



- (1) پروژه ها برای گروه دو نفره (ارشدها بصورت تک نفره) است. اعضای گروه و پروژه انتخابی را تا تاریخ **جمعه 13 اردیبهشت** ایمیل کنید.
- (2) پروژه باید توسط نرم افزار *Altera Quartus* (برای بورد *Altera DE0-CV*) یا *Xilinx ISE* (برای بورد *Spartan6*) پیاده سازی شود.
- (3) کل پوشه های مربوط به پیاده سازی و گزارش را در یک فایل *zip* شده به نام "*CADProject_Name1_Name2*" قرار دهید.
- (4) قبل از زمان **تحويل حضوری**، فقط یک فایل *zip* شده به آدرس زیر ارسال کنید:
<https://www.dropbox.com/request/3knDRN3iOOxdTWABo7Nh>
- (5) زمان **تحويل فاز اول** پروژه، بصورت غیرحضوری، تا تاریخ **چهارشنبه 25 اردیبهشت** خواهد بود.
- (6) زمان **تحويل** نهایی پروژه، **بصورت حضوری**، در تاریخ احتمالی **چهارشنبه 8 خرداد** خواهد بود (تاریخ دقیق متعاقبا اعلام خواهد شد).
- (7) **توجه مهم:** تمام اعضای گروه باید به تمام بخش های پروژه مسلط باشند و ندانستن بخشی به بهانه تقسیم کار پذیرفته نیست.

در این پروژه می توانید از فایل *CAD_VGA_Quartus* (برای *Quartus*) یا *CAD972VGA_Xilinx* (برای *Xilinx*) که روی دراپ باکس قرار داده شده است کمک بگیرید. قالب اصلی این فایل را تغییر ندهید تا بتوانید Pin-Assignment مربوط به بوردها را استفاده نمایید. در این فایل، کنترلر VGA پیاده سازی شده است تا بتوان داده را از طریق پورت VGA روی مانیتور نشان داد. همچنین، برای مثال اولیه، یک مربع متحرک نیز روی مانیتور نمایش می دهد که هرگاه به انتهای صفحه برسد باز می گردد. چند خروجی نمونه نیز روی 7Segment ها نمایش داده شده است. ابتدا کد قرار داده شده را بصورت کامل نگاه کنید و کاملاً رفتار کد را تحلیل کنید. سپس این کد را باید به گونه ای تکمیل کنید که یکی از دو بازی زیر را براساس انتخاب خودتان پیاده سازی کند.

فاز اول: در فاز اول پروژه کفایت صفحه بازی (نمایش پرند و موانع بازی اول و دوم) را آماده کنید.

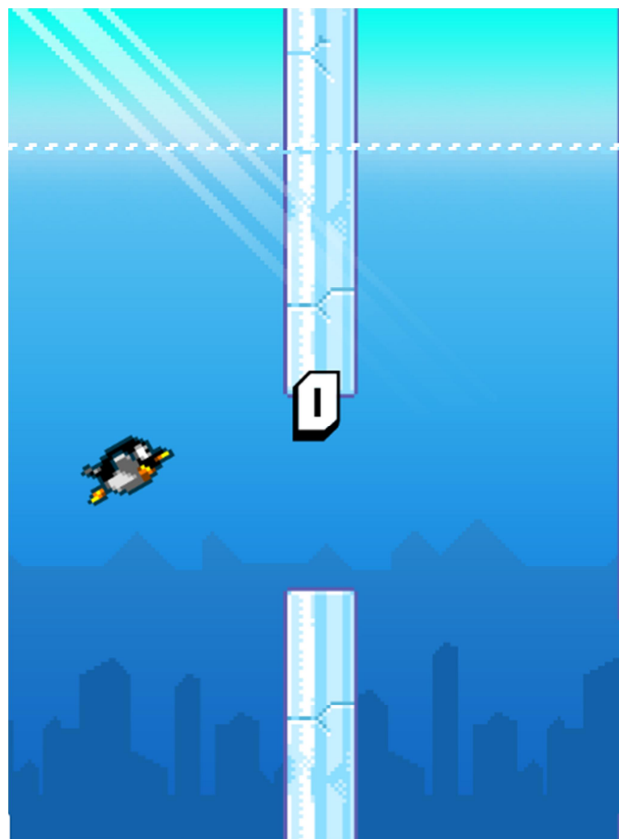
بازی اول: Skipper

هدف از این پروژه پیاده سازی بازی Skipper است. یک پرند در حال حرکت می باشد که با هر بار زدن دکمه به سمت بالا حرکت می کند و اگر دکمه ای زده نشود به صورت پیش فرض با یک سرعت ثابت به سمت پایین حرکت می کند. با این نحوه کنترل، پرند باید از موانع نمایش رد شود.

