

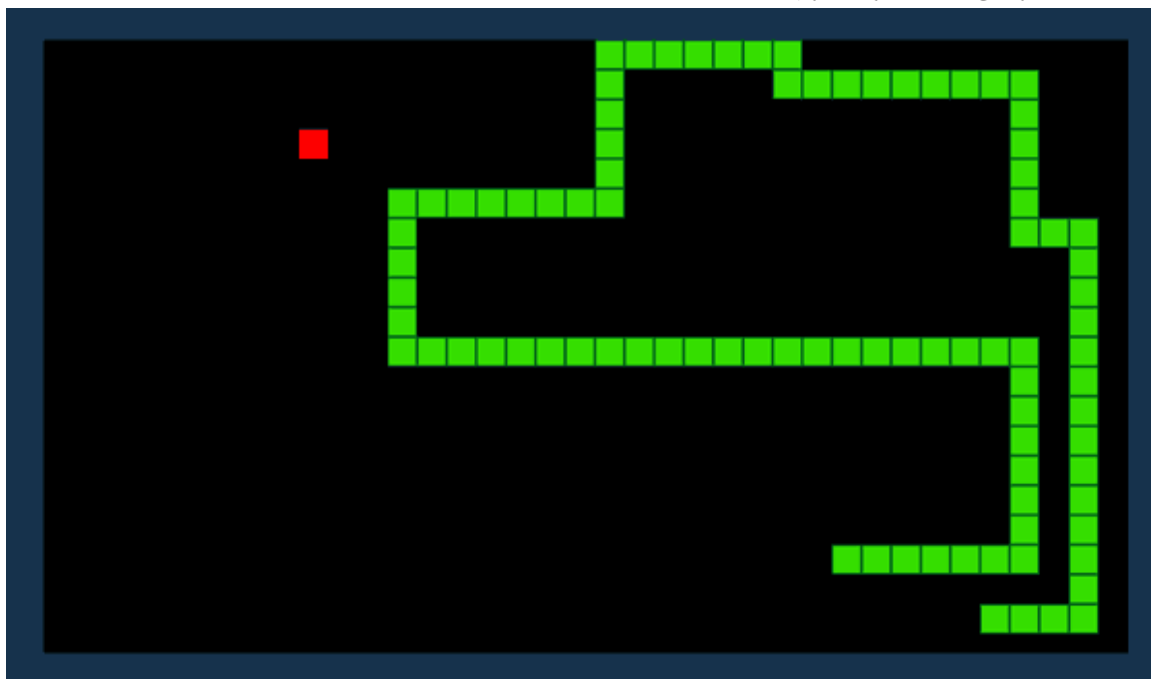


- (۱) پروژه ها برای گروه دو نفره (ارشدها بصورت تک نفره) است. اعضای گروه را تا تاریخ سوم خرداد ایمیل کنید.
- (۲) پروژه باید توسط نرم افزار *Altera Quartus* (برای بورد *Altera DE0-CV*) یا *Xilinx ISE* (برای بورد *Spartan6*) پیاده سازی شود.
- (۳) کل پوشه های مربوط به پیاده سازی و گزارش را در یک فایل zip شده به نام "*CADProject\_Name1\_Name2*" قرار دهید.
- (۴) قبل از زمان تحویل حضوری، فقط یک فایل zip شده به آدرس زیر ارسال کنید:  
زمان تحویل نهایی پروژه، بصورت حضوری، در تاریخ احتمالی شنبه ۱۱ تیر خواهد بود (تاریخ دقیق متعاقبا 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) 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 بگونه ای تکمیل کنید که بازی را اجرا کند.

