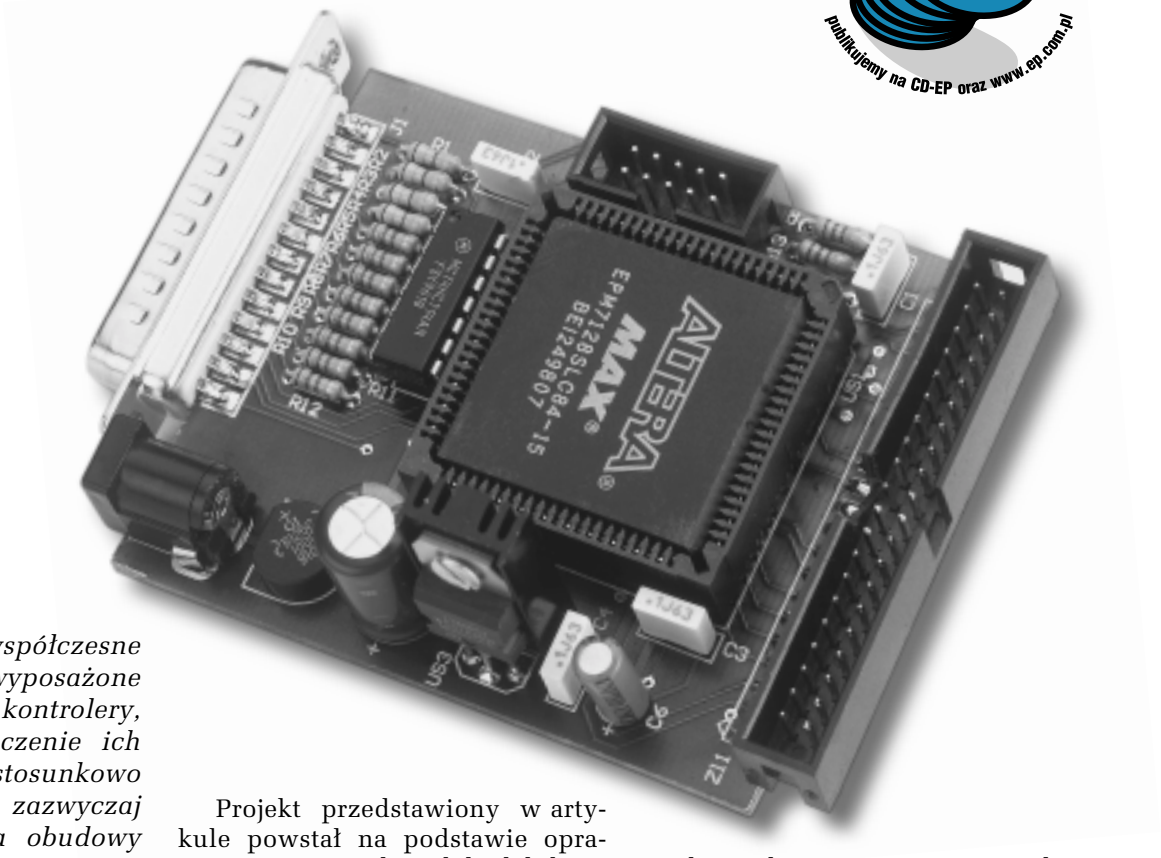
Interfejs IDE2LPT

AVT-5069



Wszystkie współczesne dyski twarde są wyposażone w wewnętrzne kontrolery, dzięki czemu dołączenie ich do komputera jest stosunkowo łatwe, choć zazwyczaj wymaga rozebrania obudowy komputera. W artykule przedstawiamy łatwy w wykonaniu interfejs, za pomocą którego można dołączyć dysk twardy do portu drukarkowego Centronics dowolnego komputera PC. Tak więc archiwizacja danych na starym, nieużywanym HDD może być nawet tańsza niż nagrywanie płyt CD-R.

