

Mastermind on FPGA: logic and implementation

Revisione Progetto

Al fine di risolvere la problematica algoritmica riscontrata nella generazione dei feedback e per poter visualizzare la sequenza corretta durante la partita, sono state apportate modifiche alle seguenti entità:

- View
- Controller
- Datapath

Inoltre nell'entità *Mastermind* sono stati evidenziati alcuni segnali precedentemente non presenti nel modello.

CONTROLLER

Il controller deve ora recepire il nuovo segnale in ingresso SHOW_SECRET_BUTTON proveniente dallo switch 0, i segnali inviati al datapath restano invariati mentre alla view è inviato, oltre al REDRAW, il nuovo segnale SHOW_SECRET.

Vengono inoltre aggiunte le variabili *show_secret_old* e *in_play* per la corretta gestione della visualizzazione della sequenza corretta:

- la prima memorizza il precedente valore di show_secret_button e notifica alla view quando lo switch 0 è abilitato o disabilitato;
- la seconda la si utilizza per accertarsi che, quando viene richiesta la visualizzazione della sequenza segreta, si stia effettivamente giocando una partita evitando che all'utente appaia la schermata di gioco pur non avendo avviato una nuova partita o semplicemente che si possa visualizzare la sequenza segreta quando questa non è ancora stata generata.

```
library ieee;
use ieee.numeric_std.all;
use ieee.std_logic_1164.all;
use work.base_package.all;

entity controller is
port
(
    CLOCK                : in  std_logic;
    RESET_N              : in  std_logic;

    CHANGE_SQUARE_BUTTON : in  std_logic;
    CHANGE_COLOR_BUTTON  : in  std_logic;
    CONFIRM_GUESS_BUTTON : in  std_logic;
    SHOW_SECRET_BUTTON   : in  std_logic;

    -- Connections with Datapath
    CHANGE_SQUARE        : out std_logic;
    CHANGE_COLOR         : out std_logic;
    CONFIRM_GUESS        : out std_logic;

    -- Connections with View
    SHOW_SECRET          : out std_logic;
    REDRAW               : out std_logic
);

end entity;

architecture RTL of controller is
begin

    process (CLOCK, RESET_N)
        variable first_time      : std_logic;
        variable confirm_guess_old : std_logic;
        variable change_square_old : std_logic;
        variable change_color_old : std_logic;
        variable show_secret_old  : std_logic;
        variable in_play         : std_logic;
    begin
        if (RESET_N = '0') then
            CHANGE_SQUARE <= '0';
```

```

CHANGE_COLOR      <= '0';
CONFIRM_GUESS     <= '0';
SHOW_SECRET       <= '0';
REDRAW            <= '0';
first_time        := '1';
confirm_guess_old := '0';
change_square_old := '0';
change_color_old  := '0';
show_secret_old   := '0';
in_play           := '0';

```

```

elsif rising_edge(CLOCK) then

```

```

CHANGE_SQUARE <= '0';
CHANGE_COLOR  <= '0';
CONFIRM_GUESS <= '0';
REDRAW        <= '0';

```

```

if (first_time = '1') then

```

```

    first_time := '0';

```

```

    REDRAW <= '1';

```

```

elsif (change_square_old = '0' and CHANGE_SQUARE_BUTTON = '1') then

```

```

    CHANGE_SQUARE <= '1';

```

```

    REDRAW <= '1';

```

```

    in_play := '1';

```

```

elsif (change_color_old = '0' and CHANGE_COLOR_BUTTON = '1') then

```

```

    CHANGE_COLOR <= '1';

```

```

    REDRAW <= '1';

```

```

    in_play := '1';

```

```

elsif (confirm_guess_old = '0' and CONFIRM_GUESS_BUTTON = '1') then

```

```

    CONFIRM_GUESS <= '1';

```

```

    REDRAW <= '1';

```

```

    in_play := '1';

```

```

elsif (in_play = '1' and show_secret_old /= SHOW_SECRET_BUTTON) then

```

```

    SHOW_SECRET <= SHOW_SECRET_BUTTON;

```

```

    REDRAW <= '1';

```

```

    show_secret_old := SHOW_SECRET_BUTTON;

```

```

end if;

```

```

change_square_old := CHANGE_SQUARE_BUTTON;

```

```

change_color_old := CHANGE_COLOR_BUTTON;

```

```

confirm_guess_old := CONFIRM_GUESS_BUTTON;

```

```

end if;

```

```

end process;

