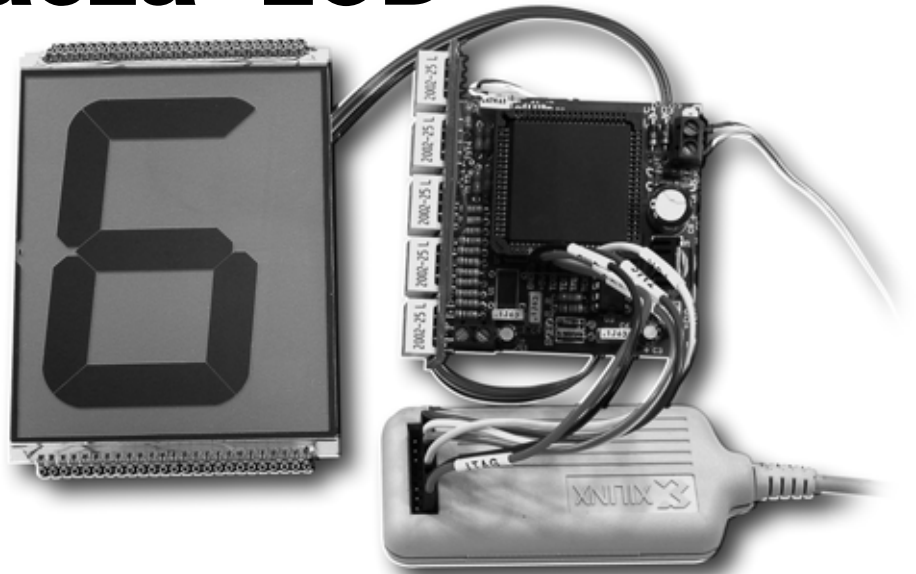


Dekoder-sterownik 7-segmentowego wyświetlacza LCD w VHDL



Do podjęcia tego tematu zachęcił mnie post na pl.misc.elektronika, w którym jeden z grupowiczów zwrócił uwagę na pewien problem przy opisie układu w języku VHDL - przypisanie jednobitowego sygnału do wielobitowego wektora. Przy okazji przedstawiamy przydatny, aczkolwiek niezbyt często stosowany w praktycznych aplikacjach blok funkcjonalny, który - dzięki uniwersalnemu językowi opisu - można traktować jak klasyczny blok IP (Intellectual Property core).

Rekomendacje: moduł IP przydatny we wszelkiego rodzaju aplikacjach, w których zastosowano klasyczne, 7-segmentowe wyświetlacze LCD.



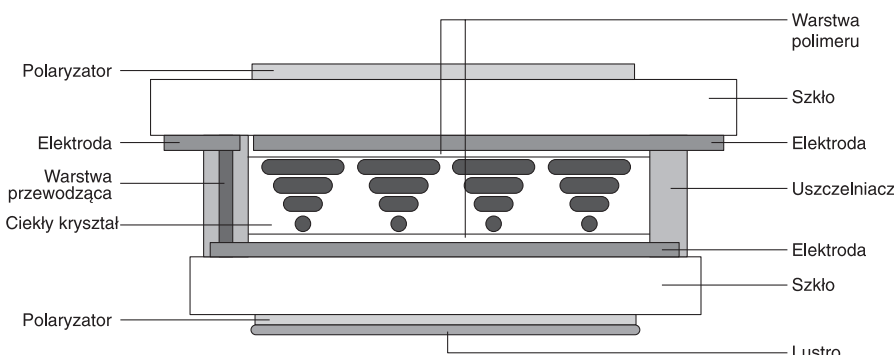
Dekoder-sterownik opisany w artykule był kilkakrotnie sygnalizowany przez Czytelników, ale ze względu na jego podobieństwo do scalonego dekodera-sterownika 7-segmentowych wyświetlaczy LED typu 4543 (układ z serii CMOS) nie zamierzałem się nim zajmować. Niesłusznie - w budowie tego układu tkwi „kruczek“, który sprawia sporo kłopotów początkującym konstruktorom korzystającym z VHDL.