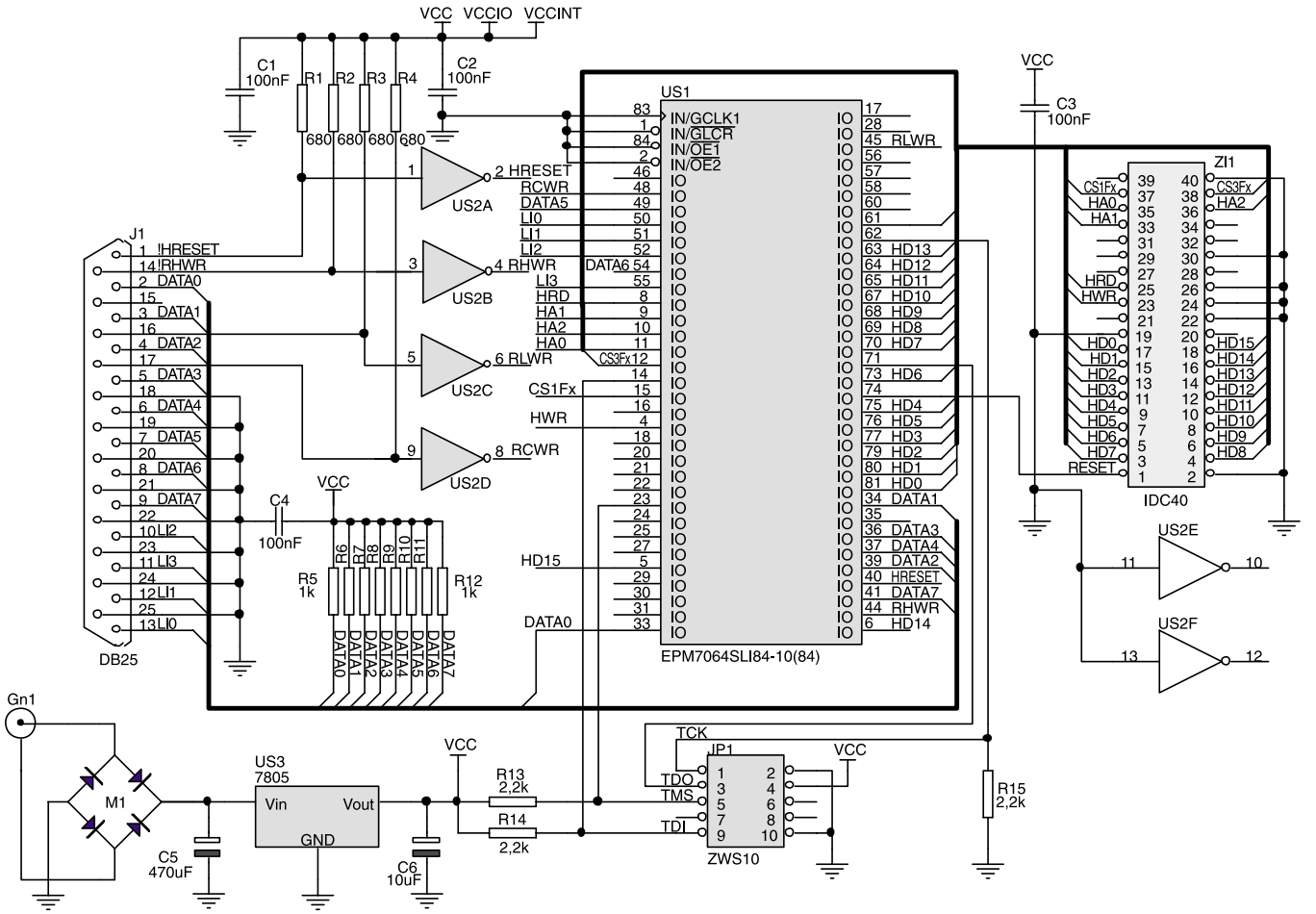
Fot. 1. Tak wyglądał protoplasta projektu prezentowanego w artykule

Projekt przedstawiony w artykule powstał na podstawie opracowania Leonida Slobodchikova z firmy AKA Curvex. Udostępnił on w Internecie, na stronie http://curvex.hypermart.net/ide2lpt/, ogólną dokumentację interfejsu IDE2LPT oraz - co najbardziej istotne - drivery dla DOS-a oraz Windows 95/98 (obydwa autorstwa Eugene Kuleshova) wraz z ich postacią źródłową. W oryginalnej wersji interfejs zbudowano w oparciu o standardowe układy TTL, których - w zależności od wykonania - było od 12 do 8 (**fot. 1**). My proponujemy nieco inne, znacznie bardziej nowoczesne podejście do sprzętowej części interfejsu: zaimplementujemy ją w całości w jednym układzie PLD (*Programmable Logic Devices*).

