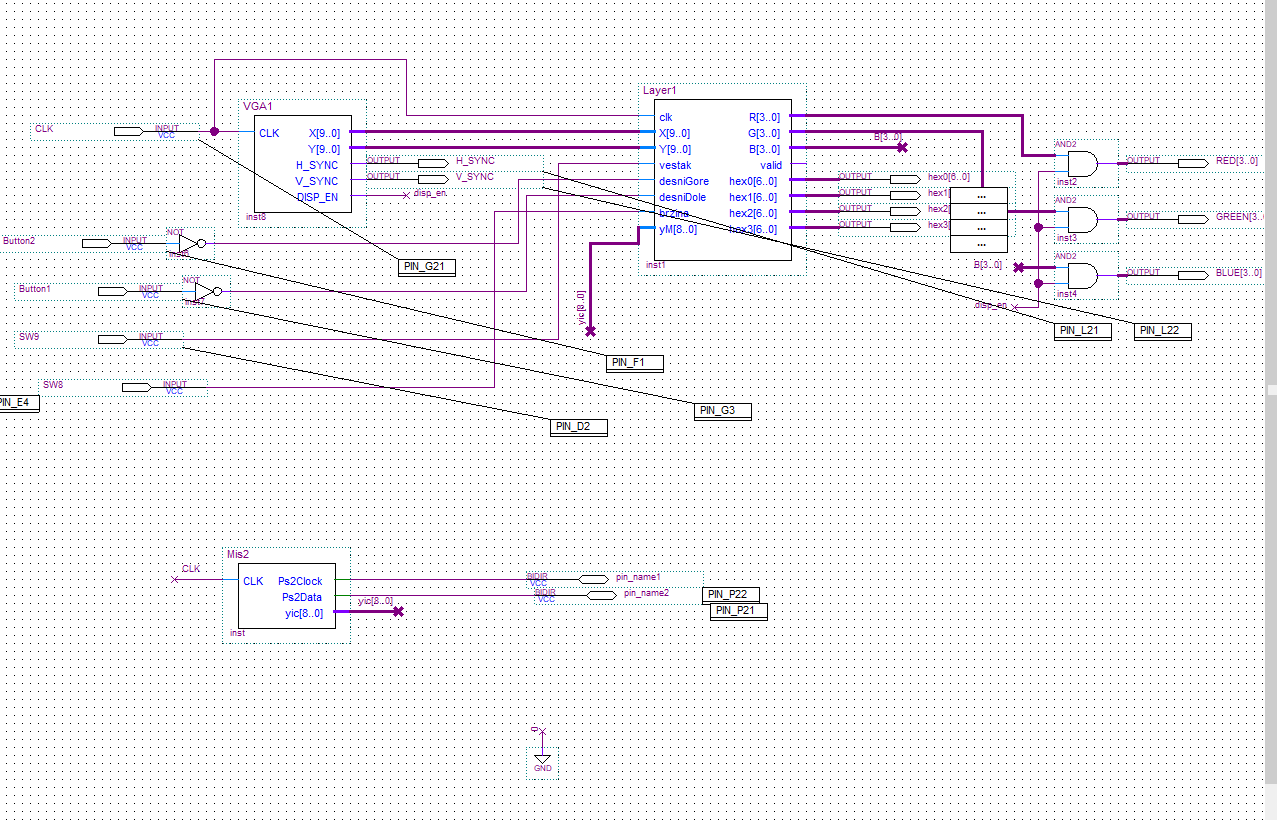
**IGRA PONG**

**OPŠTE KARAKTERISTIKE IGRE**

Igra Pong je 2D pojednostavljena simulacija igre nalik na stoni tenis. Svaki igrač na raspolaganju ima dve palice kojima se kreće gore-dole i sprečava lopticu da stigne do ivice ekrana koja odgovara tom igraču. Ukoliko u tome ne uspe, drugi igrač dobija poen. Nakon što je postignut poen, loptica nastavlja kretanje iz sredine ekrana, u smeru igrača koji je postigao poen.

Celokupna igra je rađena po ugledu na igru koja se može pronaći na ponggame.org uz neke modifikacije koje su urađene u skladu sa Altera pločicom koja je bila na raspolaganju. Za beleženje rezultata nije korišćen sam ekran nego sedmosegmentni displej. Leva palica se kontroliše mišem koristeći PS2 protokol, dok se desna palica može kontrolisati ručno ili se kontrola može prepustiti veštačkoj inteligenciji koristeći prekidač SW9. Ukoliko je veštačka inteligencija uključena, prekidačekm SW8 se reguliše težina. Ukoliko je SW8 isključen, biće lakše postići poen, a ako je isključen biće teže. Ukoliko je veštačka inteligencija isključena, kontrola palice se vrši korišćenjem dugmića button1 i button2.

**GLAVNA ŠEMA**



*Slika 1: Glavna šema*

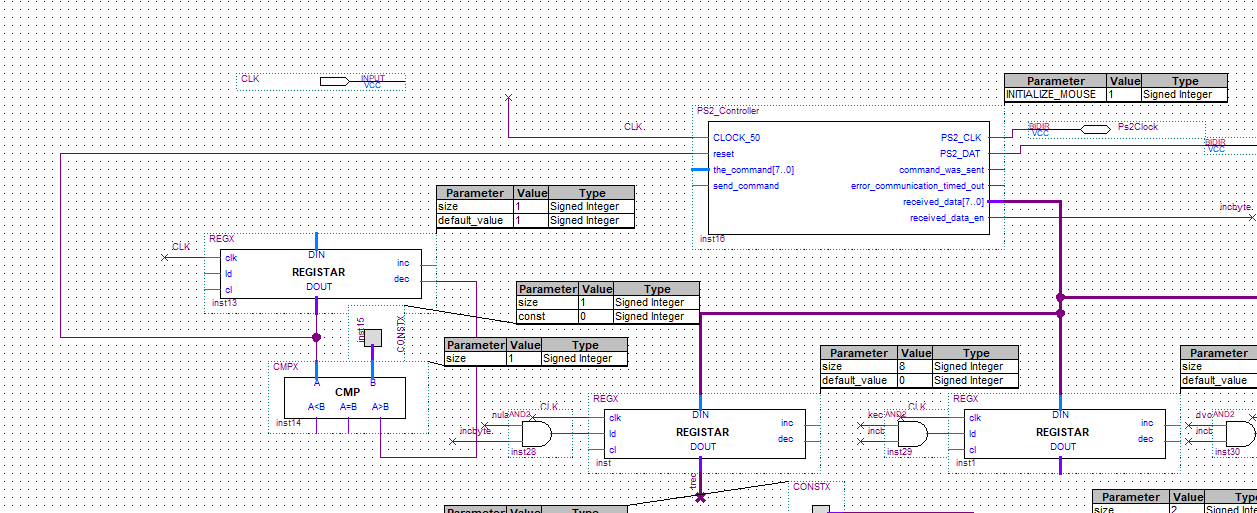
Glavna šema za ulazne pinove ima clock pločice učestanosti 50MHz, prekidače i dugmiće koji omogućavaju komunikaciju sa korisnikom, kao i ulazno/izlazne pinove koji služe za komunikaciju sa mišem preko PS2 protokola. Takođe postoje 4x3 izlazna pina u vidu RGB nijansi koji se koriste za bojenje ekrana koristeći VGA protocol, kao i H\_SYNC i V\_SYNC koji služe za sinhronizaciju piksela koji se šalju preko VGA protokola. Postoje i pinovi kojima se omogućava povezanost sa diodama na sedmosegmentnom displeju. Od pomoćnih modula postoje VGA1, Mis2 i Layer1.

VGA1 služi da generiše koordinate X i Y koje se dalje prosleđuju komponenti Layer1 koja za datu koordinatu generiše boju kojom treba obojiti dati piksel na ekranu. Ukoliko su koordinate u opsegu kada treba odraditi horizontalnu ili vertikalnu sinhronizaciju (krajevi intervala rezolucije ekrana) tada su signali H\_SYNC i V\_SYNC jednaki jedan. Isto važi i za signal disp\_en koji je aktivan kada treba omogućiti prikaz boja na ekranu za trenutnu koordinatu. Od ulaznih signala za komponentu VGA1 potreban je samo clock uređaja.

