

Kółko i krzyżyk

Marcel Chołodecki 275818, IEA W12N

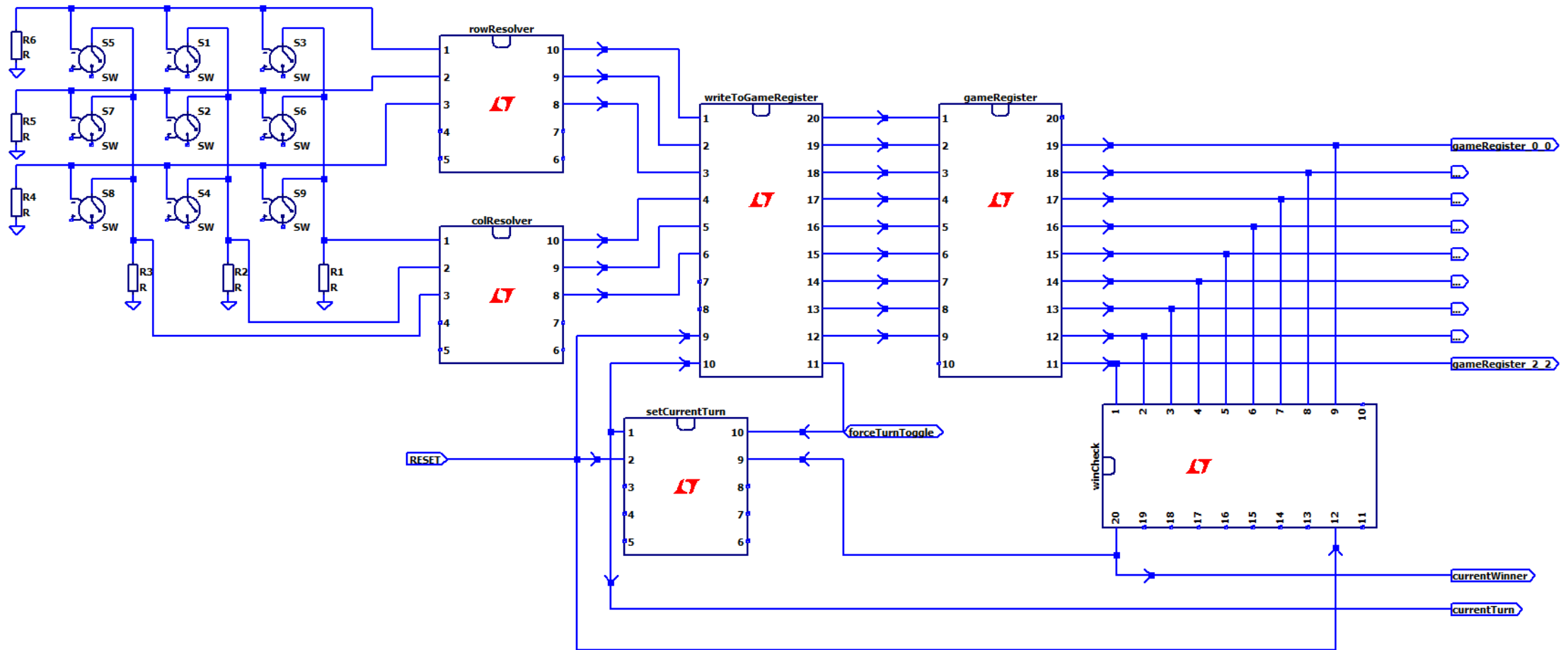
Odgórne wymagania

- Plansza 3x3
- Dynamiczne sprawdzanie wyniku
- Zapis planszy i wyniku do pliku

Wymagania własne

- System pracy na przyciskach (rząd + kolumna)
- Uniemożliwienie dwukrotnego wciśnięcia dwóch przycisków
- Praca asynchroniczna
- Wejścia: RESET, COL (2 downto 0), ROW (2 downto 0)
- Wyjścia: WINER, TURN, GAME (0 to 2) (0 to 2)
- Plansza: macierz 3x3 typu wyliczeniowego t_OX ('-', 'O', 'X')

