طراحی سیستم‌های دیجیتال
Digital System Design
دانشکده مهندسی کامپیوتر
مدرس: دکتر گودرزی

دانشگاه صنعتی شریف

نام و نام خانوادگی: نگار رحمتی، نیلوفر صالحی
شماره دانشجویی: 88102026، 88110664

پروژه بازی تتریس

پیاده‌سازی بازی تتریس روی برد FPGA

• شرح بازی

بازی تتریس را همه می‌شناسند. در این بازی هفت شکل وجود دارد که به صورت رندم درصفحه ظاهر می‌شوند و روی شکل‌های قبلی قرار می‌گیرند. هرگاه یک ردیف کامل پر شود، آن ردیف پاک می‌شود و بازی تا وقتی که شکل‌ها تا بالای صفحه روی هم قرار بگیرند ادامه پیدا می‌کند.

در این فاز اول پروژه ما با استفاده از برد DE2 بخشی از این بازی را طراحی کردیم که شامل موارد زیر بود:

- در نظر گرفتن بخشی از صفحه نمایش برای انجام بازی. شکل‌ها به صورت رندم در بالای این صفحه ظاهر می‌شوند، به سمت پایین حرکت می‌کنند و از آن خارج می‌شوند.
- در نظر گرفتن کلیدهایی برای حرکت دادن شکل‌ها به چپ، راست و چرخاندن آن و کلیدی برای افزایش سرعت حرکت شکل به سمت پایین.

-در نظر گرفتن بخشی از صفحه برای نمایش شکل بعدی که قرار است ظاهر شود.

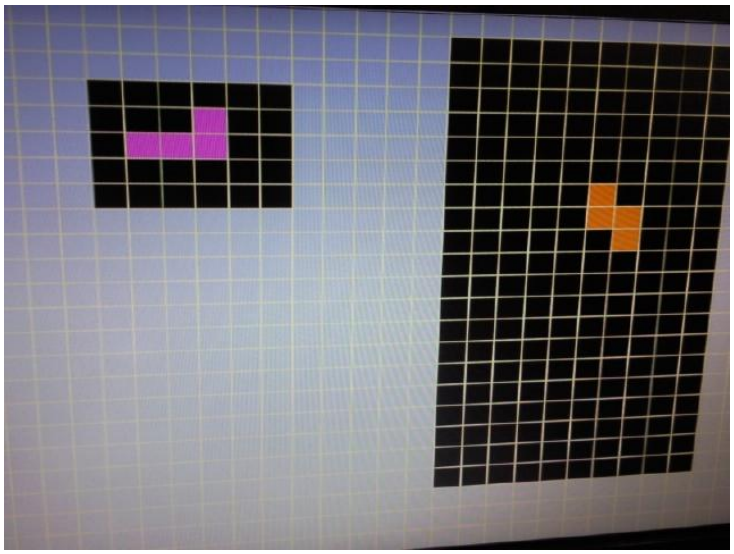
کلیدهای استفاده شده در بازی:

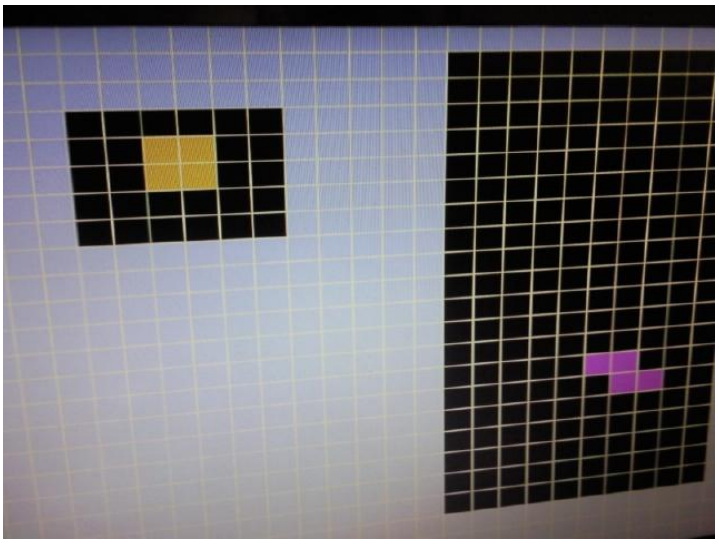
- right key → حرکت شکل به راست
- left key → حرکت شکل به چپ
- up key → چرخاندن شکل
- down key →افزایش سرعت حرکت شکل به پایین
- r → reset
- p →pause
- s →start

توجه!

SW[0] روی برد باید 1 باشد تا بازی شروع شود. بعد از فشردن pause بازی با s شروع می-شود.