Kao što je već rečeno, komponenta Layer1 generiše boju za zadatu koordinatu u RGB formatu. Pored toga, ovaj modul za ulaze ima i prekidače i dugmiće kojima korisnik obavlja komunikaciju sa igrom. To je neophodno da bi se uključila/isključila veštačka inteligencija (ulaz veštak), kao i da bi se kontrolisala brzina i kretanje same palice. Na ulaz Y se dovodi pomeraj po Y koordinati iz komponente Mis2 koji predstavlja pomerajaj u odnosu na poslednji poslati pomeraj napravljen mišem. Od izlaznih pinova postoji valid kojim se označava da trenutni layer ima prioritet u odnosu na druge, ali pošto je ovo jedini layer na šemi, ovaj signal se ne koristi. Takođe se šalju signali hex0…hex3 kojima se brojevi ispisuju na sedmosegmentnom displeju.

**MIŠ2**

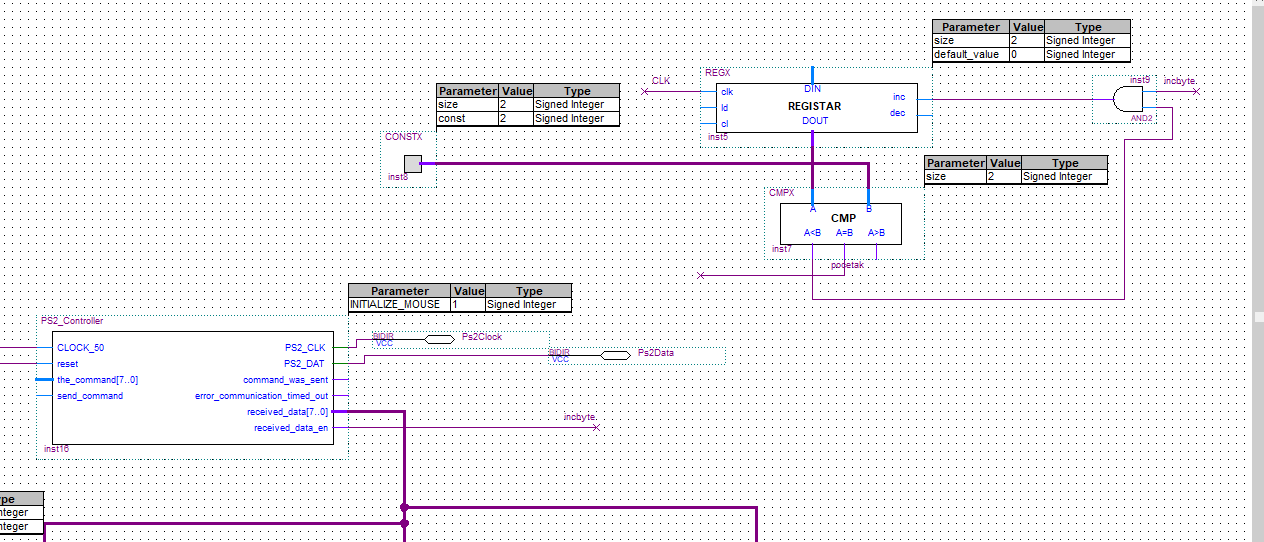
Najvažnija komponenta za korišćenje miša je PS2\_Controller koja šalje bajt po bajt koordinate i statusni bajt u paketima od po 3 bajta putem received\_data. Kada je novi bajt u okviru tog paketa spreman za primanje, izlaz received\_data\_en postaje aktivan. Potrebni su i dvoulazni pinovi za PS2 Clock i PS2 Data, kao i ulaz za clock same pločice. Pošto data komponenta funckionise kao mašina stanja, potreban je i ulaz reset kojim se postavlja u početni stanje. Da bi otpočeo rad ovog modula (PS2\_Controllera) potrebno je da taj signal (reset) bude aktivan samo tokom jednog takta clock-a. Mehanizam koji to omogućava prikazan je na slici 2.



*Slika 2: Mehanizam za pokretanje PS2 Controllera*

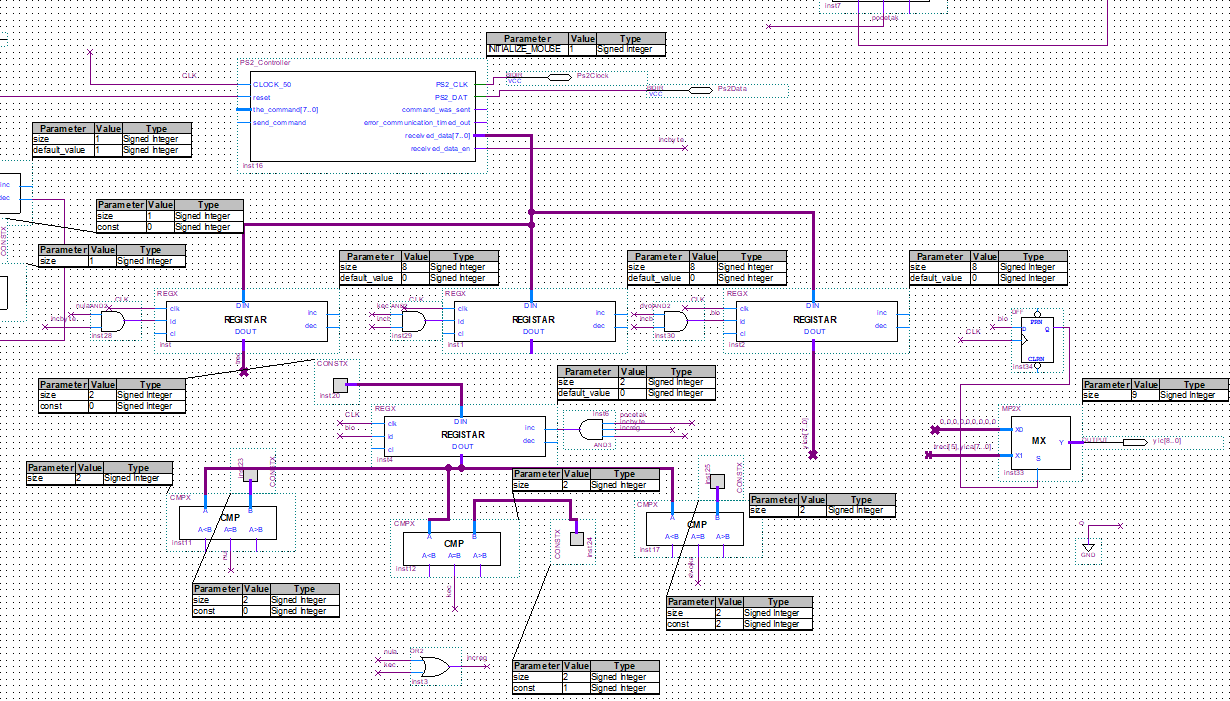
Imajući u vidu da nam je funkcionalnost miša potrebna od samog pokretanja celokupne igrice, omogućili smo da samo u prvom signal takta signal reset bude 1. Početna vrednost parametrizovanog registra čiji se izlaz dovodi na ulaz reset je 1. Kada naiđe prvi signal takta, njegova vrednost se smanjuje na 0, jer je izlaz iz komparatora aktivan (sadržaj registra je veći od 0). Međutim, pre tog smanjenja za koje je potrebno da prođe jedan ceo signal takta, aktiviraće se reset koji je direktno povezan na sadržaj registra koji je u tom trenutku 1. Kako je od trenutka smanjenja pa nadalje, vrednost registra konstantno 0, nijednom se neće aktivirati signal reset sve do ponovnog pokretanja igrice.

Da bi se omogućila pravila komunikacija sa mišem, prva dva bajta koja predstavljaju kontrolne bajtove koje šalje miš treba preskočiti. Tek nakon toga bajtove treba tumačiti kao pomeraj miša i statusni bajt. Mehanizam kojim se to postiže je dat na slici 3.