کد قرار داده شده را باید به گونه ای تکمیل کنید که صفحه نمایش را به یک جدول دو بعدی فرضی تقسیم کند که هر خانه می تواند روشن یا خاموش باشد. بعد از ریست که بازی شروع می شود، یک مربع (جایگزین پرند) در وسط صفحه نمایش دهد و با هر بار فشار دادن Button مربع به سمت بالا حرکت کند و در غیر این صورت به طور پیش فرض به سمت پایین در حرکت باشد. موانع به صورت بلوک هایی که یک قسمت خالی به صورت رندوم دارد از سمت راست تصویر به سمت مربع (پرند) در حرکت می باشد و با کنترل مربع (پرند) باید آن را از قسمت خالی مانع رد داد.



<https://www.gamee.com/game-bot/OmRJ0j4IO>



برای پروژه مراحل زیر را پیاده‌سازی کنید:

- ✓ بعد از زدن کلید Reset، روی نمایشگرهای 7Segment روی برد دو رقم انتهایی شماره دانشجویی اعضای گروه نمایش داده شود، و دو نمایشگر سمت چپ مقدار صفر را نشان دهند. روی مانیتور نیز مربع (پرنده) بصورت غیرمتحرک نمایش داده شود تا یکی از Button ها زده شوند. تمام LEDها نیز خاموش باشند.
- ✓ با زدن یکی از Buttonها بازی آغاز شود و مربع (پرنده) شروع به حرکت کند و بازیکن با فشار دادن یکی از Buttonها پرنده را کنترل کند.
- ✓ دو 7Segment میانی زمان گذشته از بازی را نشان دهند.
- ✓ دو 7Segment امتیاز بازیکن را نشان دهند، بدین صورت که یک مانع به صورت رندوم تولید شود و با رد شدن مربع (پرنده) از قسمت خالی موانع یک امتیاز اضافه گردد.
- ✓ هر 15 ثانیه باید سرعت حرکت موانع را افزایش دهید.
- ✓ شرط انتهایی بازی، رسیدن زمان به 99 ثانیه، یا رسیدن امتیاز یک بازیکن به 10 یا سوختن پرنده است. شرط سوختن پرنده برخورد به موانع می باشد.
- ✓ در این حالت بازی متوقف شود و پرنده حرکت نکند. همچنین تمام LED ها روشن شوند.

بخش اختیاری:

بخش اختیاری – پرنده بصورت خودکار و هوشمند بازی کند.

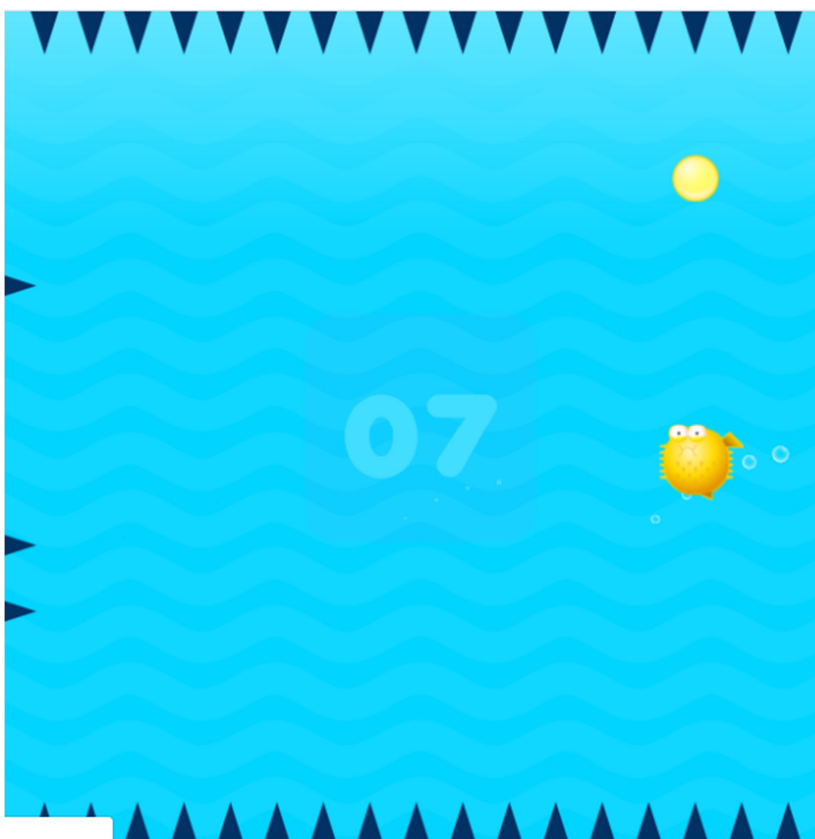


بازی دوم: Spiky fish

هدف از این پروژه پیاده‌سازی بازی SpikyFish است. یک صفحه که دیوارهای بالا و پایین آن موانع به صورت پیش‌فرض وجود دارد و یک پرنده در حال حرکت است. با کنترل پرنده باید از برخورد آن به بالا و پایین و موانع رندوم دو طرف راست و چپ جلوگیری کرد.

