

WOJSKOWA AKADEMIA TECHNICZNA
im. Jarosława Dąbrowskiego

**LABORATORIUM UKŁADÓW
PROGRAMOWALNYCH I SPECJALIZOWANYCH**

SPRAWOZDANIE

Temat: *Projekt notesu elektronicznego w języku VHDL przy użyciu
układów firmy Altera z serii Cyclone II*

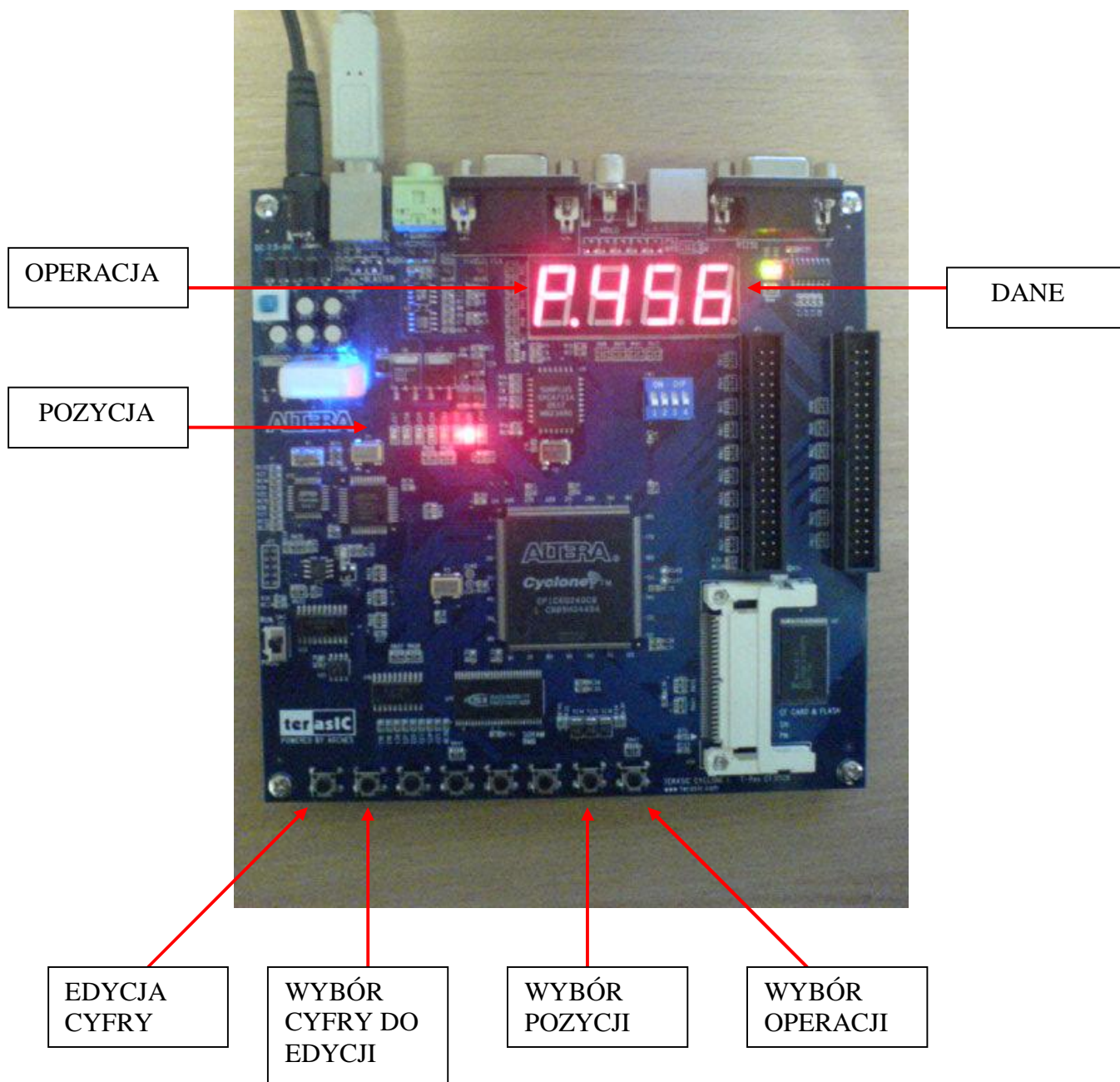
Wykonali:
Tomasz Trzciniński
Dariusz Wojtczuk
Gr. E4B2S0

Niniejsze sprawozdanie zawiera opis projektu, jakim jest „notes elektroniczny” do przechowywania krótkich informacji w postaci liczb.

