



- (۱) پروژه ها برای گروه دو نفره (ارشدها بصورت تک نفره) است. اعضای گروه و پروژه انتخابی را تا تاریخ **شنبه ۲۷ آذر** ایمیل 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 **حضور**، فقط یک فایل zip شده به آدرس زیر ارسال کنید:  
<https://www.dropbox.com/request/3knDRN3iOOxdTWABo7Nh>
- (۵) زمان تحویل نهایی پروژه، **احتمالا بصورت غیر حضوری و آنلاین**، در تاریخ احتمالی **سه شنبه پنجم بهمن** خواهد بود.
- (۶) **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) 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 بازمی گردد. چند خروجی نمونه نیز روی 7Segment ها نمایش داده شده است. ابتدا کد قرار داده شده را بصورت کامل نگاه کنید و کاملا رفتار کد را تحلیل کنید. سپس این کد را باید به گونه ای تکمیل کنید که بازی را اجرا کند.

## بازی Battle City

هدف از این پروژه پیاده سازی نسخه ی ساده شده ی بازی Battle City است که ابتدا در سال ۱۹۸۵ توسط Namco عرضه شد. در این بازی که به شکل **تک نفره** اجرا خواهد شد، شما کنترل یک تانک را بر عهده دارید و هدف شما عبور از یک Maze و تسخیر یک نقطه ی مشخص در نقشه ی بازی در یک زمان مشخص است.





تانک شما توانایی حرکت در ۴ جهت بالا، پایین، چپ و راست را دارد و در صورت عدم وجود دکمه‌ی ۵ ام بر روی بورد انتخاب شده، تانک شما باید در فواصل زمانی مشخص به سمت جلو شلیک کند. کد قرار داده شده را باید به گونه‌ای تکمیل کنید که صفحه نمایش را به یک جدول دو بعدی فرضی تقسیم کند. در هر شروع بازی، Map بازی باید به صورت کاملاً تصادفی توسط خانه‌های این جدول Generate شود. هر خانه می‌تواند یکی از حالات زیر را نمایش دهد:

- فضای قابل رفت و آمد تانک: همانطور که از اسم آن مشخص است، تانک می‌تواند در این فضا رفت و آمد داشته باشد. این فضا می‌تواند از نظر Surface به دو حالت تقسیم شود:
  - Ice: سرعت حرکت تانک بر روی این Surface سریع تر از حالت معمول است.
  - Blank Space: فضایی که در Gameplay حرکتی تانک و مخفی شدن آن تأثیری ندارد و تانک می‌تواند به طور مشخص در این فضا با سرعت معمول حرکت کند.
- Brick Walls: این Block با اصابت مقدار مشخصی از گلوله نابود شده و به فضای خالی (Blank Space) و قابل عبور برای تانک تبدیل می‌شود.
- Metal Walls: این نوع از Block غیر قابل نفوذ (Impenetrable) است و با اصابت گلوله‌ی تانک نابود نخواهد شد.
- Flag: تنها یک خانه از نقشه را شامل می‌شود و هدف تانک‌ها رسیدن و تسخیر این نقطه از روی نقشه است. این خانه معمولاً توسط خانه‌های Brick Walls و Metal Walls محاصره شده است (باید توجه داشت که حتماً باید یک راه ورودی به این خانه وجود داشته باشد یا قابل بازگشایی باشد).
- Tank: مکان قرار گیری تانک در شروع بازی است که در ردیف بالای نقشه قرار می‌گیرد.

**توجه:** برخورد تانک با Block های Brick Wall و Metal Wall باعث تغییر جهت تانک به سمت مخالف می‌شود.

در این بازی تانک‌ها و Block های مختلف توسط مربع‌هایی به رنگ‌های مشخص برای هر نوع Block نشان داده می‌شوند. هر تانک باید با عبور کردن از Maze به وجود آمده در هر بار Generate شدن نقشه، خود را به نقطه‌ی پایان برساند. با شروع حرکت تانک در نقشه، تایمر به شکل ثانیه شمار در 7-Segment ها شروع به شمردن می‌کند. (دو 7-Segment برای نمایش ثانیه و یک 7-Segment برای نمایش دقیقه).

برای اجرای بازی اصلی می‌توانید از این لینک استفاده کنید:

[https://www.retrogames.cz/play\\_014-NES.php](https://www.retrogames.cz/play_014-NES.php)

### برای پروژه مراحل زیر را پیاده‌سازی کنید:

- ❖ بعد از زدن کلید Reset، روی نمایشگرهای 7Segment روی بورد دو رقم انتهایی شماره دانشجویی اعضای گروه نمایش داده شود، و دو نمایشگر سمت چپ مقدار صفر را نشان دهند. روی مانیتور نیز نقشه‌ی Generate شده بدون تانک ها نمایش داده شود تا یکی از Button ها زده شوند. تمام LED ها نیز خاموش باشند.
- ❖ به نوع و عملکرد هر Block توجه کنید و مطمئن شوید هر نوع Block حداقل یک بار در صفحه Generate خواهد شد. وجود Brick Wall ها و Metal Wall به صورت دسته‌ای و منظم باعث بهبود Gameplay بازی خواهد شد. Ice به صورت دسته‌ای (حداقل ۳ خانه‌ی کنار یکدیگر) و نه تکی Generate شود.
- ❖ با زدن یکی از Button ها بازی آغاز شود و تانک سر جای خودش قرار داده شوند. بازیکن با فشار دادن کلید ها تانک را کنترل کند.
- ❖ سه 7-Segment زمان گذشته از بازی را نشان دهند.
- ❖ شرط انتهایی بازی، رسیدن بازیکن به نقطه‌ی Flag است و شرط Failure گذشت زمان از 9:59 باشد.
- ❖ در این حالت بازی متوقف شود و تانک حرکت نکند. همچنین تمام LED ها روشن شوند.
- ❖ با زدن کلید Reset، Map بازی باید دوباره به صورت تصادفی Generate شود.