```

```

end architecture;

```

VIEW

Nella view sono stati inseriti il segnale SHOW_SECRET e lo stato associato DRAW_SECRET per poter visualizzare la sequenza corretta a video utilizzando lo switch 0; di seguito il codice.

```
when DRAW_SECRET =>
  FB_COLOR      <= guess_peg_to_color(CHECK(square_to_draw));
  FB_X0         <= LEFT_MARGIN + LEFT_SPACING_TO_CENTER +
    (LEFT_SPACING_TO_CENTER / 2)
    + SQUARE_SPACING * square_to_draw
    + SQUARE_SIZE * square_to_draw;
  FB_Y0         <= TOP_MARGIN * 4 + TOP_SPACING_TO_CENTER * 3
    + GUESSES_NUMBER * (SQUARE_SIZE + SQUARE_SPACING)
    + SQUARE_SPACING + SQUARE_SIZE;
  FB_X1         <= LEFT_MARGIN + LEFT_SPACING_TO_CENTER
    + (LEFT_SPACING_TO_CENTER / 2)
    + SQUARE_SPACING * square_to_draw
    + SQUARE_SIZE * (square_to_draw + 1);
  FB_Y1         <= TOP_MARGIN * 4 + TOP_SPACING_TO_CENTER * 3
    + GUESSES_NUMBER * (SQUARE_SIZE + SQUARE_SPACING)
    + SQUARE_SPACING + SQUARE_SIZE * 2;
  FB_DRAW_FILLED_RECT <= '1';
  if (square_to_draw + 1 < SQUARES_NUMBER) then
    square_to_draw <= square_to_draw + 1;
    substate      <= DRAW_SECRET;
  else
    square_to_draw <= 0;
    substate      <= FLIP_FRAMEBUFFER;
  end if;
```

DATAPATH

In questa entità sono state inserite due nuove variabili, settate a '0' nello stato Title, rispettivamente per:

variable spot_checked : std_logic_vector((SQUARES_NUMBER - 1) downto 0);
tenere traccia, ad ogni verifica di sequenza, di quanti colori giusti al posto giusto ci sono e l'effettiva loro posizione.

variable color_checked : std_logic_vector((SQUARES_NUMBER - 1) downto 0);
tenere traccia, ad ogni verifica di sequenza, di quanti colori giusti ci sono e l'effettiva loro posizione.

Ciò è stato fatto per inviare in modo corretto i feedback per l'utente elaborati nello stato SubAns, in particolare la corretta gestione dei quadrati bianchi riferiti a "colore esatto, posto sbagliato" nel caso in cui siano presenti colori duplicati. Dal codice osserviamo che nel primo **for** si fa il check di quanti colori giusti al posto giusto ci sono (quadrati neri) mentre nel secondo osserviamo se i colori inseriti dall'utente sono eventualmente presenti ma nel posto sbagliato: tutto ciò avviene non tenendo più conto dei colori corrispondenti ai quadrati neri e quindi già considerati e poi scartati.

```

when SubAns =>

    for i in 0 to SQUARES_NUMBER - 1 loop
        if check_r(i) = guess_board(current_guess_r)(i) then
            r_spot_count := r_spot_count + 1;
            spot_checked(i) := '1';
        end if;
    end loop;
    for i in 0 to SQUARES_NUMBER - 1 loop
        if spot_checked(i) = '0' then
            for j in 0 to SQUARES_NUMBER - 1 loop
                if check_r(i) = guess_board(current_guess_r)(j) and (spot_checked(j) =
'0' and color_checked(j) = '0') then
                    r_color_count := r_color_count + 1;
                    color_checked(j) := '1';
                    exit;
                end if;
            end loop;
        end if;
    end loop;

    spot_checked := (others => '0');
    color_checked := (others => '0');

    for i in 0 to SQUARES_NUMBER - 1 loop
        if i < r_spot_count then
            hint_board(current_guess_r)(i) <= PLACE;
        elsif i < r_spot_count + r_color_count then
            hint_board(current_guess_r)(i) <= COLOR;
        else
            hint_board(current_guess_r)(i) <= NOTHING;
        end if;
    end loop;

    current_guess_r := current_guess_r + 1;

    if (r_spot_count = 4) then
        GAME_WON <= '1';
        State <= Finished;
    elsif (countdown_r = 0) then
        GAME_LOST <= '1';
        State <= Finished;
    else
        r_spot_count := 0;
        r_color_count := 0;
        State <= Play;
    end if;

```