ca datele primite de la tastatură să fie salvate într-un FIFO și apoi date ca input la automatul care conține logica jocului - îmbunătățiri care pot fi aduse :

- adăugarea pe ecran a scorurilor

- un logo pe ecran (se poate face o stare întreagă, iar tranziția se poate face după un anumit interval de timp)

- un set de reguli care să apară pe ecran, de asemenea, și instrucțiuni de folosire

- sunete la lovirea de perete și de pad a mingii

- un mesaj pentru câștigător sau un mesaj de gameover

- credite afișate pe ecran