



- (۱) پروژه ها برای گروه دو نفره است. اعضای گروه را به من اعلام کنید.
- (۲) پروژه باید توسط نرم افزار *Altera Quartus* (برای بورد *Altera DE0-CV*) یا *Xilinx ISE* (برای بورد *Spartan6*) پیاده سازی شود.
- (۳) کل پوشه های مربوط به پیاده سازی و گزارش را در یک فایل zip شده به نام "CADProject_Name1_Name2" قرار دهید.
- (۴) قبل از زمان تحویل حضوری، فقط یک فایل zip شده به آدرس زیر ارسال کنید:
<https://www.dropbox.com/request/3knDRN3iOOxdTWABo7Nh>
زمان تحویل نهایی پروژه، فقط بصورت حضوری، در تاریخ سه شنبه ۱۰ بهمن یا چهارشنبه ۱۱ بهمن خواهد بود (زمان دقیق بعدا اعلام خواهد شد).
- (۵) 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 *CAD_VGA_Xilinx* (برای Xilinx ISE) که روی دراپ باکس قرار داده شده است کمک بگیرید. قالب اصلی این فایل (پورت های فایل اصلی) 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 بازمی 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 بگونه ای تکمیل کنید که بازی زیر را اجرا کند.

Neon Blaster

هدف از این پروژه پیاده سازی بازی Neon Blaster است. بازیکن دارای یک خانه با قابلیت شلیک در پایین زمین بازی وجود دارد که بصورت ثابت و سرعت ثابت در حال شلیک گلوله به بالا هست و توسط بازیکن به کمک کلیدها صرفاً می تواند به چپ یا راست حرکت کند (از انتهای گوشه چپ یا راست نمی تواند خارج شود). از بالا نیز بصورت رندم، مربع هایی با زاویه رندم شروع به پایین آمدن می کنند (تا مربع قبلی در زمین قرار دارد، مربع جدیدی وارد زمین بازی نشود). اگر گلوله بازیکن به این مربع اصابت کند، امتیاز بازیکن افزایش می یابد. در غیر این صورت، وقتی مربع به پایین رسید، براساس زاویه برخورد دوباره به سمت بالا و سپس پایین حرکت می کند. هر موقع این مربع به بازیکن برخورد کند، بازیکن می باززد.

در ابتدا زمان و امتیاز شما صفر است. با زدن اولین کلید برای حرکت مهره بازیکن، بازی آغاز می شود و تایمر بازی از صفر شروع به شمارش می کند و هر یک ثانیه باید افزایش بیابد. هر بار که گلوله به مربع های وارد شده صابت کرد، یک امتیاز به امتیاز بازیکن و اضافه می شود. اگر مربع وارد شده به مهره بازیکن برخورد کند، بازیکن می باززد و بازی تمام است. اگر زمان بازی به شصت برسد یا امتیاز بازیکن به ده برسد، بازیکن برنده می شود و بازی تمام است.



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 شوند.
- ❖ با زدن یکی از Buttonها، بازی آغاز شود. بازیکن توسط دو Button، جهت حرکت مهره خود را کنترل کند. مهره بازیکن دائما و با سرعت ثابت در حال شلیک گلوله به بالا هست.
- ❖ با شروع بازی، دو 7Segment زمان بازی را نشان دهند، به این صورت که با آغاز بازی تایمر زمان صفر را نشان دهد و هر ثانیه یکی اضافه شود.
- ❖ دو 7Segment امتیاز بازیکن را نشان دهند، به این صورت که با هر بار اصابت گلوله به مربع ورودی یکی به امتیاز بازیکن اضافه شود.
- ❖ مربع‌ها باید بصورت رندم از مختصات در بالای صفحه زمین و با زاویه رندم وارد زمین بازی شوند.
- ❖ شرط برد بازی، رسیدن به امتیاز ۱۰ یا رسیدن زمان بازی به شصت ثانیه می‌باشد. در هنگام برد، بازی باید متوقف شود و تایمر شمارش نکند و چراغ‌های برد روشن شوند.
- ❖ شروط باخت نیز، برخورد مربع ورودی به مهره بازیکن می‌باشد. در این حالت نیز باید بازی و تایمر متوقف شوند و چراغ‌ها روشن شوند.



بخش‌های اختیاری:

بخش اختیاری اول – همانند بازی اصلی، مربع‌ها با اعداد مختلف وارد شوند (حداقل تا چهار برسند)، که نشان‌دهنده تعداد گلوله مورد نیاز برای نابودی آنهاست. بنابراین، امتیاز بازیکن باید به اندازه اعداد روی مربع افزایش یابد و شرط خاتمه فقط براساس زمان شصت ثانیه یا باخت بازیکن باشد. (یک و نیم نمره)

بخش اختیاری دوم – سرعت ورود و حرکت مربع‌ها، با ورود هر مربع افزایش یابد. شرط اتمام بازی، همانند بخش اختیاری اول خواهد بود. (نیم نمره)