## Snake

هدف از این پروژه پیاده سازی بازی Snake است. یک مار با طول اولیه (مثلا ۴ خانه) که در حال حرکت است و بازیکن باید با خوردن خانه ای که بصورت رندوم روشن شده است، طول مار را افزایش دهد و امتیاز کسب کند. مار می تواند از هر سمتی از صفحه خارج شود و از طرف دیگر صفحه، وارد بازی شود. به طور مثال اگر از سمت راست صفحه خارج شود، باید از همان ردیف، ولی از سمت چپ صفحه وارد شود. و یا اگر از بالای صفحه خارج می شود، باید از همان ستون، ولی از پایین صفحه وارد شود. در بازه های زمانی مشخص، باید یک بمب در صفحه ظاهر شود و اگر مار به آن برخورد کند، و یا در زمان منفجر شدن آن و پس از زمانی مشخص هنوز سر مار در حوالی آن باشد (شعاع انفجار هر مرحله بزرگتر شود)، از بین خواهد رفت و بازی تمام خواهد شد. در هر دور از بازی، موانعی نیز به طور رندوم در بازی قرار بگیرند.





### 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 و مار شروع به حرکت کند. بازیکن توسط چهار Button جهت حرکت مار را کنترل کند.
- ❖ دو 7Segment زمان گذشته از بازی را نشان دهند.
- ❖ دو 7Segment امتیاز بازیکن را نشان دهند، بدین صورت که یک خانه بصورت رندوم روشن شود و با برخورد مار به آن یکی به طول مار افزوده شود و یک امتیاز به بازیکن داده شود.
- ❖ بصورت رندم بمب‌هایی در صفحه ظاهر شوند و بعد از چند لحظه منفجر شوند. بمب‌ها نباید روی بدنه مار ظاهر شوند.
- ❖ هر ۱۵ ثانیه باید سرعت حرکت مار را افزایش دهید، تعداد بمب‌های بیشتری ظاهر شوند، و شعاع انفجار افزایش یابد.
- ❖ شرط برد بازی، رسیدن به امتیازی مشخص و یا تمام شدن زمان بازی (۹۹ ثانیه) می باشد. در هنگام برد، بازی باید متوقف شود و تایمر شمارش نکند و چراغ‌های برد روشن شوند.
- ❖ شروط باخت نیز، برخورد سر مار با اجزای بدن خود، برخورد با موانع یا بمب، و یا از بین رفتن توسط بمب می باشد. در این حالت نیز باید بازی و تایمر متوقف شوند و چراغ‌ها روشن شوند

### بخش اختیاری:

- بخش اختیاری اول – هر ثانیه موانع ثابتی به صفحه بازی اضافه شوند (با رنگ مجزا) تا در صورت برخورد مار بازی تمام شود.
- بخش اختیاری دوم – علاوه بر بخش قبل، مار بصورت خودکار و هوشمند بازی کند.