*Slika 3: Mehanizam za preskakanje prva dva bajta PS2 Controllera*

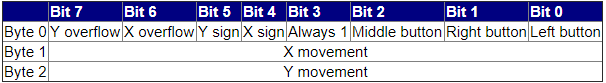
Koristi se slična ideja kao malopre. Potreban je registar kojim se broji koliko puta je bio aktivan signal incbyte (broj poslatih bajtova iz kontrolera). Kada on postane 2, aktivira se signal pocetak kojim se otpočinje faza za upis u prihvatne registre za bajtove miša. Prihvatne registre miša čine tri osmobajtna registra: X, Y i “treci”. Njihov raspored dat je na slici 4.



*Slika 4: Prihvatni registri*

Registar “treci” je za statusni bajt, drugi za pomeraj po X osi, a treći za pomeraj po Y osi. Ucitavanje u date registre se obavlja ukoliko je aktivan gore objašnjeni signal pocetak, kao i odgovarajući signal brojačkog registra (slično realizaciji upravljačke jedinice pomoću brojača koraka). Brojački registar ima 3 stannja: nula, kec i dvojka. Nula je aktivna na početku. Tada se vrši učitavanje statusnog bajta. Zatim stanje prelazi u keca kada se aktivira učitavanje bajta za X pomeraj, a zatim u dvojku kada se učitava Y. Treba istaći signal increg koji je aktivan samo kada je registar stanja u nuli ili kecu kako bi se izbeglo istovremeno clear-ovanje i increment-iranje registra. Registar se vraća na nulu kada se pojavi novi bajt, a trenutnjo stanje je dvojka.

Za našu igru, potreban nam je samo Y pomeraj miša, tako da se samo bus za Y pomeraj šalje iz ove komponente. To se čini preko multipleksera, koji propušta nule sve dok se ne učita novi bajt u registar za Y. To znači da je ceo novi paket od tri bajta poslat. Učitavanje u registar za Y se registruje signalom bio koji se potom dovodi na D flip-flop. To se radi kako bi se ostvarilo kašnjenje od jednog takta potrebno da se vrednost učita u registar. Nakon što je vrednost učitana, ona se propušta kroz multiplekser sve do izlaza, dok na D flip-flop sada dolazi nula i već u sledećem signal takta, propuštaće se nule, tako das mo postigli slanje Y pomeraja koje traje samo jedan signal takta. Pored vrednosti pomeraja miša po Y osi potreban nam je i smer kretanja miša. Smer kretanja miša dat je petim bitom statusnog (prvog bajta). Opšti izgled bajtova koji se prihvataju dat je na slici 5.

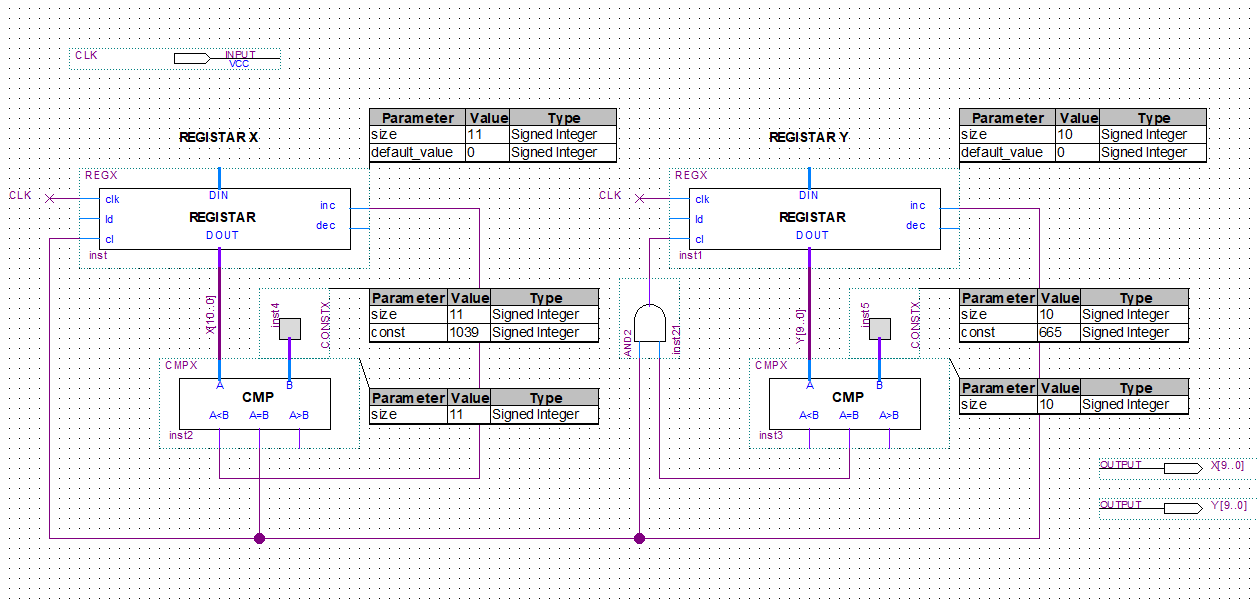


*Slika 5: Bajtovi relativnog pomeraja*

Treba dodatno obratiti pažnju da je relativni pomeraj bilo po X ili Y osi (u našem slučaju) dat kao pozitivan ceo broj ukoliko se mišem krećemo u istom smeru u kom su usmerene ose, a kao negativan ceo broj ukoliko se mišem krećemo u suprotnom smeru od orjentacije pomenutih osa.

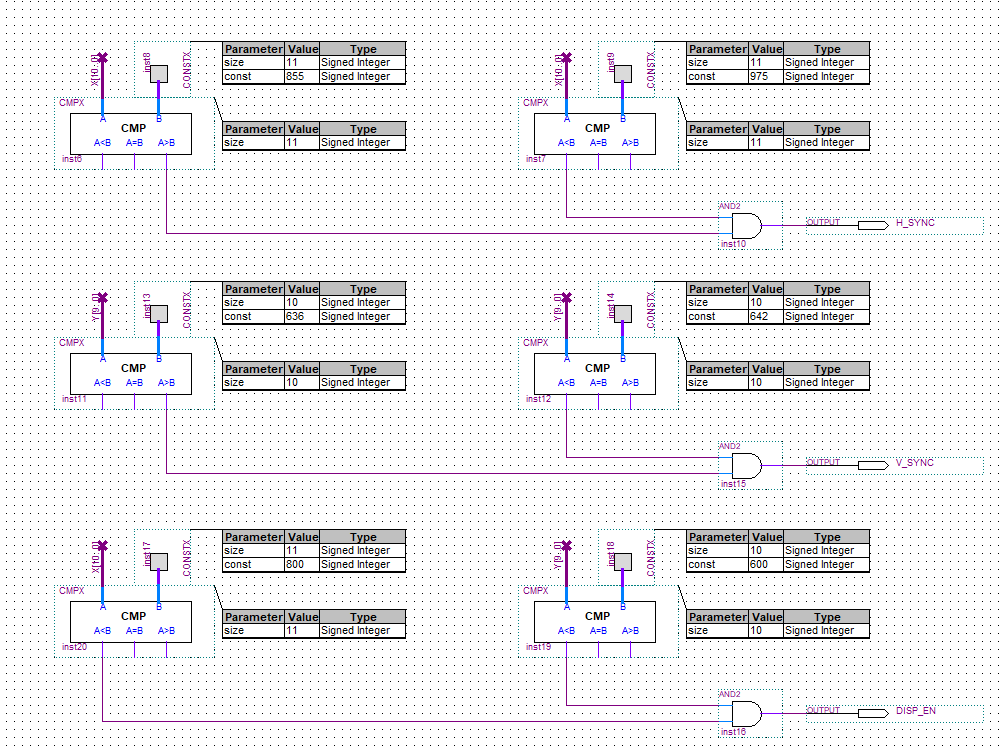
**VGA1**

Za uspešno generisanje koordinata piksela i ostalih statusnih signala neophodnih za uspostavljanje VGA protokola, od ulaznih podataka nam je potreban samo clock. On je zaslužan za funkcionisanje registara u kojima se čuvaju koordinate. Registar X se inkrementira sve dok se ne dođe do 1039 što predstavlja završni piksel ekrana po horizontali. Tada se umesto inkrementiranja radi clear tj. počinje se ponovno generisanje od leve ivice ekrana. Istovremeno se vrši inkrementiranje registra Y što za posledicu ima prelazak u novi red, sve dok se ne dođe do 665. piksela u registru Y kada se umesto inkrementiranja radi clear, jer se sada došlo do kraja ekrana po vertikali. To se može videti na slici pod rednim brojem 6.