Projekt wykonany jest w języku VHDL przy użyciu układów firmy *Altera* z serii *Cyclone II* z wykorzystaniem środowiska projektowego *Quartus*.

Wykorzystana płytko posiada możliwości współpracy z innymi urządzeniami, jednak projekt został ograniczony do wykorzystania możliwości samej płytki. Dla potrzeb notesu wykorzystany został czterocyfrowy wyświetlacz siedmiosegmentowy, co znacznie ogranicza możliwość wyświetlania zapisanych w naszym notesie danych. Po pierwsze ogranicza ilość wyświetlanych jednocześnie znaków na wyświetlaczu siedmiosegmentowym (nie ma możliwości wyświetlenia wszystkich znaków alfanumerycznych). Na płytce mamy do dyspozycji 8 przycisków, które posłużą do obsługi notesu oraz 8 diod LED, które będą wskazywały pozycje w notesie. Ogólnie rzecz biorąc wykonany notes ma za zadanie przechowywanie przez użytkownika pewnych danych, czyli potrzebne są następujące operacje: wpisywania do notesu (d – dodaj), odczytywanie zapisanych informacji (P – pokaż , oraz kasowanie informacji niepotrzebnych (U – usuń). Która z operacji jest aktualnie aktywna sygnalizuje litera wyświetlana na pierwszym od lewej wyświetlaczu.

Jak widać na poniższym zdjęciu notes elektroniczny wykorzystuje tylko 4 przyciski, 8 diod LED oraz wyświetlacz segmentowy.



WYBÓR OPERACJI

Wskazany na poniższym zdjęciu przycisk służy do wyboru operacji jaką chcemy w danym momencie wykonywać (odczytywać, dodawać czy usuwać).



WYBÓR POZYCJI

Drugi od prawej przycisk służy do wyboru pozycji w notesie. Do notesu można wpisać tyle pozycji ile jest diod na płytce gdyż zapalona dioda sygnalizuje pozycje, która jest aktualnie wyświetlana (dlatego do naszego notesu można wpisać 8 pozycji gdyż tyle jest dostępnych diod).



DODAWANIE DANYCH LUB EDYCJA

Aby wpisywać dane do notesu należy ustawić tryb dodawania (pojawia się litera „d”). Dodawanie zostało zrealizowane przy użyciu dwóch przycisków (skrajnych z lewej strony). Pierwszym wybieramy, która z trzech cyfr chcemy zmienić a drugim edytujemy cyfrę poprzez zwiększanie o 1 aż do uzyskania żądanej.



