PROJEKT KAČA

Samo Hribar, Filip Gros

ZASTAVLJENI CILJI

- Implementirati vsem poznano igro kače na FPGA čip
- Zabaven in zanimiv gameplay
- Uporaba akselometra (gyrota) za premikanje
- VGA za prikaz slike
- 7-segmenti display za prikaz točk

MODULI PROJEKTA

- Design Sources (1)
  - top(Behavioral) (top.vhd) (7)
    - kaca_engine : kaca_engine(Behavioral) (kaca_engine.vhd) (1)
      - ram : generic_RAM(Behavioral) (generic_RAM.vhd)
    - snake_move_prescaler : prescaler(Behavioral) (prescaler.vhd)
    - kaca_premikalnik : kaca_premikalnik(Behavioral) (kaca_premikalnik.vhd)
    - displayRam : framebuffer_RAM2(Behavioral) (framebuffer_RAM2.vhd) (2)
      - index2sprite : index2sprite(Behavioral) (index2sprite.vhd)
      - ram : generic_RAM(Behavioral) (generic_RAM.vhd)
    - vgaController : vgaController(Behavioral) (vgaController.vhd) (2)
      - hsync : hsync(Behavioral) (hsync.vhd)
      - vsync : vsync(Behavioral) (vsync.vhd)
    - scoreDisplay : seven_seg_display(Behavioral) (seven_seg_display.vhd) (5)
      - prescalerForAnodes : prescaler(Behavioral) (prescaler.vhd)
      - prescalerOneSec : prescaler(Behavioral) (prescaler.vhd)
      - anodeSelect : anode_select(Behavioral) (anode_select.vhd)
      - valueToDigit : value_to_digit(Behavioral) (value_to_digit.vhd)
      - digitToSegments : digit_to_segments(Behavioral) (digit_to_segments.vh
    - Gyro : gyro(Behavioral) (Gyro.vhd) (1)
      - Inst_AccelerometerCtl : AccelerometerCtl(Behavioral) (AccelerometerCtl.
- Constraints (1)
- Simulation Sources (1)
  - sim_1 (1)
    - top_tp(Behavioral) (top_tp.vhd) (1)
      - uut : top(Behavioral) (top.vhd) (7)

```
type game_state is (
    CHECK_POS_0,
    CHECK_POS_1,
    CHECK_POS_2,
    CHECK_POS_3,
    CHECK_POS_4,
    DODAJ_SADEZ_0,
    DODAJ_SADEZ_1,
    POCAKAJ_ZAPIS_SADEZA,
    POPRAVI_STARO_GLAVO_0,
    POPRAVI_STARO_GLAVO_1,
    POCAKAJ_ZAPIS_STARE_GLAVE,
    ZAPISI_NOVO_GLAVO_0,
    POCAKAJ_ZAPIS_NOVE_GLAVE,
    POPRAVI_STARI_REP_0,
    POPRAVI_STARI_REP_1,
    POPRAVI_STARI_REP_2,
    POCAKAJ_ZAPIS_STAREGA_REPA,
    ZAPISI_NOVI_REP_0,
    ZAPISI_NOVI_REP_1,
    POCAKAJ_ZAPIS_NOVEGA_REPA,
    RESET_GAME,
    END_GAME
);
```

Smeri:

A	B	smer
0	0	desno
0	1	gor
1	0	levo
1	1	dol

```
type istate is (ASSIGN_POS, CHECK_POS, WRITE_POS);
```

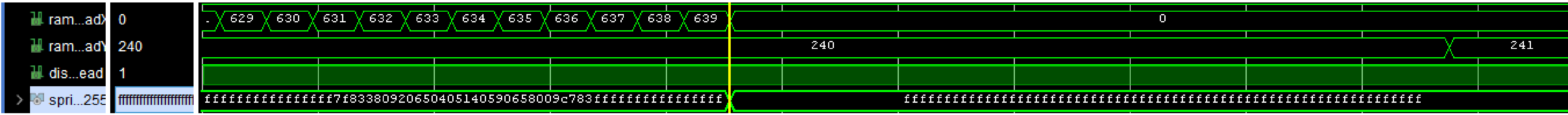
```
--asinhrono branje (enega bita)
display_bit_read <= sprite_image_vector((addr_readY mod sprite_size) * sprite_size + addr_readX mod sprite_size)
```

```
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111" when "00000",
00000000010000100001000001010010001010001001010000010100000110100000100001110000000001111111111111111111111111111111111111111111" when "00100",
00100100101111001100001100111100000000000001111000000000000011110000000000011100011111100011110100000011011110000000000011" when "00101",
00000000010010100001000010010010001010001001100000101000010111100001000001011111100000000011111111111111111111111111111111" when "00110",
```

```
(1|2|3|4|6 => '0', others => '1')    when digValue = 13 and showGameOver = '0' else --D
(0|3|4|5|6 => '0', others => '1')    when digValue = 14 and showGameOver = '0' else --E
(0|5|6|4 => '0', others => '1')      when digValue = 15 and showGameOver = '0' else --F
--napis GAME OVER, zakodiran kot G=0, A = 1..
(0|5|6|4|3|2 => '0', others => '1')  when digValue = 0 and showGameOver = '1' else --G = 6
(0|1|2|4|5|6 => '0', others => '1')  when digValue = 1 and showGameOver = '1' else --A
(2|6|4 => '0', others => '1')        when digValue = 2 and showGameOver = '1' else --M
(0|3|4|5|6 => '0', others => '1')    when digValue = 3 and showGameOver = '1' else --E
(0|1|2|3|4|5 => '0', others => '1')  when digValue = 4 and showGameOver = '1' else --O = 0
(1|2|3|4|5 => '0', others => '1')    when digValue = 5 and showGameOver = '1' else --V
(0|3|4|5|6 => '0', others => '1')    when digValue = 6 and showGameOver = '1' else --E
(4|6 => '0', others => '1')          when digValue = 7 and showGameOver = '1' --R
;
```

```
dol <= '0' when TO_INTEGER(unsigned(ACCEL_X)) < 255 + ACL_OFFSET else '1';
gor <= '0' when TO_INTEGER(unsigned(ACCEL_X)) > 255 - ACL_OFFSET else '1';
desno <= '0' when TO_INTEGER(unsigned(ACCEL_Y)) < 255 + ACL_OFFSET else '1';
levo <= '0' when TO_INTEGER(unsigned(ACCEL_Y)) > 255 - ACL_OFFSET else '1';
```

Primer simulacije: Izris Jabolka proti koncu vrstice



IZBOLJŠAVE

- Pavza
- 4 različne hitrosti igranja (težavnosti)
- Naključno generiranje sadežev
- Reset igre s pritiskom na CPU_RESET