### بخش اختیاری:

بخش اختیاری – تانک دومی در نقشه به صورت خودکار و هوشمند بازی و رقابت کند.

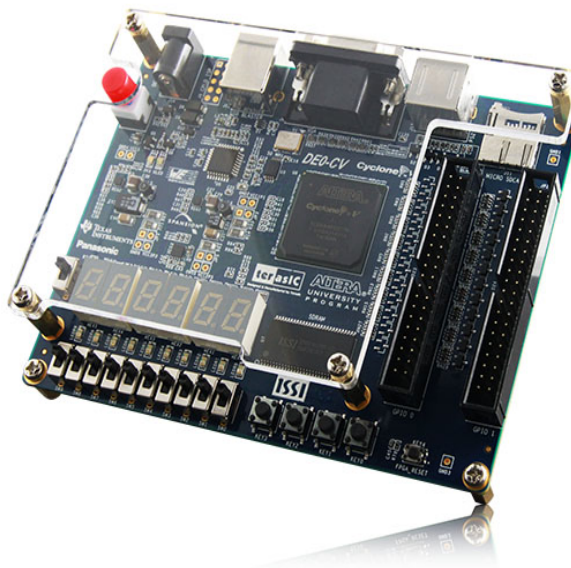
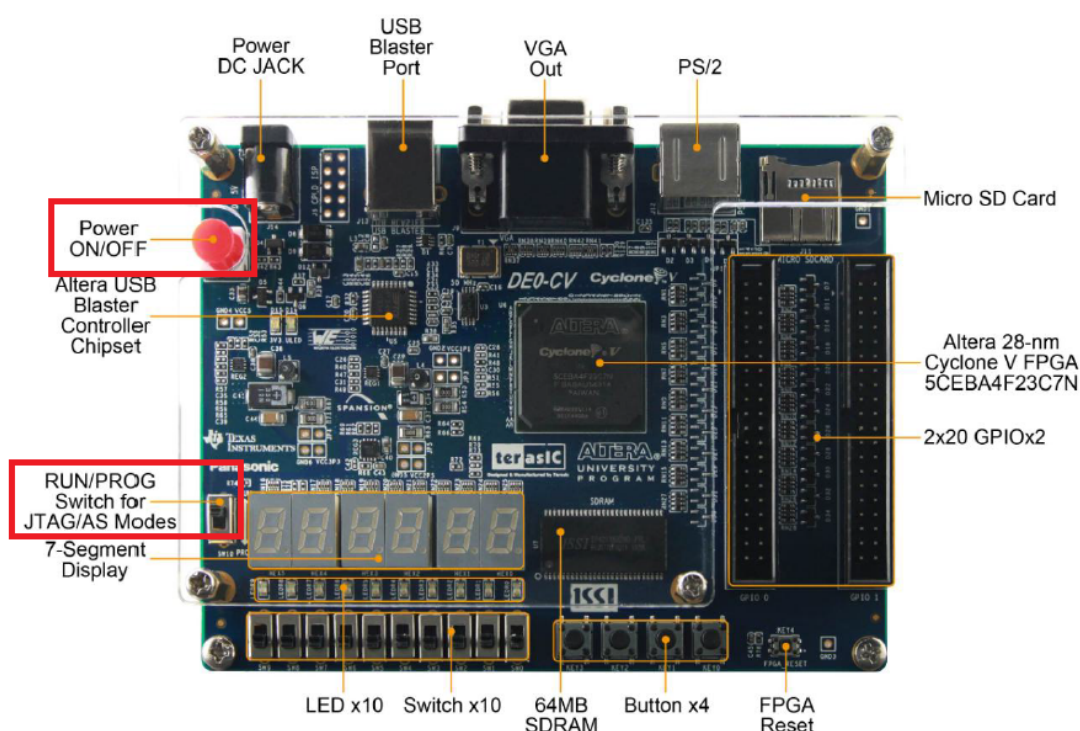


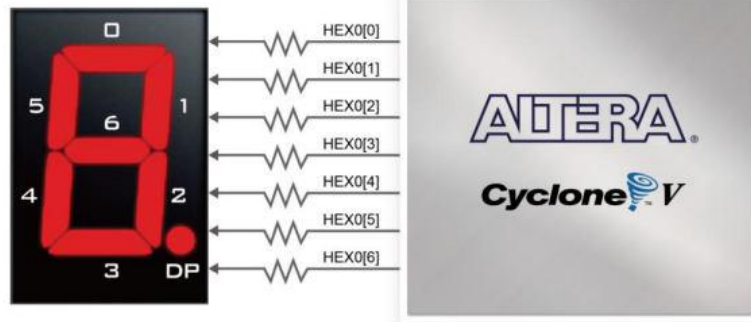
University of Guilan  
Computer Engineering Department

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

**برخی نکات:**

- ❖ ورودی‌های پروژۀ Clk، Reset\_n و چهار Button است (یک Button برای حرکت پرندۀ کافیسٹ).
- ❖ نمای برد را در پایین مشاهده می‌کنید. برد 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 SW استفاده کنید.
- ❖ برای نمایش چهاربیت روی 7Segment می‌توانید 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 دراپ‌باکس 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**

طراوت منصف، مهدی آرزومند، سهیل فلاح، امیرحسین هادی‌پور  
مهدی امینیان