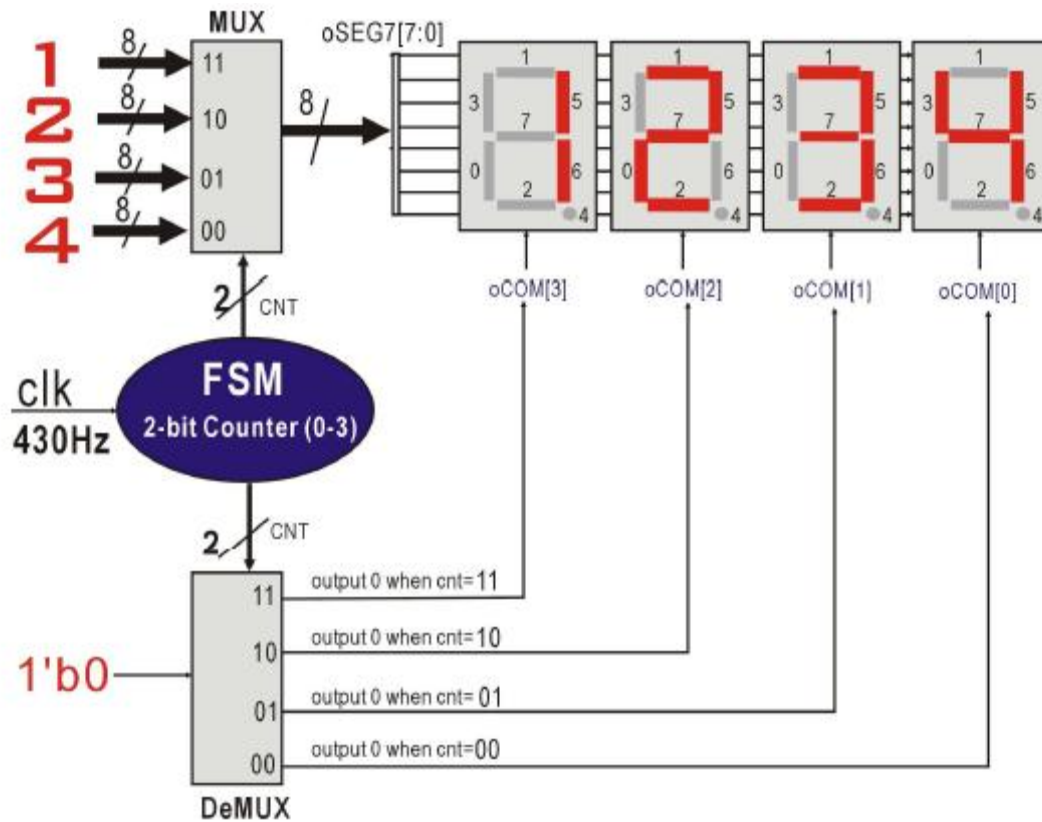
USUWANIE DANYCH

Kolejną operacją jest usuwanie zapisanych danych. W tym celu wystarczy przejść do wybranej pozycji w notecie, którą chcemy skasować, a następnie przyciskiem zmiany operacji wybrać U – usuń, pozycja zostanie automatycznie skasowana.



WYŚWIETLANIE

Wyświetlanie zostało zrealizowane według poniższego schematu.



Szczegółowy opis zarówno wyświetlania jak i innych funkcji został umieszczony poniżej w postaci opisu kodu VHDL.

RAPORT Z KOMPILACJI

Flow Summary	
Flow Status	Successful - Mon Jan 15 17:07:00 2007
Quartus II Version	5.0 Build 148 04/26/2005 SJ Web Edition
Revision Name	notes
Top-level Entity Name	notes
Family	Cyclone
Device	EP1C6Q240C8
Timing Models	Final
Met timing requirements	Yes
Total logic elements	690 / 5,980 (11 %)
Total pins	29 / 185 (15 %)
Total virtual pins	0
Total memory bits	0 / 92,160 (0 %)
Total PLLs	0 / 2 (0 %)

OPIS W JEZYKU VHDL

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity notes is

    port(

        --przyciski na klawiaturze
        klawiatura: in std_logic_vector(7 downto 0);

        --diody na płytce określające pozycje wyświetlanego numeru
        diody :out std_logic_vector(7 downto 0);

        --jedna pozycja wyświetlacza
        segment :inout std_logic_vector(0 to 6);

        --zegar 50Mhz
        clk :in std_logic;

        --wektor służący do wyświetlania
        com : inout std_logic_vector(0 to 3):="1110";

        --sygnał zapalający kropkę
        kropka : inout std_logic:='1'
            );
    end notes;

architecture a1 of notes is

    --powołanie nowego typu zawierającego jeden 3 cyfrowy numer
    type linia is array (0 to 2) of std_logic_vector(0 to 6);

    --powołanie nowego typu zawierającego zbiór numerów
    type tablica is array(0 to 7) of linia; signal lin: linia;

    --powołanie nowego sygnału typu tablica (zaalokowanie tablicy)
    signal tab: tablica;

begin

    --proces reaguje na zmiany sygnału zegarowego i
    --naciśnięcie klawisza na klawiaturze
    process(klawiatura,clk)
```