Wyświetlacze LCD-TN

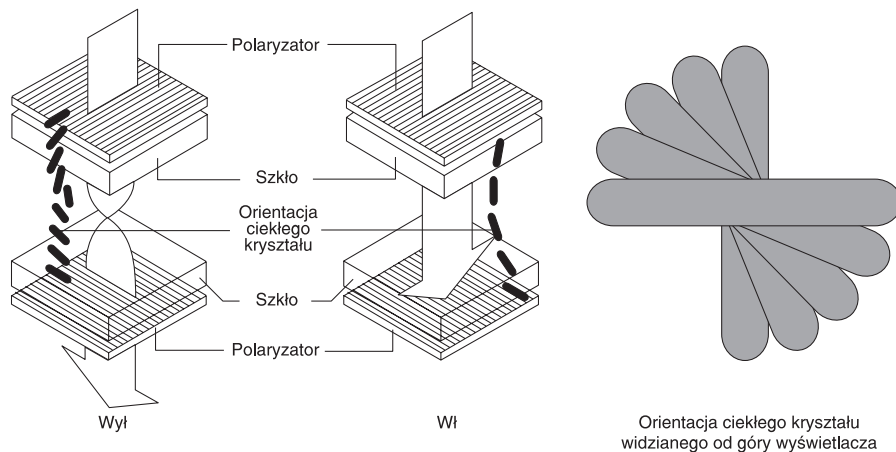
LCD to akronim pochodzący od *Liquid Crystal Display*, co oznacza wyświetlacz ciekłokrystaliczny. Ciekłe krysztale posiadają

fizyczne własności zarówno ciała stałego, jak i płynu. Jedną z ich właściwości jest to, że zmieniają one swoje położenie w zależności od przyłożonego napięcia (w zasadzie od natężenia pola elektrycznego pomiędzy okładkami „kondensatora“, w którym dielektrykiem jest ciekły kryształ).

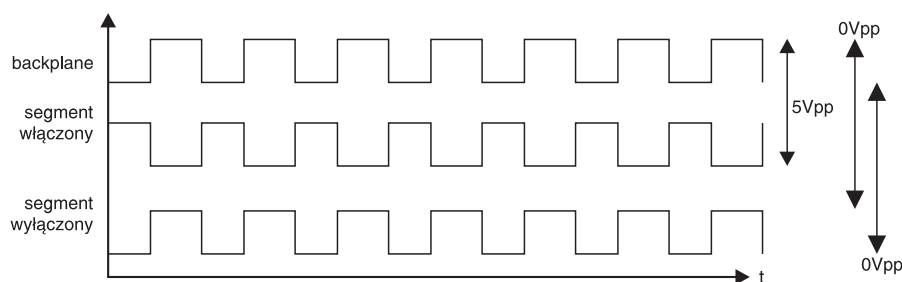
Ciekłe krysztale w wyświetlaczach stosuje się najczęściej w postaci cienkiej warstwy umieszczonej między dwiema szklanymi płaszczyznami, które spełniają rolę elektrod (rys. 1). Oddziaływania powierzchniowe między częsteczkami ciekłego kryształu a materiałem elektrod powodują powstanie określonego ułożenia (tekstury) cząsteczek w warstwie ciekłokrystalicznej. Spośród wielu znanych tekstur molekularnych ciekłych kryształów najważniejsze są: tekstura planarna, charakteryzująca się równoległym ułożeniem cząstek ciekłego kryształu w stosunku do płaszczyzny elektrod i tekstura homeotropowa - o prostopadłym układzie cząsteczek. Bardzo ważną odmianą tekstury planarnej jest konfiguracja TN (*Twisted Nematics*), czyli tzw. skręconego nematyka, która charakteryzuje się skręceniem osi