*Slika 6*

Pored koordinata X i Y potrebno je izvesti i signale H\_SYNC, V\_SYNC i DISP\_EN. H\_SYNC je aktivan kada se X koordinata nalazi između 855 i 975. Do donje granice 855 se dolazi kao rezolucija displeja po X + front porch (800+56-1=855). Do gornje granice 975 se dolazi kao rezolucija displeja po Y + front porch + sync pulse (800+56+120-1=975). Analogno se dolazi do izraza za Y\_SYNC, samo što važi: rezolucija displeja po Y = 600, front porch = 37, sync pulse = 6. Signal DISP\_EN je aktivan kada se piksel nalazi u okviru rezolucije displeja (X koordinata je manja od 800, a Y koordinata je manja od 600). Šema kojom se određuju dati signali je prikazana na slici 7.



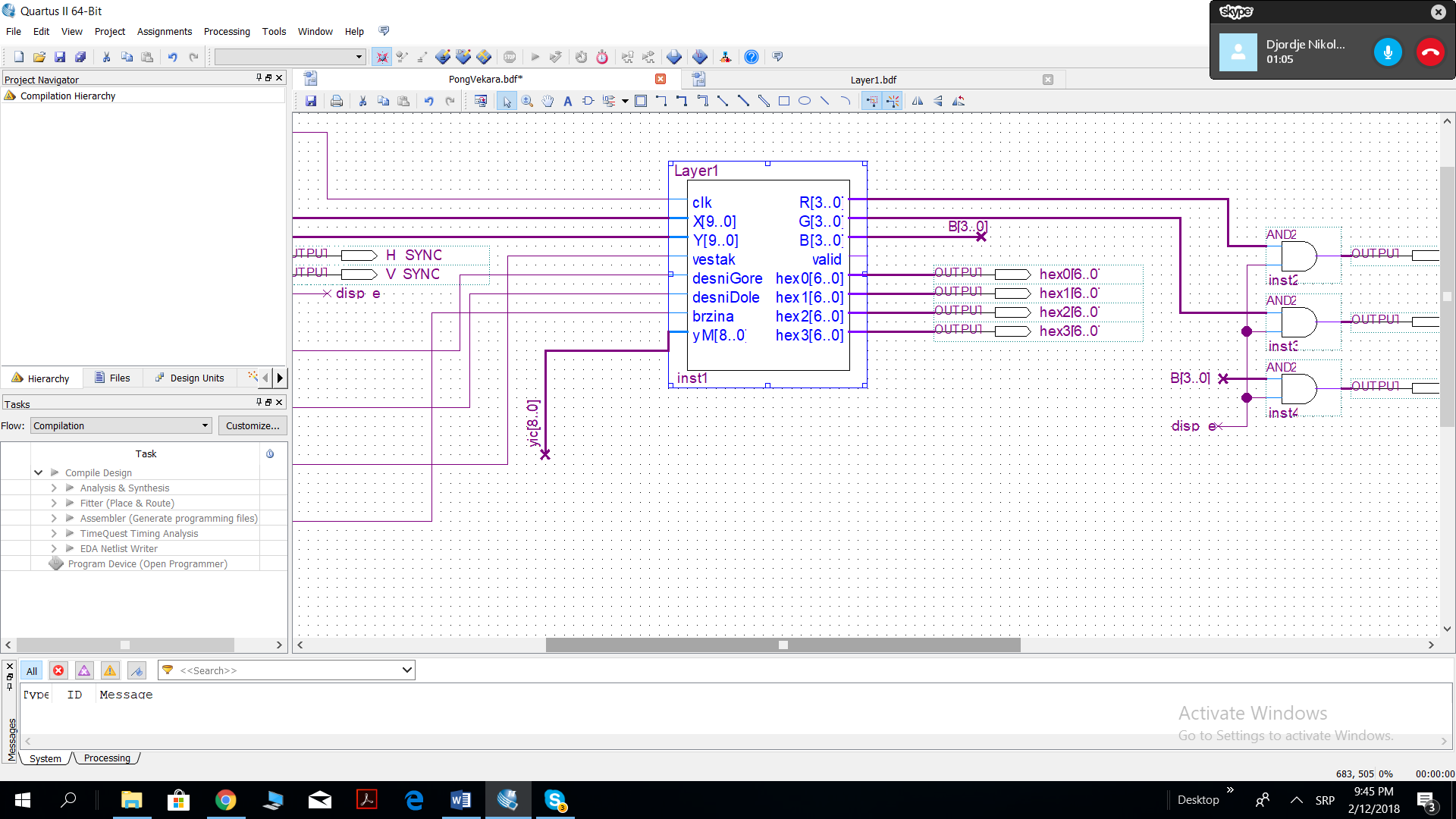
*Slika 7*

**LAYER1**

Modul pod nazivom Layer1 predstavlja ključnu komponentu koja omogućava formiranje svih objekata koji se nalaze na ekranu. U njemu se nalazi najveća količina kombinacionih i sekvencijalnih mreža našeg projekta i on pored boja i rasporeda iscrtanih oblika (palica, loptice i mreže) sadrži i ostale mnogo kompleksnije komponente koje nam npr. govore u kojem stanju se nalazi loptica.

Od ulaza ovaj modul ima naravno klok (ulaz clk), X i Y koordinate piksela koji se iscrtavaju na ekranu, koje potiču iz modula VGA1, ulaz „vestak“ koji nam govori da li je uključena veštačka inteligencija za desnu palicu ili će ona biti upravljana putem buttona1 i buttona2 (ti buttoni se ukoliko je potrebno koriste preko ulaza desniGore i desniDole Layera). Takođe pored tih ulaza postoji i ulaz „brzina“ na koji se dovodi svič SW8 i koji nam omogućava podešavanje težine bota koji će igrati ukoliko je uključena veštačka inteligencija i ulaz yM koji predstavlja koordinatu miša po y osi (ona je potrebna za pokretanje leve palice).

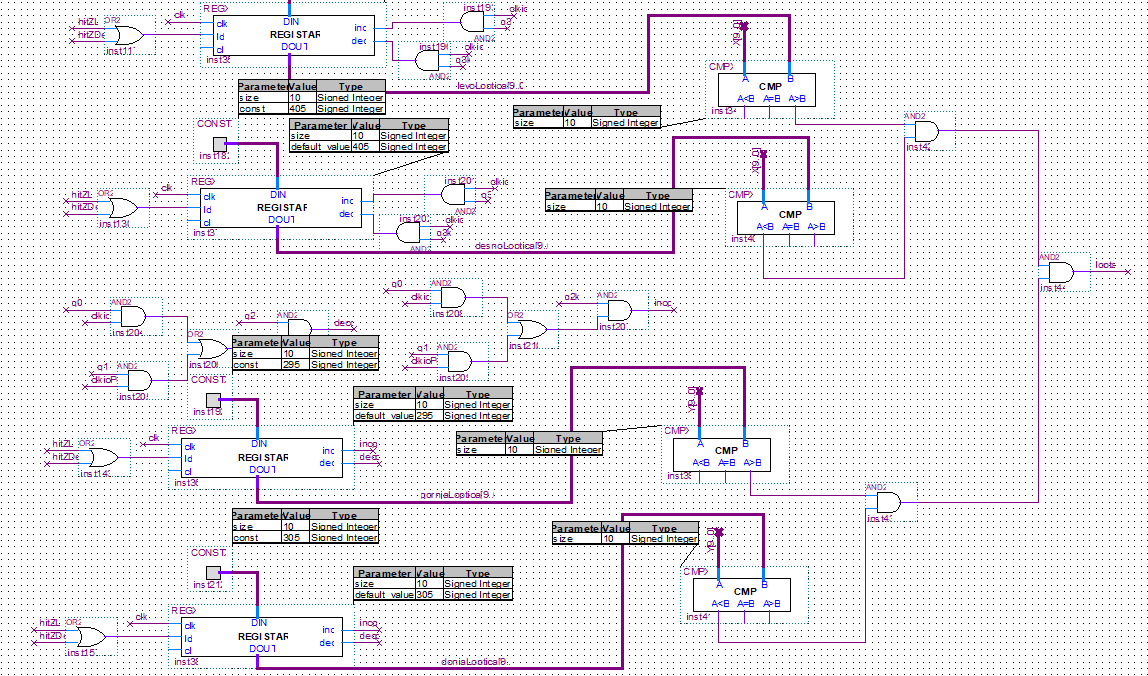
Na izlaze ovaj modul propušta rezultat za sedmosegmentni displej (izlazi hex0,hex1,hex2 i hex3) i boju piksela u formatu RGB (za svaku boju po jedan izlaz). Postoji i bit valid koji mi nismo koristili, čija je funkcija objašnjena u uvodnom delu. Spoljašnji prikaz modula dat je na slici 8.