شکل بازی در این فاز به صورت زیر است:





- **جزئیات پیاده‌سازی**

برای پیاده‌سازی بازی ابتدا به کار با برد مسلط شدیم. ماژول‌های DE2_default, VGA و key board را از سایت DE2 board برداشتیم .
برای نمایش، از VGA در DE2_default یک نمونه گرفتیم و ورودی‌های مربوطه را طوری تعیین کردیم که در هر مرحله شکل مورد نظر را در صفحه بکشد. مختصات گوشه بالا سمت چپ شکل را با دو متغییر X و Y در نظر گرفتیم. به طوری که Xبا فشردن کلید های چپ و راست تغییر کند و Y با کلاک به صورت مرتب افزایش یابد.برای مثال:

```
always @(posedge MYCLOCK[21])
begin
    if(SW[0] || Y==20)
    begin
        X = 500;
    end
```

```
        else if(keyOut == 'b01101011 && 400<X)//left
            X = X-20;
        else if(keyOut == 'b01110100 )begin //right
            if ( rightX<600 )
                X = X+20;
        end
    end
```

برای کار با key board ابتدا خروجی‌های آن را روی برد نمایش دادیم تا کد ASCII مربوط به هر کلید را بفهمیم و سپس یک خروجی از ماژول key board گرفتیم که مشخص می‌کرد کاربر چه کلیدی را فشرده است.

برای رسم شکل ها مقادیری را که با توجه به X و Y محاسبه می‌شد با آرایه ای از سیم ها assign کردیم. برای هر شکل اگر coord_X و coord_Y که VGA می‌فرستد داخل محدوده شکل باشد مقدار سیم مربوط به آن شکل یک می‌شود و آن نقطه به رنگ شکلی که ما می خواهیم در می آید. برای مثال:

```
wire [3:0]shape [6:0];
assign shape[0][0]=(Coord_X<X+40 && Coord_X>X &&
Coord_Y>Y && Coord_Y<Y+40);
```

برای کشیدن شکل بعدی در مربع جداگانه هم همین کار را کردیم.

برای این که شکل از صفحه بیرون نزند با توجه به X و Y ، مختصات چپ و راست و پایین اشکال را ذخیره کردیم.

برای این که شکل به راحتی به چپ و راست منتقل شود کلاک بازی را به یک کانتر دادیم و کلاک بخشی را که مربوط به تاثیر کلیدها بود از کلاک مربوط به پایین آمدن شکل سریع‌تر در نظر گرفتیم.
برای نمایش رندم اشکال خودمان عدد رندم تولید کردیم.

- **ویژگی‌ها و ایده‌ها**

-برای بازی یک کلید هم برای سریع پایین آمدن در نظر گرفتیم که بازی را جذابتر کند.
-در مواردی از دستورات data flow مانند assign استفاده کردیم که سرعت کامپایل را بسیار افزایش داد.
-با استفاده از کلاک‌های مختلف حرکت شی را آسان و ران کردیم. به طوری که شکل به دلخواه کاربر و با توجه به نحوه فشردن کلید ها تند یا کند حرکت کند.
-چون $random سنتز نمی‌شد خودمان عدد رندم تولید کردیم.

- **شرح وظایف**

برای پیاده سازی بازی هر دوی ما در هر زمان ممکن در آزمایشگاه حاضر می‌شدیم تا بخش‌هایی از بازی را پیاده‌سازی کنیم و بلافاصله روی برد

امتحان کنیم. منطق کلی بازی را به کمک هم طراحی کردیم. برای بخش‌های دیگر تقسیم وظایف در مجموع به صورت زیر بود:

-نیلوفر صالحی: راه اندازی نمایش گر و key board، مشخص کردن رنگ‌ها، تنظیم کلاک ها، بیرون نزدن شکل از صفحه، چپ و راست شدن شکل.
-نگار رحمتی: چرخیدن اشکال ، راه اندازی reset, pauseو start ، نمایش شکل بعدی در قسمت مربوطه، پایین آمدن شکل.

- **فاز دوم**

برای فاز دوم که هم اکنون نیز قسمتی از آن را پیاده سازی کرده ایم، در نظر داریم تا ماتریسی متناظر با کل مربع‌های درون صفحه بازی دست کنیم و رنگ هر خانه را با کدی که به هر رنگ map می-کنیم ذخیره کنیم. سپس خانه های این ماتریس را به ورودی sram که مستقیم به ورودی vga وصل است، می دهیم.
برای هر شکل یک عدد 16 بیتی در نظر می‌گیریم که مشخص می‌کن این شکل کدام خانه‌ها را با توجه به Xو Y رنگی می کند .باتوجه به این عدد هر جا که لازم بود شکل ثابت شود آن را در ماتریس ذخیره می‌کنیم.
امتیاز و زمان بازی را با ایده seven segment در صفحه نمایش خواهیم داد.

فعلا به عنوان ایده برای فاز 2 یک مربع در نظر گرفته ایم که رنگ آن مدام تغییر می کند و وقتی به پایین رسید کل سطر به عنوان جایزه پاک می شود.