Rys. 1. Przekrój ilustrujący budowę klasycznego wyświetlacza LCD



Rys. 2. Zasada działania wyświetlacza LCD-TN



Rys. 3. Włączenie segmentu wyświetlacza LCD wymagaysterowania elektrody backplane sygnałem o przeciwnej fazie niż sygnału sterującego segment

cząstek równolegle ułożonych przy obu powierzchniach o kąt 90 lub 45° (rys. 2). Strukturę skręconego nematyka można uzyskać ze zwyczajnych prętopodobnych cząsteczek. Wymaga to jednak specjalnej obróbki powierzchni płytek szklanych. Ich powierzchnię pokrywa się cienką, przezroczystą warstwą polimeru i poleruje w jednym kierunku. Prętopodobne cząsteczki ustawiają się zgodnie z kierunkiem polerowania. Jeżeli drugą szklaną płytkę wypolerowaną analogicznie przekreścimy o 90° i między te płytki wprowadzimy ciekły kryształ nematyczny, to przyjmie on strukturę skręconą o 90°. Taka struktura skręca płaszczyznę polaryzacji światła o 90°. Jeżeli powierzchnie płytek szklanych pokryjemy cienką, przezroczystą warstwą przewodzącą, to mamy możliwość wytworzenia pola elektrycznego pomiędzy okładzinami „kondensatora”. Prętopodobne cząsteczki będą się ustawiały zgodnie z liniami sił pola elektrycznego, prostopadle do powierzchni szkła.

Jeżeli przygotowane płytki z warstwą przewodzącą (rys. 1),

warstwą orientującą i warstwą skręconego nematyka wstawimy między skrzyżowane polaryzatory (rys. 2), to światło białe przejdzie przez górny polaryzator jako spolaryzowane, w warstwie skręconego nematyka kierunek drgań ulegnie skręceniu o 90° i padając na drugi skrzyżowany polaryzator, będzie miało kierunek drgań zgodny z dolnym polaryzotorem. Zostanie

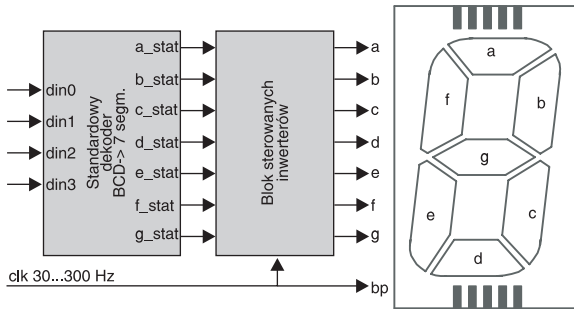
więc przepuszczone. Po przyłożeniu napięcia, cząsteczki białego kryształu ustawią się prostopadle do powierzchni szkła i polaryzatorów. Nie będą skręcały płaszczyzny polaryzacji światła. Zostanie więc ono zatrzymane przez drugi, dolny polaryzator.

Właściwości ciekłych kryształów pozostają w ścisłym związku z ich budową chemiczną. Najogólniej można powiedzieć, że są to substancje, których cząsteczki o wydłużonych kształtach mają charakter polarny, tzn. w ich molekułach występuje nierównomierny rozkład ładunku elektrycznego. Cząsteczki o takich właściwościach będą reagowały na pole elektryczne.

Współcześnie produkowane wyświetlacze LCD, ze względu na słabą odporność stosowanych ciekłych kryształów na długotrwałą polaryzację napięciem stałym (polem elektrycznym o stałym, dość dużym natężeniu i kierunku), wymagają sterowania impulsowego. Polega ono na przyłożeniu do wspólnej elektrody wszystkich segmentów (*backplane*) sygnału prostokątnego o częstotliwości 30...300 Hz i napięciu o wartości 2...5 V. Maksymalna wartość napięcia stałego polaryzującego segmenty zazwyczaj nie może być większa od 50 mV. Wyświetlacze LCD-TN są stosunkowo wolne - w temperaturze 25°C czas włączania segmentu wynosi co najmniej 60 ms, natomiast czas wyłączenia zazwyczaj nie jest krótszy niż 80 ms.