*Slika 8: Spoljašnji prikaz modula Layer1*

Prva velika celina koja se nalazi u modulu Layer1 predstavlja skup registara koji čuvaju granične vrednosti leve palice, desne palice i loptice.

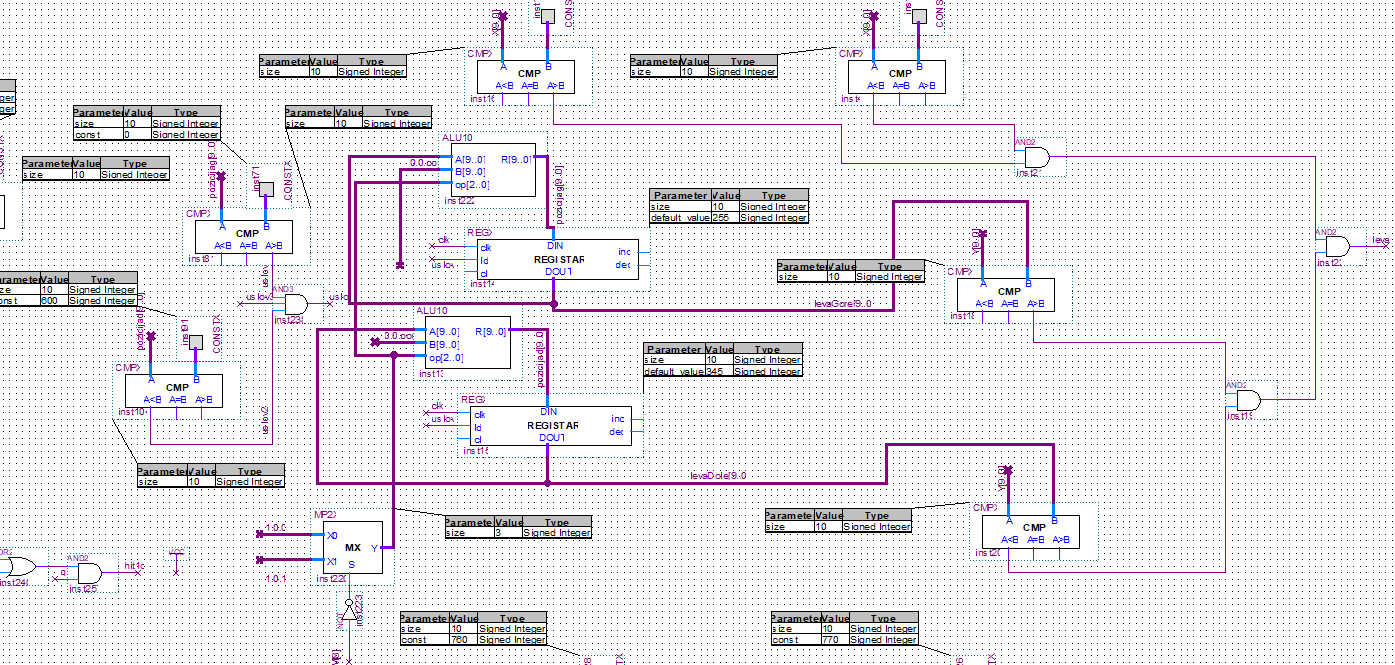
Za lopticu koja je veličine 10x10 piksela koriste se desetobitni registri (iz razloga što se za mapiranje na ekranu koristi 10 bita) levoLoptica, desnoLoptica, donjaLoptica i gornjaLoptica. Ovi registri kao što je već naglašeno iznad čuvaju levu,desnu,donju i gornju granicu loptice. Između leve i desne i gornje i donje granice loptica će morati biti obojena drugom bojom (u odnosu na pozadinu). Tim granicama moraće biti omogućeno pomeranje u zavisnosti od pravca i smera kretanja loptice. Za početne vrednosti registara uzeli smo vrednosti takve da loptica na početku svake partije polazi iz sredine. Te vrednosti loadovaće se u registre svaki put kada dođe do udarca u levi ili desni zid što je realizovano ili kolom na ulazu ld tih registara. Na inkrement i dekrement će se dovoditi signali q1…q3 u zavisnosti od toga u kojem se stanju loptica nalazi (stanja će biti objašnjena kasnije u tekstu) kao i signal „clkic” koji označava klok određene brzine.(usporen pomoću komponente CLK DEVIDER koja menja frekvenciju kloka). Vrednosti tih registara se dovode na komparatore gde se upoređuju sa vrednošću X i Y koordinate piksela koji se trenutno iscrtava. Ako se piksel nalazi u tim granicama biće iscrtan jednom bojom (bojom loptice) a ako ne, biće iscrtan drugom bojom (bojom pozadine). Registri loptice prikazani su na slici 9.



*Slika 9: Registri loptice*

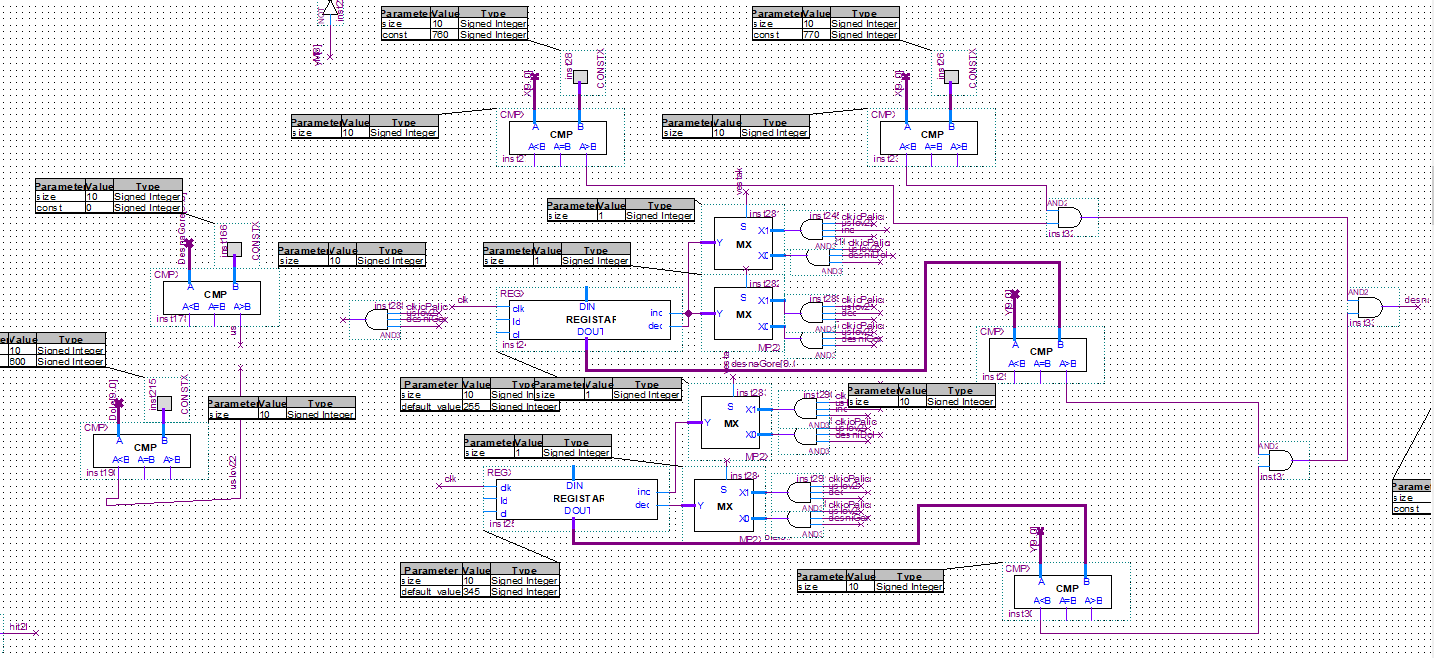
Za registre leve i desne palice važi praktično ista stvar kao za registre loptice osim što su palice veličine 90x10, ne čuvaju se leva i desna granica palice jer su one konstantne (palice se ne kreću po x osi) i što se leva palica kreće pomoću miša gde dolazi do dodatnih komplikacija.