Przed dalszą częścią opisu musimy zwrócić uwagę Czytelników na fakt, że łatwość dołączenia dysku do dowolnego komputera PC z interfejsem Centronics jest okupiona niezbyt dużą szybkością wymiany danych pomiędzy dyskiem i komputerem. Z powodów konstrukcyjnych maksymalna szybkość pracy interfejsu w trybie bezpośredniego dostępu do rejestrów (bez wykorzystania klasycznych mechanizmów wymiany danych w trybie SPP) nie przekracza 200kB/s, a zazwyczaj wynosi ok. 130...160kB/s. Z tego powodu nie ma sensu wykorzystywania jako zewnętrznego dysku obecnie produkowanych, szybkich dysków ATA-2/3 - ich możliwości będą w takiej aplikacji marnowane.

Opis układu

Schemat elektryczny interfejsu wykonanego z układem CPLD firmy Altera EPM7064 (US1) pokazano na **rys. 2**. Ponieważ na podstawie schematu elektrycznego interfejsu trudno jest wywnioskować jak działa interfejs, pomocny będzie schemat układu cyfrowego zrealizowanego wewnątrz US1, który pokazano na **rys. 3**. Schemat ten jest dokładną kopią schematu najnowszej wersji klasycznego interfejsu IDE2LPT, a narysowano go w edytorze schematów pakietu projektowego firmy Altera Max+Plus II (**rys. 4**). Rysowanie schematu oddającego wewnętrzną budowę układu PLD jest sposobem bardzo często stosowanym przez projektantów korzystających z Max+Plus II (także innych sys-



Rys. 2. Schemat elektryczny interfejsu

temów EDA dla układów PLD), przede wszystkim ze względu na czytelność i łatwość interpretacji takiego sposobu opisu. Nie jest to jednak jedyny możliwy sposób opisania struktury tego układu, co pokażemy w dalszej części artykułu.

Realizacja projektów na układach PLD nie wymaga praktycznie żadnych nakładów. Doskonałe narzędzia projektowe, w tym kompilatory VHDL są udostępniane bezpłatnie.

Oznaczenia sygnałów zastosowane w projekcie układu PLD są identyczne z oznaczeniami sygnałów umieszczonymi na schemacie elektrycznym. Transfer danych z dysku do komputera odbywa się w paczkach 4-bitowych poprzez port LI[3...0]. Linie te są dołączone do następujących linii wejściowych interfejsu Centronics (odpowiednio):

Numer bitu rejestru *Base+1*	Nazwa Centronics	Nazwa IDE2LPT
7	Busy	LI3
6	Select	LI2
5	PE	LI1
4	ACK	LI0