**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:

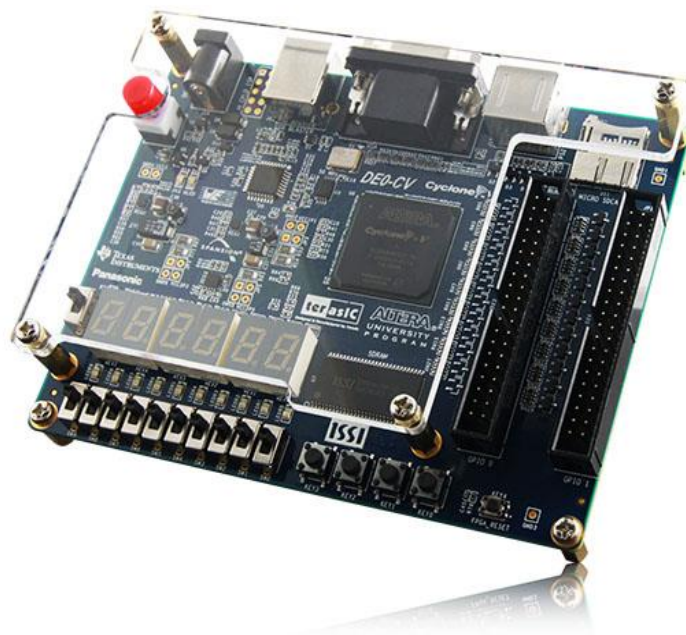
```
process(clk)
-- maximal length 32-bit xor LFSR
function lfsr32(x : std_logic_vector(31 downto 0)) return std_logic_vector is
begin
    return x(30 downto 0) & (x(0) xor x(1) xor x(21) xor x(31));
end function;
begin
    if rising_edge(clk) then
        if resetn='0' then
            pseudo_rand 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



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

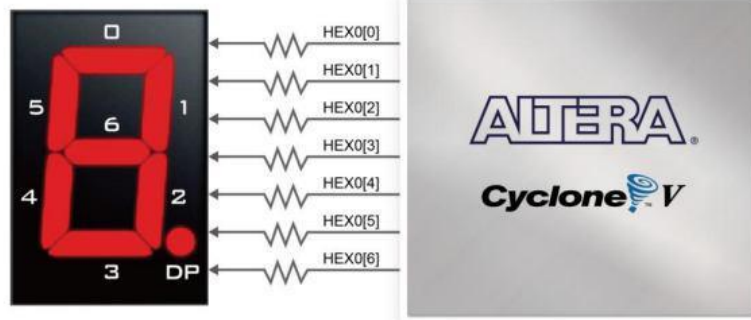
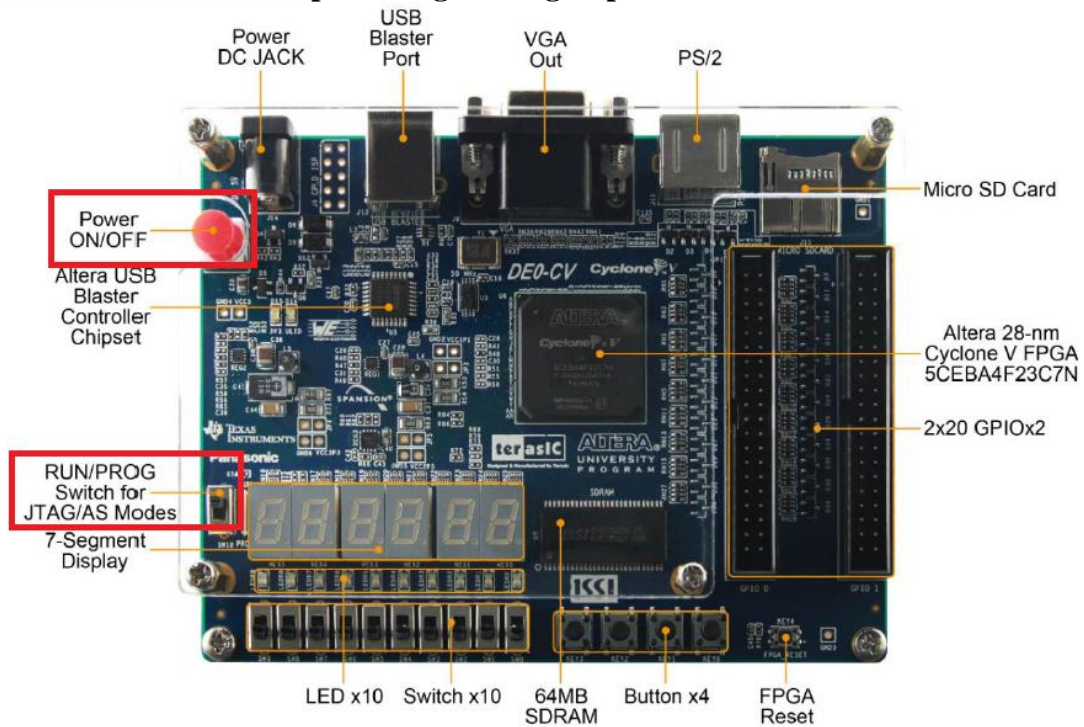
### برخی نکات:

- ❖ ورودی‌های پروژۀ Clk، Reset\_n و چهار Button برای حرکت مار یا انتخاب رنگ است.
- ❖ نمای برد را در بالا مشاهده می‌کنید. برد دارای شش نمایشگر، و تعدادی Button و LED و سوییچ (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 **سوییچ** 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 برد به کامپیوتر به کمک فایل "QUARTUS II" که روی دراپ‌باکس هست، مراحل نصب درایور USB-Blaster را انجام دهید.





University of Guilan  
Computer Engineering Department



موفق باشید  
احمدی، کاوش، حبیبی  
امینیان