```

--powołanie zmiennych pomocniczych
variable wiersz : integer:=0;
variable kol : integer:=0;
variable dzielnik: std_logic_vector(16 downto 0);
variable operacja: std_logic_vector(0 to 6);
variable com1 : std_logic_vector(1 downto 0):="00";
variable dzialanie :std_logic_vector(1 downto 0):="00";
variable wypelnienie :integer:=0;
variable lin1: linia;
variable tabl: tablica;
variable kod: std_logic_vector(6 to 0);

begin

--układ zmieniający rodzaj wykonywanej operacji (licznik mod2)
if rising_edge(klawiatura(0)) then -- wybor operacji

    if dzialanie <= "01" then
        dzialanie:=dzialanie+1;
    else
        dzialanie:="00";
    end if;
end if;

--opis działania operacji wyświetlania
if dzialanie = "00" then operacja := "0010010";

--opis działania operacji dodawania
elsif dzialanie = "01" then operacja := "0101000";

if rising_edge(klawiatura(7))
    then
--odczyt danych z tablicy na pozycji równej pozycji zmiennej
--wiersz
        tabl:=tab;
        lin1:=tabl(wiersz);

--inkrementacja wartości wyświetlanej pozycji w kodzie
siedmio-segmentowym;

if lin1(kol)="0000001" then lin1(kol):="1111001";
tab(wiersz)<=lin1;
elsif lin1(kol)="1111001" then lin1(kol):="0001010";
tab(wiersz)<=lin1;
elsif lin1(kol)="0001010" then lin1(kol):="1001000";
tab(wiersz)<=lin1;

```

```

elsif lin1(kol)="1001000" then lin1(kol):="1110000";
tab(wiersz)<=lin1;
elsif lin1(kol)="1110000" then lin1(kol):="1000100";
tab(wiersz)<=lin1;
elsif lin1(kol)="1000100" then lin1(kol):="0000100";
tab(wiersz)<=lin1;
elsif lin1(kol)="0000100" then lin1(kol):="1011001";
tab(wiersz)<=lin1;
elsif lin1(kol)="1011001" then lin1(kol):="0000000";
tab(wiersz)<=lin1;
elsif lin1(kol)="0000000" then lin1(kol):="1000000";
tab(wiersz)<=lin1;
elsif lin1(kol)="1000000" then lin1(kol):="1111111";
tab(wiersz)<=lin1;
elsif lin1(kol)="1111111" then lin1(kol):="0000001";
tab(wiersz)<=lin1;
end if;
end if;

```

--opis działania operacji usuwania

```

elsif dzialanie = "10" then operacja := "0100001";

```

--wygaszenie wyświetlacza na trzech pozycjach

```

    lin1(0):="1111111";
    lin1(1):="1111111";
    lin1(2):="1111111";
    tab(wiersz)<=lin1;

```

```

end if;

```

-- przewijanie do kolejnej pozycji numeru

```

if rising_edge(klawiatura(1)) then
    if wiersz <= 6 then
        wiersz:=wiersz+1;
    else
        wiersz:=0;
    end if;
end if;

```

```

end if;

```

-- przewijanie do kolejnej pozycji w numerze

```

if rising_edge(klawiatura(6)) then

    if kol <= 1 then
        kol:=kol+1;
    else
        kol:=0;
    end if;
end if;

```

```

end if;

```

```

--realizacja operacji wyświetlania
if rising_edge(clk) then

    --dzielenie przez 131071
    if dzielnik="111111111111111111" then

--realizacja wyświetlania numerów w tablicy na wyświetlaczu
if com="1110" then lin1:=tab(wiersz); segment<=lin1(0);
com<="1101"; kropka<='1';
elsif com="1101" then lin1:=tab(wiersz); segment<=lin1(1);
com<="1011"; kropka<='1';
elsif com="1011" then lin1:=tab(wiersz); segment<=lin1(2);
com<="0111"; kropka<='1';
else segment<=operacja; com<="1110"; kropka<='0';
end if;

    --realizacja wyświetlania pozycji numeru na diodach
    if wiersz=0 then diody<="00000001";
    elsif wiersz=1 then diody<="00000010";
    elsif wiersz=2 then diody<="00000100";
    elsif wiersz=3 then diody<="00001000";
    elsif wiersz=4 then diody<="00010000";
    elsif wiersz=5 then diody<="00100000";
    elsif wiersz=6 then diody<="01000000";
    else diody<="10000000";
    end if;

    end if;
    dzielnik:=dzielnik+1;
end if;
end process
end al;

```