پرنده با هر برخورد به دیواره‌های سمت چپ و راست مسیر خود را عوض کرده و در جهت مخالف آن حرکت می‌کند و با فشار دادن هر Button به سمت بالا حرکت کرده و حرکت سمت پایین آن به صورت پیش‌فرض با سرعت ثابت می‌باشد. موانع دیواره‌های سمت چپ و راست نیز با هر برخورد پرنده به دیواره‌های سمت چپ و راست به صورت رندوم تغییر می‌کند.

<https://www.gamee.com/game-bot/YXu4mYBb>



برای پروژه مراحل زیر را پیاده‌سازی کنید:

- ✓ بعد از زدن کلید Reset، روی نمایشگرهای 7Segment روی برد دو رقم انتهایی شماره دانشجویی اعضای گروه نمایش داده شود، و دو نمایشگر سمت چپ مقدار صفر را نشان دهند. روی مانیتور نیز صفحه بازی با پرنده غیر متحرک و موانع هر دیواره نمایش دهد. تمام LEDها نیز خاموش باشند.
- ✓ با زدن یکی از Buttonها بازی آغاز شود و مربع (پرنده) به سمت دیواره طرف مقابل حرکت کند، و بازیکن با فشار دادن یکی از Buttonها پرنده را کنترل کند.
- ✓ دو 7Segment میانی زمان گذشته از بازی را نشان دهند.
- ✓ دو 7Segment امتیاز بازیکن را نشان دهند، بدین‌صورت که با هر بار برخورد پرنده به دیواره‌ها یک امتیاز اضافه شود و موانع دیوار روبرو بصورت رندوم ظاهر شوند.



- ✓ هر 15 ثانیه باید سرعت حرکت پرنده را افزایش دهید.
- ✓ شرط انتهایی بازی، رسیدن زمان به 99 ثانیه، یا رسیدن امتیاز یک بازیکن به 10 یا سوختن پرنده است. شرط سوختن پرنده برخورد به موانع می باشد.
- ✓ در این حالت بازی متوقف شود و همچنین تمام LED ها روشن شوند.

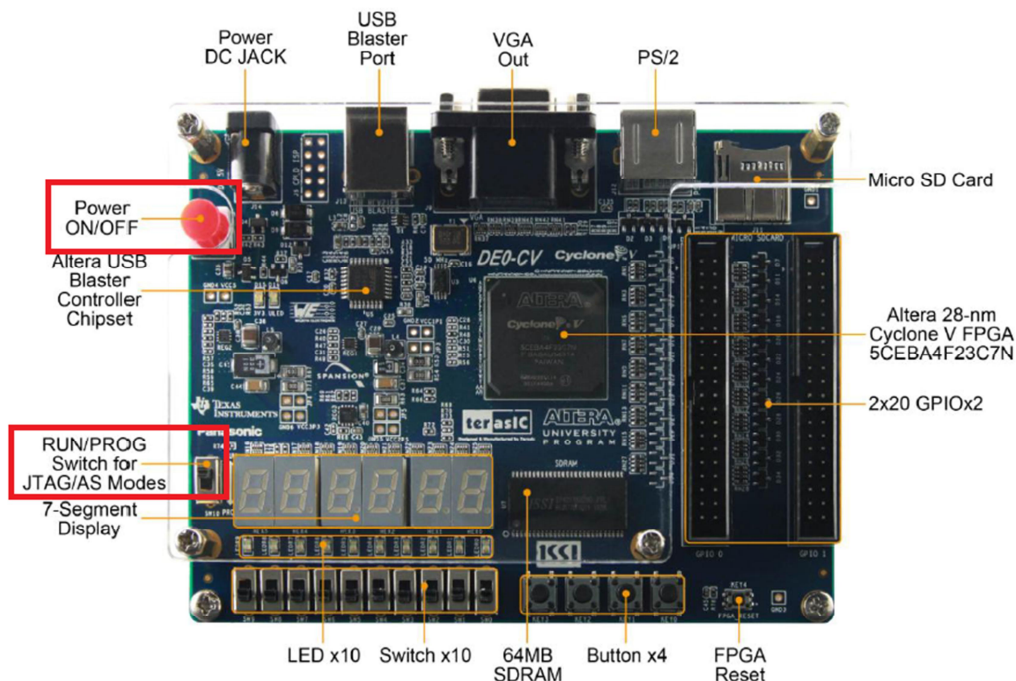
بخش اختیاری:

