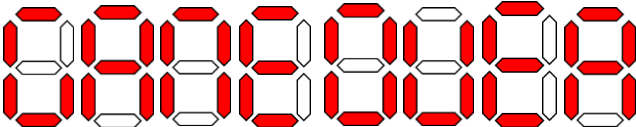


## Opgave 7: Space invaders

### Overzicht opdracht

Stuur een VGA scherm aan en maak hierop het spel “[Space invaders](#)”. De basisopgave omvat:

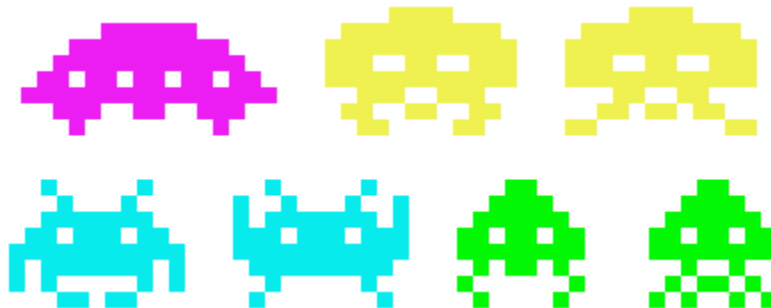
- een rode kader van exact 1 pixel breed op de buitenste rand van het scherm (dus een rechthoek van pixel (0,0) tot en met pixel (639,479)).
- een kanon onderaan het scherm. Dit mag een rechthoek zijn.
- de levens van de speler op het rechtse 7-segment display.
- de score van de speler (tot 9999) op de vier linkse 7-segment displays.
- beweging van het kanon. Dit moet gebeuren aan de hand van de BTNL en BTNR knoppen op het Digilent bord. Let op dat het kanon niet door de rand van het scherm gaat.
- de mogelijkheid van het kanon om één kogel tegelijk recht omhoog te schieten. Er kan pas een volgende kogel gevuld worden van zodra de vorige kogel verdwenen is van het scherm. Een kogel vuurt af wanneer er op de BTNC knop gedrukt wordt.
- een bewegende alien (mag een rechthoek zijn) bovenaan het scherm. Deze beweegt horizontaal over het scherm tot hij de rand raakt. Dan schuift hij zijn eigen hoogte aan pixels naar beneden en begint in de andere richting horizontaal te bewegen (zoals in het filmpje hierboven). Als de alien ooit onderaan het scherm geraakt, verliest de speler een leven. Als de alien geraakt wordt door een kogel, dan verdwijnt hij en begint hij terug bovenaan het scherm.
- een “game over” melding op de 7-segment displays wanneer de speler 0 levens heeft:
 
- een looplicht waarbij er slechts één led tegelijkertijd oplicht en telkens over en weer beweegt van links naar rechts en vice versa. Dus eerst brandt led 15, dan 14, 13,... tot en met led 0 (van links naar rechts dus), daarna weer led 1, 2, 3,... tot en met led 15 (van rechts naar links dus), dan weer 14, 13, enz. Telkens wanneer het looplicht op het uiteinde is (led 15 of 0), dan gaat de snelheid van de alien omhoog. Zorg ervoor dat dit looplicht niet te snel gaat, zodat je het nog met het blote oog kan zien.

Als dit gelukt is (en slechts nadat dit gelukt is), mag je hier ook zelf uitbreidingen aan toevoegen. Hiervoor mag je je fantasie de vrije loop laten, maar ter inspiratie staan hieronder enkele voorbeelden:

- vier bunkers (rechthoeken) boven het kanon. Als een bunker geraakt wordt door een kogel en/of een alien, dan verdwijnt hij (gedeeltelijk).
- de alien schiet ook één kogel tegelijk, recht naar beneden:
  - Er kan pas een volgende kogel vertrekken, wanneer de vorige van het scherm verdwenen is.
  - Als de kogel een bunker raakt, verdwijnt de bunker.
  - Als de kogel het kanon raakt, verliest de speler een leven.
  - Je kan dit volgens vaste tijdstippen laten gebeuren (bijvoorbeeld één seconde tussen verschillende kogels), of je kan het pseudorandom<sup>1</sup> maken.
- meerdere aliens die bewegen zoals op het filmpje. Je kan je hiervoor laten inspireren door onderstaande code:

```
type t_AlienAlive is array (0 to 9, 0 to 3) of boolean;
type t_AlienX is array (0 to 9, 0 to 3) of integer range 0 to 639;
signal AlienAlive: t_AlienVisible := (others => (others => true));
signal AlienX: t_AlienX;
...
begin
  process...
  begin
    YLoop: for Y in 0 to 3 loop
      XLoop: for X in 0 to 9 loop
        AlienX(X,Y) <= (X*50)+10;
        ...
      end loop XLoop;
    end loop YLoop;
```

- figuurtjes in plaats van rechthoeken. Ter inspiratie de pixel figuurtjes uit het originele spel:



- alles wat het spel nog meer op het oorspronkelijke “Space Invaders” doet lijken.
- ... (eigen fantasietjes)

<sup>1</sup> Lees hoofdstuk 5.4.8 in het handboek (LFSR) om meer te weten over pseudorandom in hardware.

### Beschrijving

Vertrek van de vorige opgave voor het aansturen van de juiste VGA synchronisatiesignalen. Nu zal je echter de rood, groen en blauw componenten moeten aanpassen in plaats van alle pixels rood te kleuren.

Net zoals de HSync en VSync signalen in vorige opgave kan je de positie van de bal en het palletje in een synchroon proces bijhouden, terwijl je het sturen van de kleurinformatie hiervan volledig combinatorisch kan afleiden.

Nu, je hebt reeds de inkomende klok van 100MHz vertraagd naar 25MHz voor de pixel klok, maar dit is nog veel te snel om de beweging van de kogel(s) en kanon aan te sturen. Als je tegen een frequentie van 25MHz (iedere 40 ns) de kogel één pixel laat opschuiven, dan zal de kogel niet meer zichtbaar zijn op het scherm. Het menselijk oog ziet maximum ongeveer 20 beelden per seconde, wat dus overeenkomt met een frequentie van 20Hz. Als je werkt met een “spelklok” van ongeveer 100Hz, dan zullen alle elementen een vlotte beweging vertonen.