Za gornju i donju ograničavajuću ivicu leve palice koriste se desetobitni registri levaGore i levaDole, dok se za levu i desnu ograničavajuću ivicu koriste samo konstante 30 i 40 respektivno. Ti registri i konstante se na isti način kao i kod loptice upoređuju sa vrednošću X i Y piksela koji se trenutno iscrtava i u zavisnosti od toga da li se piksel nalazi u pomenutim opsezima (iznad donje, ispod gornje, levo od desne i desno od leve granice) se pali signal leva, koji će se kasnije koristiti da se koordiniše boja kojom će taj piksel biti obojen. Vrednosti regisara koji predstavljaju donju i gornju granicu dobijaju se tako što se u svakom signalu takta prethodna vrednost registra sabira sa relativnim pomerajem iz modula MIŠ2 koji je dobijen kao posledica fizičkog pomeranja miša. Za operaciju na aritmetičko logičkim jedinicama odabira se sabiranje ili oduzimanje u zavisnosti od smera kretanja miša (to nam govori bit yM[8] ). Prikaz registara za pokretanje leve palice dat je na slici 10.



*Slika 10: Registri za granice leve palice*

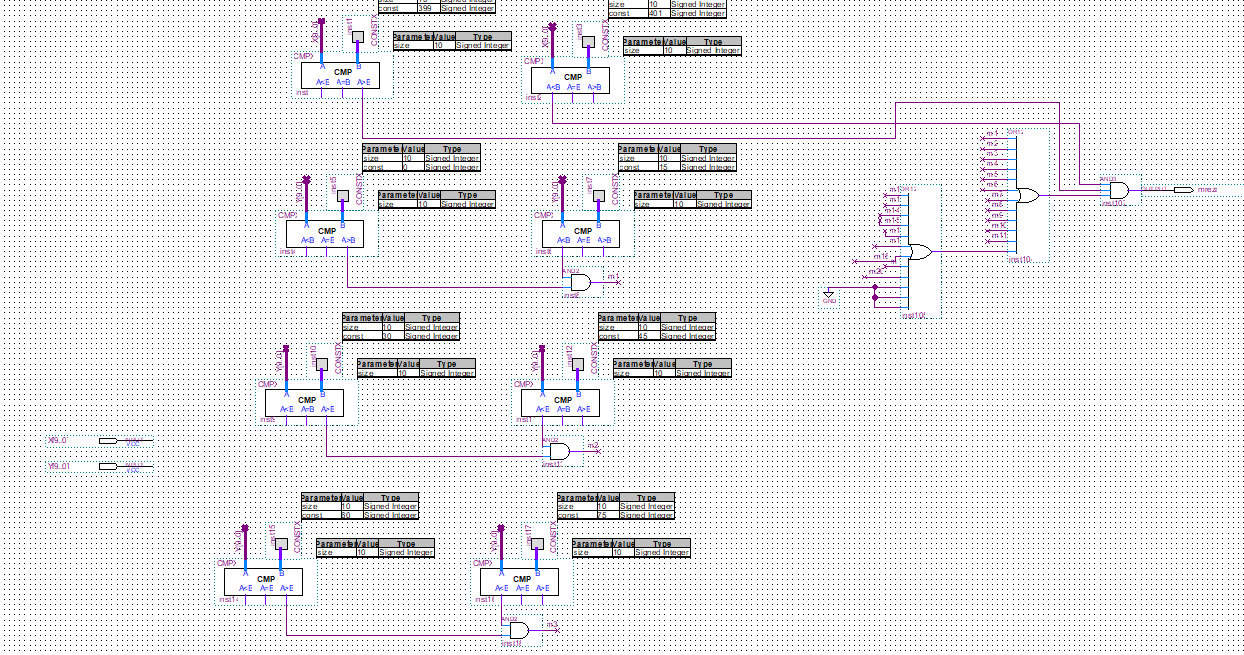
Za ograničavajuće desne palice koriste se registri desnaGore i desnaDole kao i konstante 760 i 770 (ponvo iz istog razloga zato što se desna palica ne kreće po x osi). Ove ivice koriste se na isti način kao i kod leve palice, samo se signal koji se aktivira naziva desna. Razlika između leve i desne palice je u tome što se desna palica ne kreće pomoću miša i pomoću sukcesivne operacije load, već se prilikom svakog takta u zavisnosti od pritisnutog buttona (kada igra drugi igrač) ili smera u kome se kreće palica izračunatom od strane veštačke intaligencije (kada ne igra drugi igrač) inkrementira ili dekrementira. Multiplekseri na ulazu inkrementa i dekrementa postoje iz razloga što postoje dve različite brzine kojima se desna palica može kretati, pa se u zavisnosti od toga dovode dva različita kloka. Prikaz registara za pokretanje desne palice dat je na slici 11.



*Slika 11: Registri za granice desne palice*

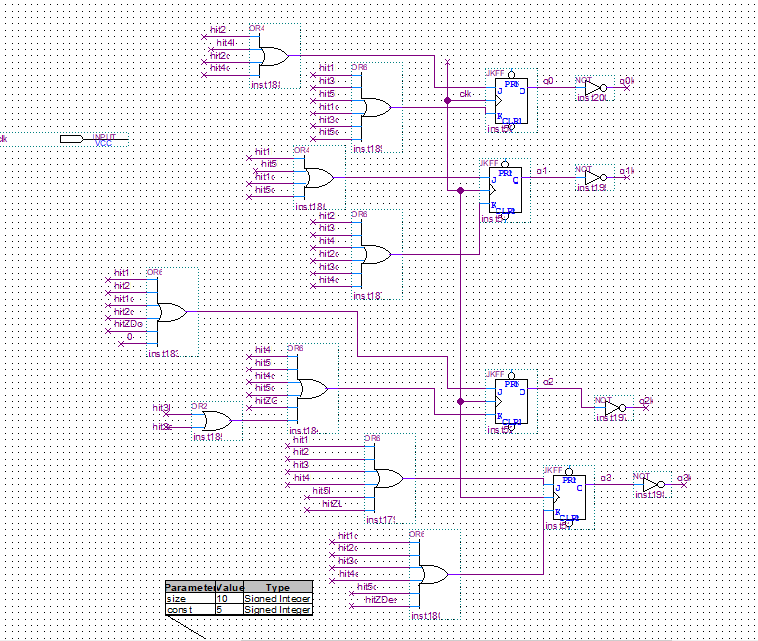
Ono što je zajedničko za obe palice jeste postojanje signala uslov koji se dovodi na “i” kolo sa svim operacijama koje se vrše nad palicama. Taj signal potoji da ukoliko palica ode van gornje ili donje granice ekrana onemogući njeno dodatno pomeranje izvan tih granica.Preciznije, igrač neće moći da probije gornju ili donju ivicu ekrana palicama za igru.

Pored ovih pokretnih objekata postoji i modul “Mreza“ koji se sastoji od mnogo komparatora i konstanti koje se na isti način kao kod palica upoređuju sa X i Y koordinatom piksela koji se iscrtava i u zavisnosti od toga aktiviraju signal mreža koji će kasnije koordinisati da se na tim mestima iscrta boja koja odgovara boji mreže. Modul mreža prikazan je na slici 12.