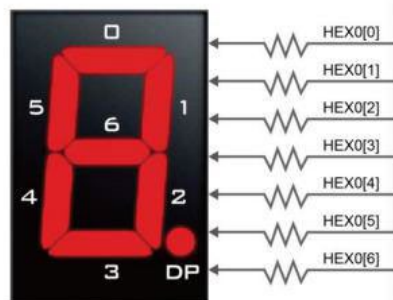
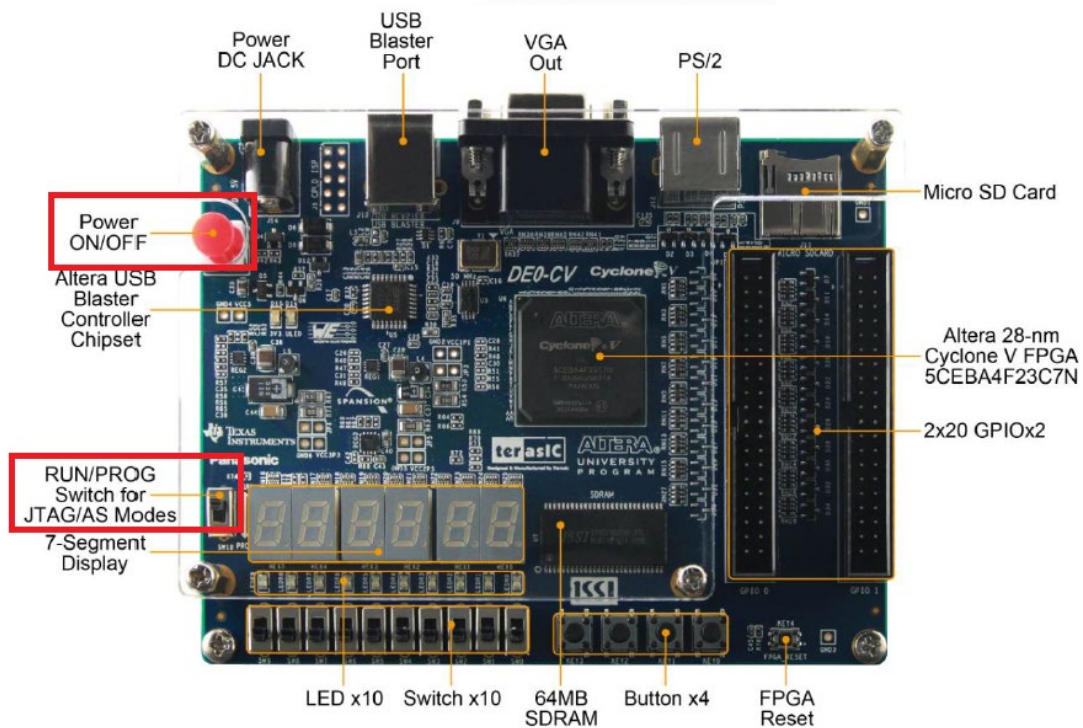
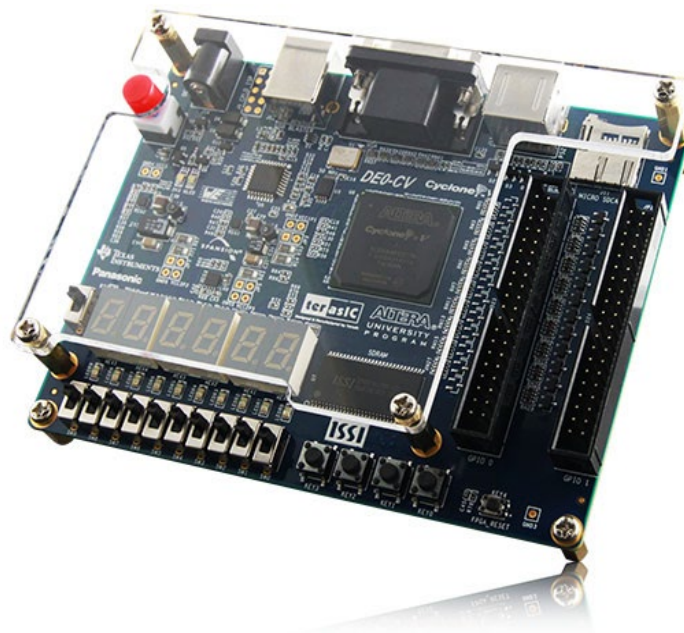
توجه (تولید اعداد تصادفی): برای تولید اعداد تصادفی می‌توانید از کد زیر استفاده کنید و به تعداد بیت‌هایی که لازم دارید از خروجی‌های آن استفاده کنید:

```
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;
```

-توجه- برای-بورد Altera : در بورد Altera-DE0، کلید Reset و Button‌ها، و خروجی‌های مربوط به 7Segment‌ها همه بصورت Active-Low هستند.

برخی نکات:

- ❖ نمای بورد را در پایین مشاهده می‌کنید. بورد دارای شش نمایشگر، و تعدادی Button و LED و سویچ (SW) است.
- ❖ برای **Push-Button** ها از پورت‌های KEY (بصورت Active-Low) استفاده کنید.
- ❖ برای نمایشگرهای **7Segment** (۶ نمایشگر، هر کدام شامل ۷ پین Active-Low) از پورت‌های HEX0 تا HEX5 استفاده کنید.
- ❖ برای **LED** ها از پورت‌های LEDR استفاده کنید.
- ❖ برای **سویچ** ها از پورت‌های Switch استفاده کنید.
- ❖ برای نمایش چهاربیت روی **7Segment** می‌توانید از تابع convSEG داخل برنامه کمک بگیرید.
- ❖ برای فعال کردن **USB-Blaster**، بعد از اتصال بورد به کامپیوتر به کمک فایل "QUARTUS II" که روی دراپ‌باکس هست، مراحل نصب درایور USB-Blaster را انجام دهید.



موفق باشید

مهدی آقایی، امیر محمدخواه، سپهر نوروزی، محمد صادق پور

امینیان