Dzieło przypadku

Ciekle kryształy, podobnie do większości wynalazków, odkryto przez przypadek. W 1888 roku austriacki botanik Friedrich Reinitzer badał, jaką rolę odgrywa w roślinach cholesterol. W jednym z eksperymentów wystawił on badany materiał na działanie ciepła, przy czym odkrył, że kryształ w temperaturze 145°C staje się mętnym płynem, zaś przy 179°C przechodzi całkowicie w stan płynny. W dwa lata później, w Karlsruhe, niemiecki fizyk Otto Lehman wprowadza pojęcie: ciekły kryształ, stwierdzając ich specyficzną strukturę fazową. Ponownie do substancji o właściwościach ciekłych kryształów wrócono w latach sześćdziesiątych XX wieku, kiedy Amerykanin James L. Ferguson stwierdził, iż ciekłe kryształy można zastosować jako wskaźniki temperatury, zaś George Heilmeyer z Princeton odkrył zjawisko tzw. zakłóceń dynamicznych światła przechodzącego przez ciekły kryształ w polu elektrycznym. Jego publikacje z roku 1968 można uznać za początek technicznego zastosowania ciekłych kryształów.

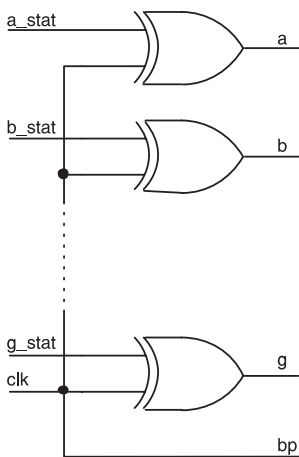


Rys. 4. Schemat blokowy dekodera BCD-> 7 segmentów

Widoczne („włączone“) segmenty wyświetlacza muszą być sterowane napięciem o takiej samej wartości, lecz fazie odwróconej o 180°. Do segmentów, które mają być niewidoczne („wyłączone“), należy przyłożyć napięcie o takiej samej amplitudzie i fazie, jak przyłożone do elektrody wspólnej. Na rys. 3 pokazano przebiegi sterujące dwoma segmentami wyświetlacza LCD: jeden z nich jest włączony, drugi wyłączony.

Opis projektu

Biorąc pod uwagę wymagania stawiane sterownikom przez wyświetlacze LCD-TN, musimy zaprojektować dekodery, na wyjściach którego będą występowały sygnały sterujące poszczególnymi segmentami wyświetlacza i - oczywiście - elektrodą wspólną (*backplane*). Dla przykładu skupimy się na zaprojektowaniu dekodera dla ciekłokrystalicznego wyświetlacza



Rys. 5. Jednym z możliwych rozwiązań sterowanego inwertera jest zastosowanie bramek Ex-OR

7-segmentowego. Jego schemat blokowy pokazano na rys. 4. Jak widać, składa się on z dwóch bloków:

- standardowego transkodera kodu BCD na kod wskaźnika 7-segmentowego (z wyjściami aktywnymi w stanie „1“),
- zespołu sterowanych inwerterów, które odwracają fazę sygnałów sterujących segmentami o 180° w stosunku do sygnału zasilającego *backplane* w przypadku, gdy wybrane segmenty mają być włączone, lub powtarzających sygnał *backplane* - w przypadku, gdy sterowane segmenty mają być wyłączone.

Narzędzia za darmo
Wszyscy liczący się producenci układów programowalnych udostępniają do celów edukacyjnych bezpłatne narzędzia do projektowania. Pomimo tego, że są udostępniane za darmo, ich funkcjonalność jest porównywalna z narzędziami komercyjnymi i bez trudu można za ich pomocą wykonać także zaawansowane projekty.