*Slika 12: Modul Mreža*

Da bismo objasnili moguća stanja loptice, potrebno je bilo da uočimo sva moguća kretanja loptice. U zavisnosti kretanja loptice levo ili desno aktivirali smo signal q3(1 ako ide desno, 0 ako ide levo) . U zavisnosti kretanja loptice gore ili dole aktivirali smo signal q2(1 ako ide dole,0 ako ide gore). U zavisnosti kretanja loptice pod 30 stepeni aktivirali smo signal q1(1 ako ide pod 30 stepeni, 0 ako ne ide). U zavisnosti kretanja loptice pod 45 stepeni aktivirali smo signal q0(1 ako ide pod 45 stepeni, 0 ako ne ide). Vrednosti ova 4 signala dovodili smo na ulaze inc i dec loptice koji su navedeni ranije u tekstu. Na taj način postizali smo raznolikost kretanja loptice. Svi ovi signali dobijaju se na izlazima JK flip-flopova prikazanih na slici 13.

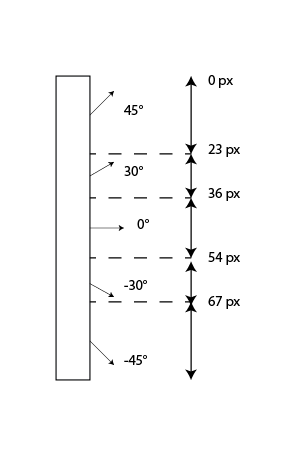


*Slika 13: Dobijanje signala stanja q1…q3*

Na ulaze JK flip-flopova dovedena su “ili” kola na čije ulaze su prikačeni signali hit1d…hit5d, hit1l…hit5l, hitZD, hitZG, hitZL i hitZDes koji označavaju različite sudare loptice i okoline (leve i desne palice i gornjeg,donjeg,levog i desnog zida). Dakle, u zavisnosti od vrste udarca doći će do eventualne promene bitova stanja q1…q3.

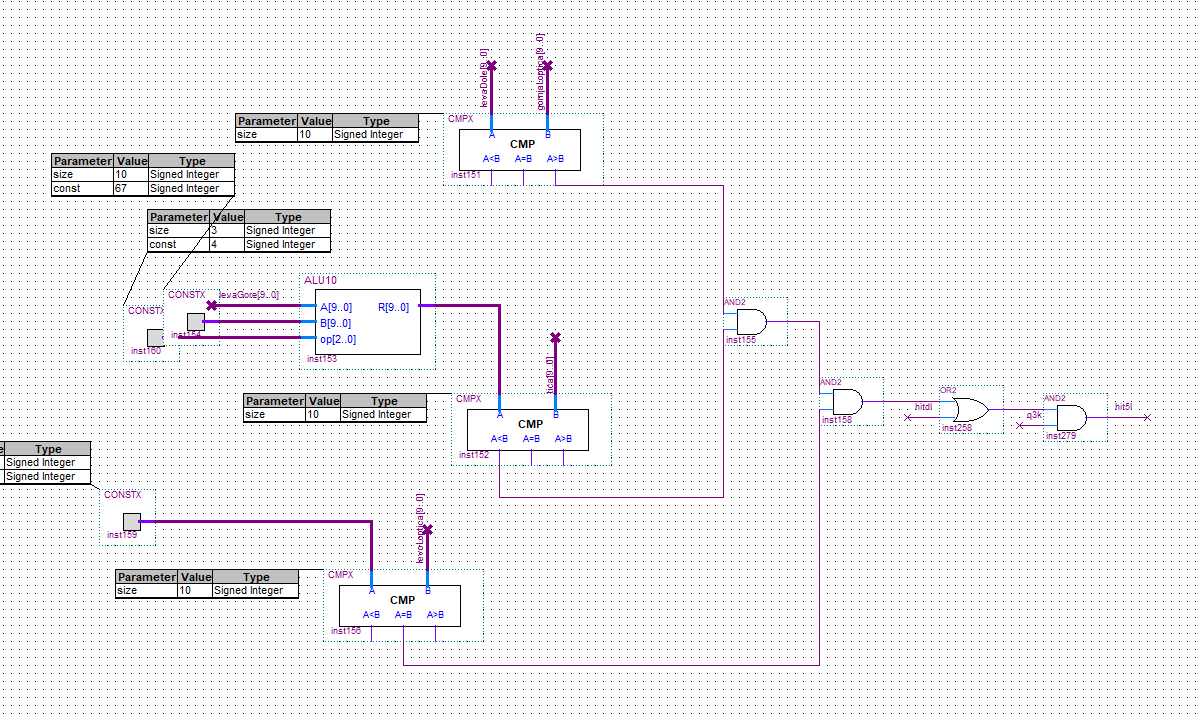
**hit-ovi:**

Radi lakše promene stanja q0…q3 koja određuju smer i pravac kretanja loptice, koristimo po 5 signala hit koji se pale u zavisnosti od toga kojim delom palice se udari loptica. Konkretno, palica je podeljena na 5 imaginarnih delova. Centralni deo odbija lopticu pod uglom 0°, a takođe su prisutna po dva segmenta sa obe strane palice koja simetrično u odnosu na X osu odbijaju lopticu pod uglovima od 30°, odnosno 45°. Detaljan segmentni prikaz palice je dat na slici 14.



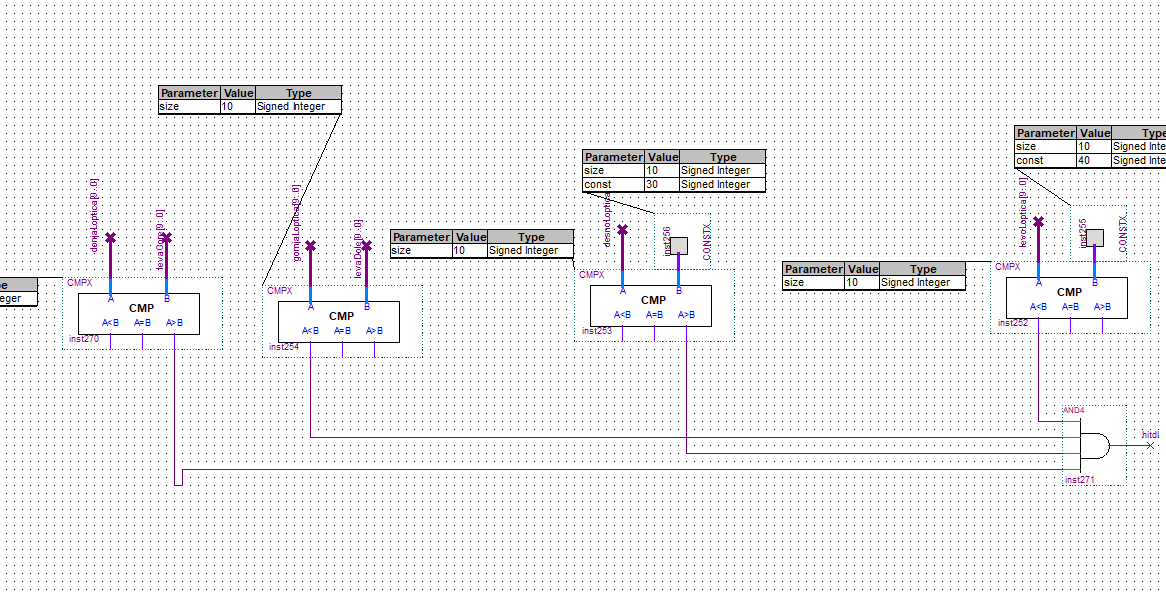
*Slika 14*

Svi hitovi se generišu na sličan način. Jedan od njih će biti objašnjen po ugledu na sledeću šemu.