Schemat blokowy układu



ConflictResolver

```
entity ConflictResolver is
  Port(
    input: in std_logic_vector(2 downto 0);
    output: out std_logic_vector(2 downto 0)
  );
end ConflictResolver;

architecture ConflictResolver_Arch of ConflictResolver is
  signal preOutput: std_logic_vector(2 downto 0) := (others => '0');
begin

  findConflict: process(input)
    variable bitsSet: Integer := 0;
  begin
    preOutput <= input;
    bitsSet := 0;
    for i in input'range loop
      if (input(i)) = '1' then
        bitsSet := bitsSet + 1;
      end if;
    end loop;

    if (bitsSet > 1) then
      -- jesli wektor wejscowy ma wiecej niz jeden element '1', to na wyjscie wystaw "000". Jesli nie, wystaw wejscie
      preOutput <= (others => '0');
    end if;
  end process findConflict;

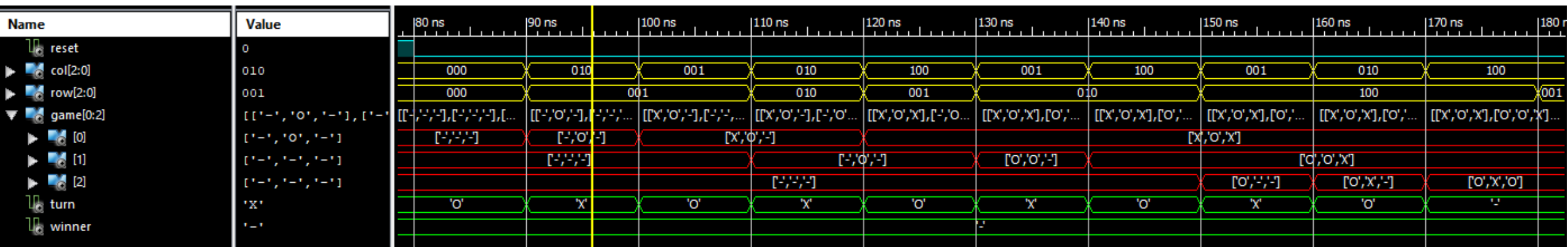
  output <= preOutput;

end ConflictResolver_Arch;
```

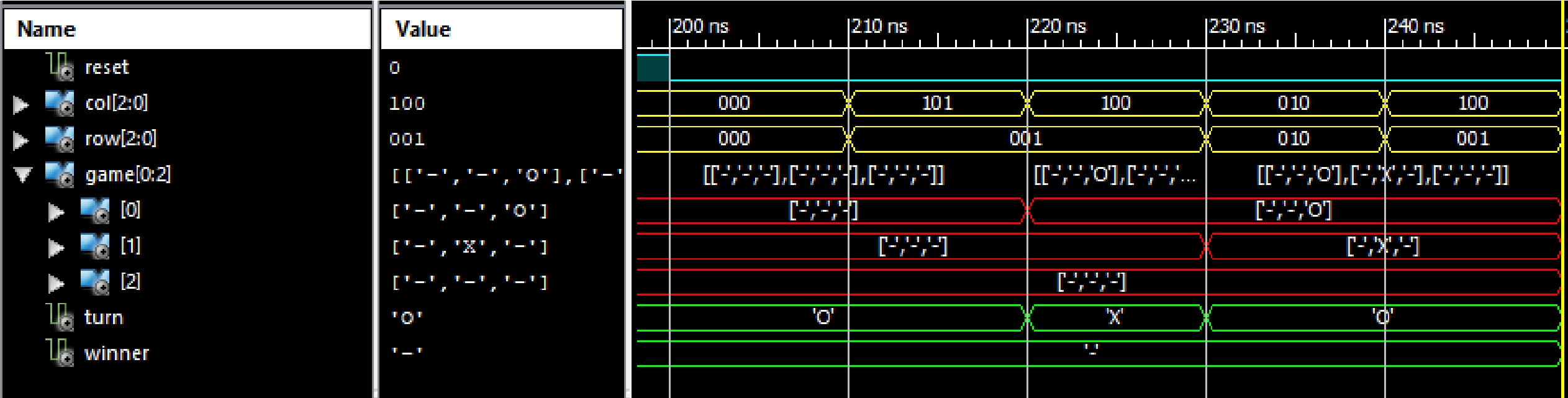
Test: wygrana, reset

[illegible]

Test: remis



Test: nieprawidłowe wejścia



Obsługa zapisu do plików wyjściowych

```
save_remis: process(WINNER, TURN, GAME)
begin
    if (TURN'event AND TURN = '-') then if (WINNER = '-') then
        for r in 0 to 2 loop
            write(remis_line, t_OX_to_char(GAME(r,0)) & ' ' & t_OX_to_char(GAME(r, 1)) & ' ' & t_OX_to_char(GAME(r, 2)));
            writeline(remis_file, remis_line);
        end loop;

        write(remis_line, "_____");
        writeline(remis_file, remis_line);
        write(remis_line, "REMIS");
        writeline(remis_file, remis_line);

    end if; end if;
end process save_remis;

save_win: process(WINNER, TURN, GAME)
begin
    if (TURN'event AND TURN = '-') then if (WINNER /= '-') then
        for r in 0 to 2 loop
            write(win_line, t_OX_to_char(GAME(r,0)) & ' ' & t_OX_to_char(GAME(r, 1)) & ' ' & t_OX_to_char(GAME(r, 2)));
            writeline(win_file, win_line);
        end loop;

        write(win_line, "_____");
        writeline(win_file, win_line);
        write(win_line, "WIN:" & t_OX_to_char(WINNER));
        writeline(win_file, win_line);

    end if; end if;
end process save win;
```

Pliki wyjściowe

```
RESULT_WIN.txt — Notatnik
Plik  Edycja  Format  Widok  Pomoc
p - -
X 0 -
X - 0
-----
WIN:0
```

```
RESULT_REMIS.txt — Notatnik
Plik  Edycja  Format  Widok  Pomoc
0 X 0
0 X X
X 0 0
-----
REMIS
|
```

Dziękuję za uwagę