

# *Dokumentacja projektu*

Architektura systemów komputerowych

Temat: Zegar czasu rzeczywistego w formacie: HH:MM:SS:MS.

Wykonał:  
Marcin Brach

## 1. Cel projektu

Celem projektu jest zaprojektowanie systemu komputerowego który składał się będzie z modułu procesora 32-bitowego Nios II interfejsu JTAG i modułu pamięci RAM oraz z modułem zegara czasu rzeczywistego w formacie HH:MM:SS:MS. Moduł portu należy zaprojektować i zaimplementować w języku vhd i dołączyć do systemu komputerowego, a następnie zaimplementować program testujący poprawność działania portu w języku C oraz przetestować go wykorzystując system prototypowy DE2.

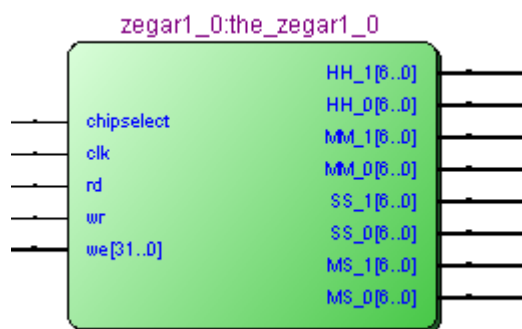
## 2. Organizacja projektowanego układu

Żegar ustawiany jest za pomocą szyny wejściowej we. Każda cyfra ustawiana jest osobno według schematu:

- we (3-0) – cyfra jednośmi milisekund
- we (7-4) – cyfra dziesiątek milisekund
- we (11-8) – cyfra jednośmi sekund
- we(15-12) – cyfra dziesiątek sekund
- we(19-16) – cyfra jednośmi minut
- we(23-20) – cyfra dziesiątek minut
- we(27-24) – cyfra jednośmi godzin
- we(31-28) – cyfra dziesiątek godzin

Każda cyfra ustawiana jest na 4-bitach co łącznie daje 32- bity.

## 3. Schemat projektowanego modułu



#### 4. Implementacja modułu w języku vhd1

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;
entity zegar1 is
port(  we  : in std_logic_vector(31 downto 0);
      clk  : in std_logic;
      wr   : in std_logic;
      rd   : in std_logic;
      chipselect : in std_logic;

      HH_1  : out std_logic_vector(6 downto 0);
      HH_0  : out std_logic_vector(6 downto 0);

      MM_1  : out std_logic_vector(6 downto 0);
      MM_0  : out std_logic_vector(6 downto 0);

      SS_1  : out std_logic_vector(6 downto 0);
      SS_0  : out std_logic_vector(6 downto 0);

      MS_1  : out std_logic_vector(6 downto 0);
      MS_0  : out std_logic_vector(6 downto 0)
);
end zegar1;
architecture RTL of zegar1 is
signal mili0: std_logic;
function dekoderekod (we: in std_logic_vector(3 downto 0)) return std_logic_vector is
variable M: std_logic_vector(6 downto 0);
begin
case we is
when "0000" => M := "1000000"; --0
when "0001" => M := "1111001"; --1
when "0010" => M := "0100100"; --2
when "0011" => M := "0110000"; --3
when "0100" => M := "0011001"; --4
when "0101" => M := "0010010"; --5
when "0110" => M := "0000010"; --6
when "0111" => M := "1111000"; --7
when "1000" => M := "0000000"; --8
when "1001" => M := "0010000"; --9
when others => M := "XXXXXXXX";
end case;
return M;
end function dekoderekod;
end architecture RTL;
```

```

begin
process(clk)
variable x: integer range 0 to 500000;
begin
    if (clk'event and clk='1') then
        if (x=500000) then
            x:=0;
            mili0<='1';
        else
            x:=x+1;
            mili0<='0';
        end if;
    end if;
end process;
process(clk,wr,rd,chipselect,we) is
variable rejestr: std_logic_vector(31 downto 0);
variable rejestr1: std_logic_vector(31 downto 0);
alias msek0 : std_logic_vector(3 downto 0) is rejestr1(3 downto 0);
alias msek1 : std_logic_vector(3 downto 0) is rejestr1(7 downto 4);
alias sek00 : std_logic_vector(3 downto 0) is rejestr1(11 downto 8);
alias sek01 : std_logic_vector(3 downto 0) is rejestr1(15 downto 12);
alias min00 : std_logic_vector(3 downto 0) is rejestr1(19 downto 16);
alias min01 : std_logic_vector(3 downto 0) is rejestr1(23 downto 20);
alias godz0 : std_logic_vector(3 downto 0) is rejestr1(27 downto 24);
alias godz1 : std_logic_vector(3 downto 0) is rejestr1(31 downto 28);
begin
if (clk'event and clk='1') then
    if (chipselect='1') then
        if (wr='1') then
            rejestr(31 downto 0) := we(31 downto 0);
        end if;
        if (rd='1') then
            rejestr1 := rejestr;
        end if;
    end if;
if mili0='1' then
    msek0 := msek0 + 1;
    if msek0 = 10 then
        msek0:="0000";
        msek1:=msek1 + 1;
    end if;
    if msek1 = 10 then
        msek1:="0000";
        sek00:=sek00 + 1;
    end if;
end if;

```

```

if sek00 = 10 then
    sek00:="0000";
    sek01:=sek01 + 1;
end if;
if sek01 = 6 then
    sek01:="0000";
    min00:=min00 + 1;
end if;
if min00 = 10 then
    min00 := "0000";
    min01 := min01 + 1;
end if;
if min01 = 6 then
    min01 := "0000";
    godz0 := godz0 + 1;
end if;
if godz0 = 10 then
    godz0 := "0000";
    godz1 := godz1 + 1;
end if;
if (godz1 = 2 and godz0 = 4) then
    godz0 := "0000";
    godz1 := "0000";
end if;
end if;
end if;
MS_0<=dekoder(msek0);
MS_1<=dekoder(msek1);
SS_0<=dekoder(sek00);
SS_1<=dekoder(sek01);
MM_0<=dekoder(min00);
MM_1<=dekoder(min01);
HH_0<=dekoder(godz0);
HH_1<=dekoder(godz1);
end process;
end RTL;

```

## 5. Kod programu testującego

```
#include <stdio.h>
```

```
#include <io.h>
```

```
main()
```

```
{
```

```
    IOWR(0x00000008,0,591418168);    //zapis dla godz 23:40:53:38
```

```
    IORD(0x00000008,0);                //odczyt
```

```
    printf("Ustawiono zegar!\n");
```

```
}
```