Sterowane inwertery można zrealizować na wiele sposobów, spośród których najodpowiedniejszym dla celów tego projektu są bramki Ex-OR połączone w sposób pokazany na rys. 5. Wejściowy sygnał *clk* połączony bezpośrednio z wyjściem *bp* jest podawany na jedno z wejść każdej z bramek Ex-OR, natomiast na każde drugie wejście jest podawany sygnał z dekodera BCD->7 segmentów (*a...g_stat*). W zależności od stanów na tych wejściach, na wyjściach bramek wystąpią przebiegi prostokątne będące w fazie z sygnałem *clk* (jeżeli na odpowiednim wejściu *a...g_stat* jest logiczne „0“) - wtedy tak sterowany segment nie jest włączony, lub w przeciwfazie (jeżeli na odpowiednim wejściu *a...g_stat* jest logiczna „1“), co powoduje włączenie segmentu.

Jak to zrobić w VHDL-u?

Przejdźmy do pokazania sposobu zapisania w języku VHDL dekodera zbudowanego zgodnie ze schematem blokowym z rys. 4. Jeden z możliwych wariantów opisu pokazano na list. 1.

W opisie architektury dekodera znajdującym się po deklaracji jednostki *entity* zastosowano instrukcję selekcyjną przypisania *with...select*, dla której szablonem wyrażen selekcyjnych jest 4-bitowy wektor *din*. Wektor *segn* jest zadeklarowany jako pomocniczy sygnał wewnętrzny (za pomocą deklaracji *signal segm: std_logic_vector(6 downto 0)*) i odpowiada on sygnałom *a...g_stat*, pokazanym na rys. 4 i 5. Wykorzystanie instrukcji *with...select* wymaga zdefiniowania odpowiedzi układu na wszystkie możliwe wartości wyrażenia selekcyjnego, w związku z czym jako ostatnią trzeba zastosować klauzulę „000000“ *when others* - bez niej kompilacja się nie uda. Założono, że wyświetlacz będzie wyłączony (żaden segment nie będzie widoczny) dla kodów wejściowych powyżej 9 (1001), czyli dekodery jest przeznaczony wyłącznie do wyświetlania cyfr z zakresu 0...9. Nic nie stoi na przeszkodzie, aby w razie takiej potrzeby samodzielnie zdefiniować znaki o kodach A...Fh (dziesiętnie 10...15), czego przykład pokazano na list. 2. Należy zwrócić uwagę, że pomimo jawnego zadeklarowania wszystkich wartości wektora

List. 1. Przykładowy opis w języku VHDL dekodera-sterownika 7-segmentowego wyświetlacza LCD

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity lcd_dek is port (
    din: in std_logic_vector(3 downto 0);
    clk: in std_logic;
    bp: inout std_logic;
    segm_o: out std_logic_vector(6 downto 0)
);
end lcd_dek;

architecture ar_dyn of lcd_dek is
    signal segm: std_logic_vector(6 downto 0);

begin
    with din select
    -- gfedcba
    segm <= "0111111" when "0000", -- 0
           "0000110" when "0001", -- 1
           "1011011" when "0010", -- 2
           "1001111" when "0011", -- 3
           "1100110" when "0100", -- 4
           "1101101" when "0101", -- 5
           "1111101" when "0110", -- 6
           "0000111" when "0111", -- 7
           "1111111" when "1000", -- 8
           "1101111" when "1001", -- 9
           "0000000" when others; -- wygaszenie

    segm_o(0) <= segm(0) xor bp;
    segm_o(1) <= segm(1) xor bp;
    segm_o(2) <= segm(2) xor bp;
    segm_o(3) <= segm(3) xor bp;
    segm_o(4) <= segm(4) xor bp;
    segm_o(5) <= segm(5) xor bp;
    segm_o(6) <= segm(6) xor bp;

    bp <= clk;

end ar_dyn;
    
```

List. 2. Przykładowa modyfikacja programu z list. 1, dzięki której dekodowane są także wektory wejściowe A...Fh

```

--      gfedcba
.....
"1111111" when "1000", -- 8
"1101111" when "1001", -- 9
"1110111" when "1010", -- kod 10 - znak "A"
"1111100" when "1011", -- kod 11 - znak "b"
"01111001" when "1100", -- kod 12 - znak "C"
"1011110" when "1101", -- kod 13 - znak "d"
"1111001" when "1110", -- kod 14 - znak "E"
"1110001" when "1111", -- kod 15 - znak "F"
"0000000" when others;
    