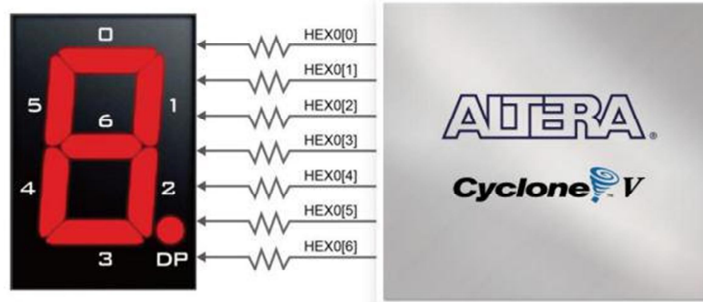
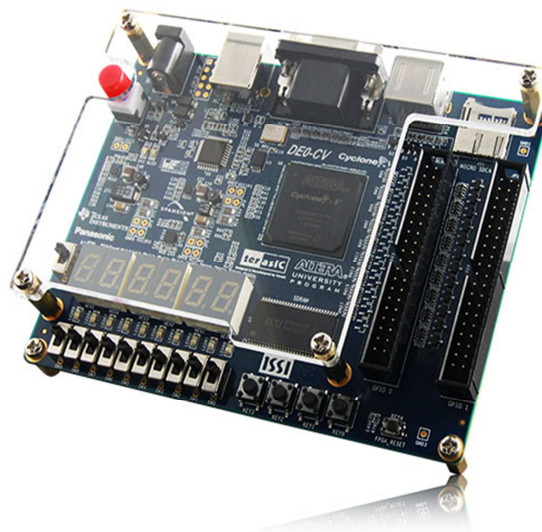
- بخش اختیاری اول – پرنده به صورت سهمی وار به سمت پایین حرکت کند.
- بخش اختیاری دوم – پرنده به صورت خودکار حرکت کرده و کسب امتیاز کند.

توجه مهم: در برد Altera-DE0، کلید Reset و Button ها، و خروجی های مربوط به 7Segment ها همه بصورت Active-Low هستند.

برخی نکات:

- ✓ ورودی های پروژ Clock، Reset_n و چهار Button است (یک Button برای حرکت پرنده کافیست).
- ✓ نمای برد را در پایین مشاهده می کنید. برد دارای شش نمایشگر، و تعدادی Button و LED و سوییچ (SW) است.
- ✓ برای **Push-Button** ها از پورت های KEY (بصورت Active-Low) استفاده کنید.
- ✓ برای نمایشگرهای **7Segment** (6 نمایشگر، هر کدام شامل 7 پین Active-Low) از پورت های HEX0 تا HEX5 استفاده کنید.
- ✓ برای LED ها از پورت های LEDR استفاده کنید.
- ✓ برای **سوییچ** ها از پورت های SW استفاده کنید.
- ✓ برای نمایش چهاربیت روی 7Segment می توانید از تابع convSEG داخل برنامه کمک بگیرید.
- ✓ برای فعال کردن **USB-Blaster**، بعد از اتصال برد به کامپیوتر به کمک فایل “QUARTUS II” که روی دراپ باکس هست، مراحل نصب درایور USB-Blaster را انجام دهید.





توجه (تولید اعداد تصادفی): برای تولید اعداد تصادفی می‌توانید از کد زیر استفاده کنید و به تعداد بیت‌هایی که لازم دارید از خروجی‌های آن استفاده کنید:

```
process(clk)
-- maximal length 32-bit xnor LFSR
function lfsr32(x : std_logic_vector(31 downto 0)) return std_logic_vector is
begin
    return x(30 downto 0) & (x(0) xnor x(1) xnor x(21) xnor x(31));
end function;
begin
    if rising_edge(clk) then
        if resetn='0' then
            pseudo_rand <= (others => '0');
        else
            pseudo_rand <= lfsr32(pseudo_rand);
        end if;
    end if;
end process;
```

- برای LFSR سی و دو بیتی (LFSR-32)، بیت‌های 0 و 1 و 21 و 31 را با هم XNOR کنید.
- برای LFSR شصت و چهار بیتی (LFSR-64)، بیت‌های 59 و 60 و 62 و 63 را با هم XNOR کنید.
- برای LFSR صد و بیست و هشت بیتی (LFSR-128)، بیت‌های 98 و 100 و 125 و 127 را با هم XNOR کنید.

موفق باشید

امیر اتفاقی، مهران رفیعی، سهیل فلاح، فاطمه گل‌پور
مهدی امینیان