Slika 15: Šema hitova

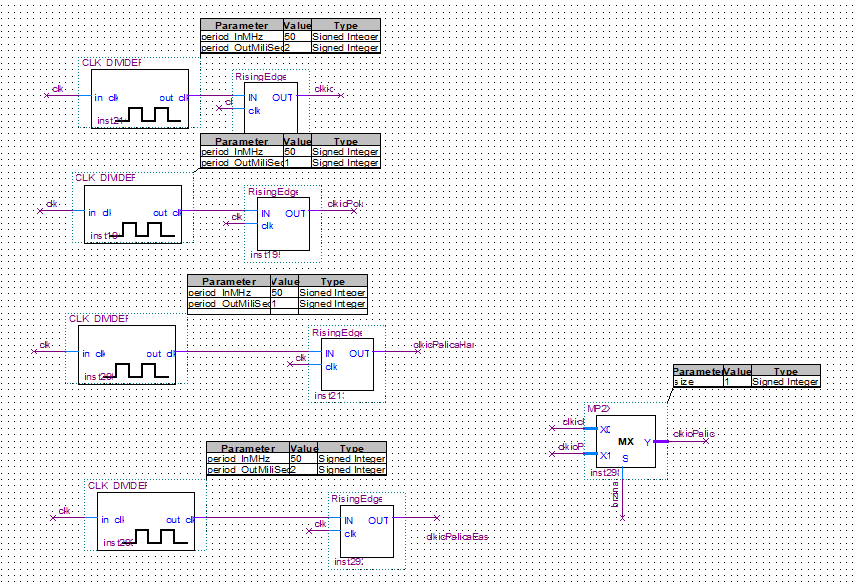
Ovde možemo videti da se gornja ivica loptice upoređuje prvo sa donjim delom palice, kao i zbir gornje ivice palice i 67 sa gornjom ivicom loptice. Da bi hit bio aktivan potrebno je da vrednost gornje ivice loptice bude između te dve vrednosti. Pored toga, moramo obezbediti da leva ivica loptice dodirne desnu ivicu palice po X osi. Taj posao vrši poslednji komparator. Taj signal se zatim dovodi na ili kolo sa signalom hitdl koji omogućava odbijanje i ukoliko se loptica udari samom gornjom ili donjom ivicom palice. Ovaj dodatak je karakterističan samo za hit1 i hit5, jer je odlučeno da se tada loptica odbija pod apsolutnim uglom od 45°. On se može videti na sledećoj slici 16.



*Slika 16*

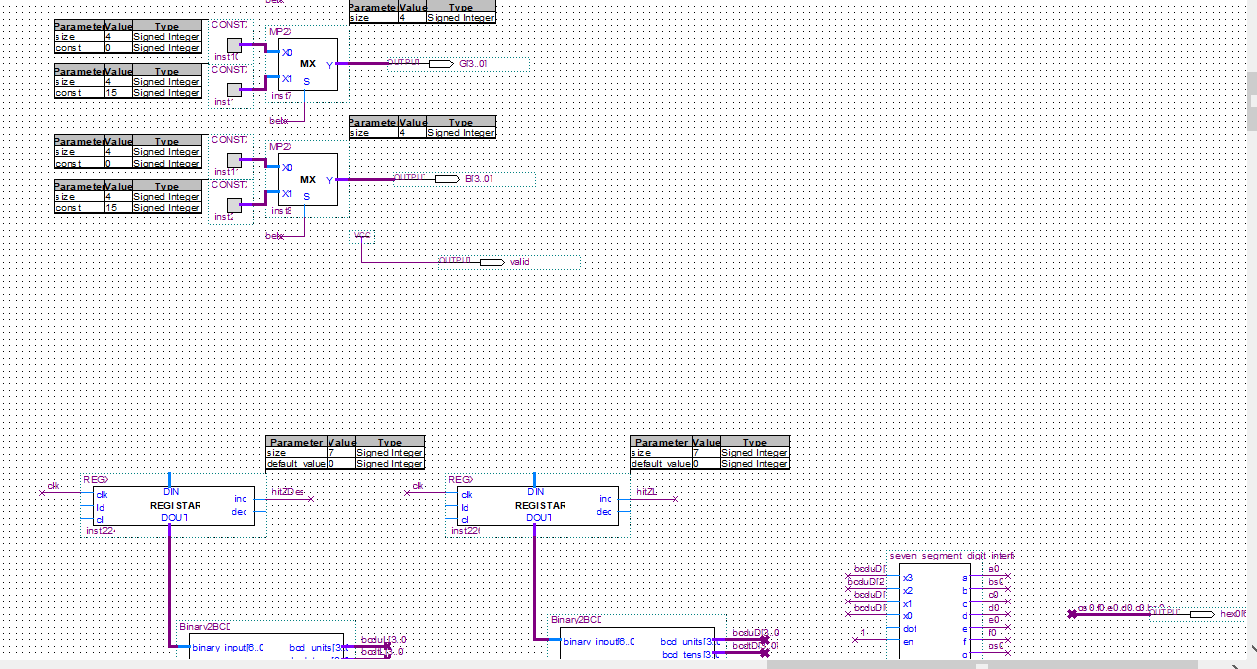
Da ne bi došlo do nekakvih nepredviđenih situacija i bagova, postarali smo se id a je do hita moguće doći samo ukoliko loptica ide u smeru ka palici od koje se odbija tj. za leve hitove je q3 jednako 0, a za desne hitove je q3 jednako 1. Za to je zaslužno poslednje i kolo.

Brzina klokova podešavana je u zavisnosti od toga u kom pravcu loptica treba da se kreće. Na primer, ako loptica treba da ide pod 45 stepeni, podjednakom brzinom će se inkrementirati X i Y koordinata. Ako se loptica odbije pod uglom od 30 stepeni, klok za kretanje po jednoj osi će biti brži od kloka kretanja po drugoj osi i na taj način će se dobiti vizualni efekat kretanja lotpice pod datim uglom. Svaki CLK DEVIDER mora biti ispraćen i RISING EDG-om da bismo dobili signal aktivacije pri svakoj ulaznoj ivici usporenog kloka. Različite brzine klokova date su na slici 17.



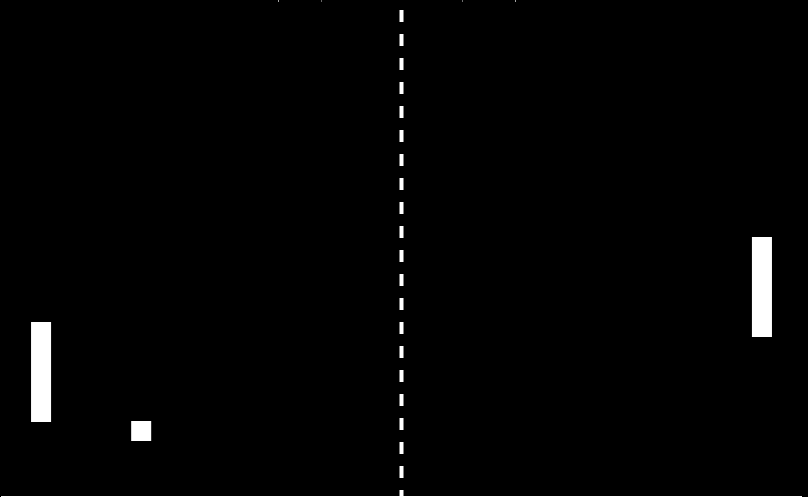
S*lika 17: CLK DIVIDERI*

Na samom kraju na izlaze smo doveli signale koji prolaze kroz dvoulazne multipleksere. Selekcioni pin ovih multipleksera su signali mreza,leva,desna i loptica spojeni pomoću ili kola u jedan signal pod nazivom “bela”. U prevodu ako je taj signal aktivan piksel će biti ofarban belom bojom, a ukoliko nije biće ofarban crnom bojom. Za boje palica loptice i mreže uzeta je bela boja a za pozadinu crna boja po uzoru na originalnu igru Pong. Pored ovih izlaza postoje i izlazi za rezultat na sedmosegmentnom displeju. Izlazi za boje i sedmosegmentni displej prikazani su na slici 18.



*Slika 18: Izlazi*

Analizom glavnih komponenti i celokupne glavne šeme, može se I shvatiti kako funkcioniše celokupna igra pong čiji je finalni izgled prikazan na slici 19.



*Slika 19: Igra Pong*

*Na projektu radili:*

*Đorđe Nikolašević 0074/2016*

*Marko Vekarić 0089/2016*