```

din nadal jest stosowana klauzula *when others*. Wynika to z faktu, że kompilatory rozpoznają także inne stany niż „0” i „1” (m.in. słabe zero i słabą jedynkę, stan wysokiej impedancji itp.) na wejściach *din*, dla których zadeklarowano typ *std_logic_vector*.

W dalszej części list. 1 znajduje się opis sterowanego inwertera, wykonanego na bramkach Ex-OR. W jego opisie pojawia się właśnie problem, o którym wspominałem na wstępie - chodzi o zapisanie sumy Ex-OR pomiędzy 7-bitowym wektorem składającym się z sygnałów sterujących segmentami wyświetlacza a 1-bitowym sygnałem *clk*. Niektóre syntezery VHDL (niezgodnie z obowiązującymi w VHDL normami) pozwalają na przeciążanie operatorów logicznych (w tym *xor*) w taki sposób, że możliwy byłby zapis *segm_o <= segm xor bp* - domyślnie jest implementowane siedem sum logicznych dla każdego sygnału z wektora *segm* z sygnałem *bp*. Standardowo operatory logiczne są przeznaczone do wykonywania operacji na wektorach o jednakowej długości, która może być większa od 1 bitu - przykład pokazano na list. 3.

List. 3. Alternatywny w stosunku do pokazanego na list. 1 sposób zapisu wielobitowej sumy Ex-OR

```

.....
bp_int(0) <= bp;
bp_int(1) <= bp;
bp_int(2) <= bp;
bp_int(3) <= bp;
bp_int(4) <= bp;
bp_int(5) <= bp;
bp_int(6) <= bp;
segm_o <= segm xor bp_int;
.....
    
```

List. 4. Alternatywny, najbardziej elastyczny, zapis wielobitowej sumy Ex-OR, ścisłego odpowiednika zapisu z list. 1

```

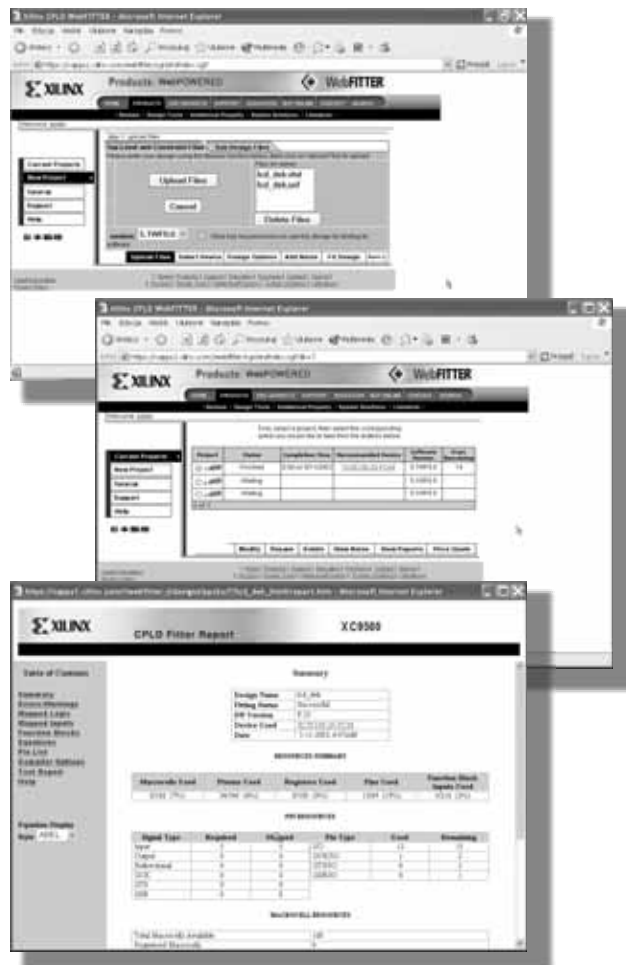
.....
mb_xor: for i in 0 to 6 generate
    segm_o(i) <= segm(i) xor bp;
end generate;
.....
    
```

Język VHDL uchodzi za jeden z bardziej skomplikowanych języków opisu sprzętu (HDL - *Hardware Description Language*). Pomimo dość rygorystycznych reguł formalnych obowiązujących podczas pisania programu, język ten charakteryzuje się znaczną uniwersalnością, co w praktyce oznacza, że dobrze przygotowany opis bloku funkcjonalnego będzie można „wbudować” zarówno w układy PLD, jak i ASIC pochodzące od różnych producentów. Niebagatelne znaczenie dla projektantów systemów cyfrowych ma fakt, że podzbiór języka VHDL, który jest obsługiwany przez programy do syntezy logicznej, jest niewielki i stosunkowo łatwy do nauczenia się.

Istotną cechą VHDL-a jest możliwość opisywania projektowanego układu na wiele sposobów m.in.: przepływowy (RTL - do którego sprowadzane są inne), behawioralny (opis zachowania się bloku w zależności od sygnałów zewnętrznych i wewnętrznych) czy też strukturalny (zwykle hierarchiczny opisujący budowę układu).

Niestety, taki zapis, podobnie do pokazanego na list. 1, nie jest zbyt wygodny, zwłaszcza gdy długość XOR-owanego wektora jest duża. Jakakolwiek zmiana w tej funkcji wymusza koniecz-

ność modyfikowania wielu miejsc programu, co jest czasochłonne i może być przyczyną błędu. Na list. 4 pokazano uproszczony zapis wielobitowej funkcji Ex-OR, w której długość wektora można



Rys. 6. Jednym z mniej znanych narzędzi projektowych jest zdalny kompilator WebFitter udostępniony bezpłatnie przez firmę Xilinx

swobodnie regulować za pomocą wartości parametru *i*. Utworzono ją za pomocą instrukcji powielania *generate*. Zapis pokazany na list. 4 jest ścisłym odpowiednikiem opisu zastosowanego na list. 1.

Implementacja

Dzięki zastosowaniu jako języka opisu sprzętu VHDL-a, z prezentowanego projektu mogą skorzystać użytkownicy dowolnej rodziny układów PLD. Dekoder zajmuje 8 makrokomórek w układach SPLD/CPLD (zmieści się więc nawet w układzie GAL22V10 lub GAL16V8 - po niewielkiej modyfikacji). Projekt był testowany na płycie ewaluacyjnej opisanej w EP9/2002, na której znajduje się

układ XC95108 (programowany w systemie CPLD ze 108 makrokomórkami). Dla wszystkich sygnałów jest niezbędne 13 wyprowadzeń (5 wejść i 8 wyjść).

Plik z opisem projektu (list. 1) skompilowano za pomocą bezpłatnych pakietów narzędziowych: Max+Plus II Student Edition (obsługuje VHDL), Quartus II 2.2 Web Edition - obydwie firmy Altera i WebPack ISE 4.2 firmy Xilinx.

Mało znanym, lecz równie skutecznym jak narzędzia stacjonarne (instalowane na komputerze użytkownika), jest zdalny internetowy kompilator firmy Xilinx - WebFitter (dostępny po zarejestrowaniu pod adresem: [*le=WebFITTER*\). Za jego pomocą można skompilować prezentowany projekt do postaci *.jed, czyli wynikowej, przeznaczonej do programowania układu docelowego. WebFitter generuje wszystkie niezbędne raporty, pozwala także kompilować projekty hierarchiczne. Widoki kilku okien pracującego WebFittera pokazano na rys. 6.](http://www.xilinx.com/xlnx/xil_prodcad_landingpage.jsp?tit-</p></div><div data-bbox=)

Piotr Zbysiński, AVT
piotr.zbysinski@ep.com.pl

Projekty dla systemu WebPack ISE 4.2 wraz z kodami źródłowym trzech wersji projektu są dostępne na stronie internetowej www.ep.com.pl w dziale Download>Dokumentacje. Publikujemy je także na płycie CD-EP7/2003B.