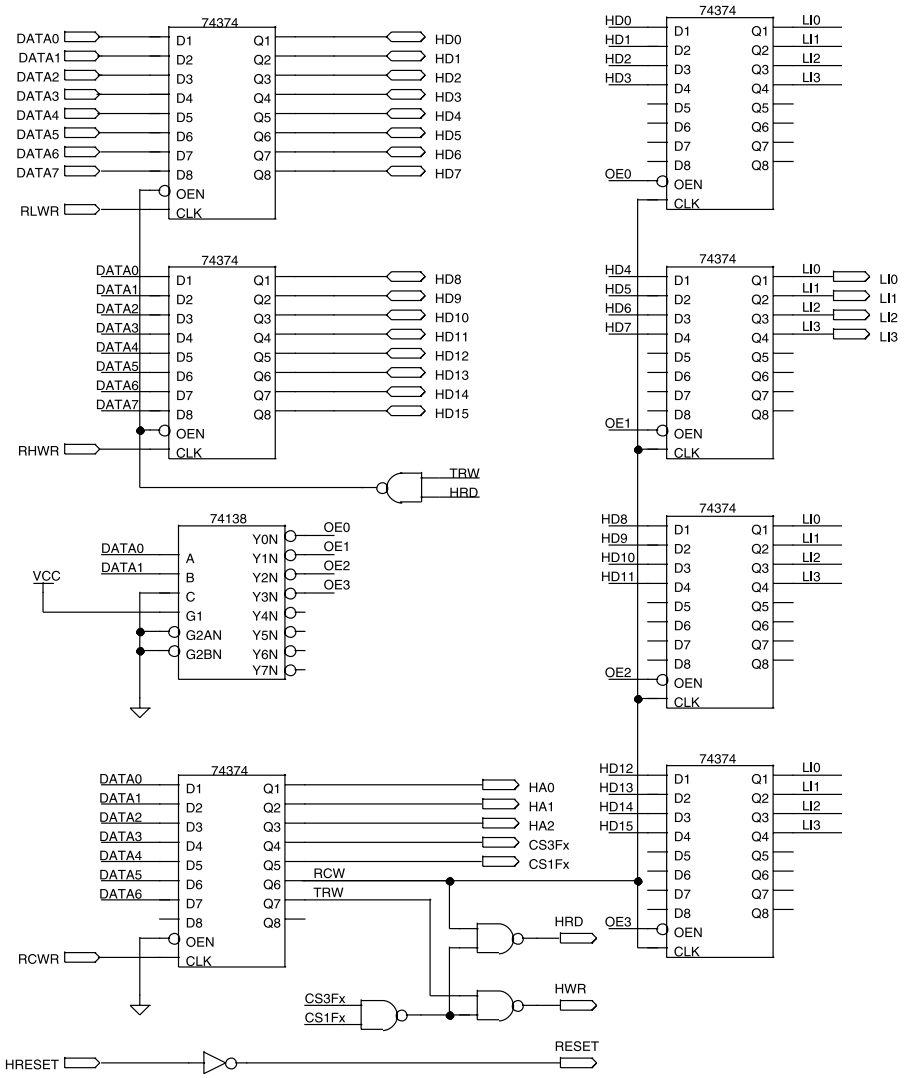
Transfer danych z komputera do dysku odbywa się poprzez rejestr 8-bitowy (ulokowany pod adresem *base=0x378/0x278*), który w złączu Centronics służy do przesyłania danych do drukarki. Sterowanie transmisją danych umożliwiają cztery sygnały pomocnicze, których przypisanie do bitów rejestru o adresie *base+2* pokazano poniżej:

Numer bitu rejestru *Base+1*	Nazwa Centronics	Nazwa IDE2LPT
3	Select in	RCWR
2	Init	RLWR
1	Auto Feed	HWR
0	Strobe	HRESET

Jak widać na rys. 3, wszystkie zastosowane rejestry są tego samego typu (odpowiedniki 74374). Ponieważ tylko dwa wbudowane w US1 rejestry są 8-bitowe, a wśród pozostałych jeden jest 7-bitowy i cztery są 4-bitowe, nasuwa się pytanie, czy taki projekt nie zajmie zbyt wiele zasobów układu US1. Jak pokazuje praktyka, w systemie Max+Plus II zastosowano doskonałe mechanizmy optymalizacyjne, w związku z czym nieużywane w projekcie fragmenty bloków funkcjonalnych zdefiniowanych przez użytkownika (w tym bufory trójstanowe na wyjściu rejestru konfiguracji - zapisywanego sygnałem RCWR) zostaną podczas kompilacji pominięte.

Interfejs wyposażono w lokalny stabilizator napięcia zasilającego US3 z mostkiem prostowniczym Graetza M1 na wejściu, dzięki czemu polaryzacja napięcia podawanego na Gn1 nie ma znaczenia.

W modelowym egzemplarzu interfejsu zastosowano układ US1 typu EPM7064S, który można programować w systemie bez konieczności stosowania programatora. Do programowania jest niezbędny tylko prosty interfejs nazywany przez firmę Altera ByteBlaster, którego



Rys. 3. Budowa wewnętrzna układu US1 po zaprogramowaniu

szczegółowe opisy wraz ze schematem można znaleźć w:
- książkach „Układy programowalne w praktyce“ (WKŁ2001/2002) i „Układy programowalne - pierwsze kroki“ (BTC2002) - na płytach CD-ROM dołączonych do książek dostępne są także wzory płytek drukowanych do programatora ByteBlaster i ByteBlasterMV,
- Internecie, pod adresami: http://www.altera.com/literature/ds/dsbytemv.pdf (opis najnowszej wersji ByteBlastera MV) http://www.altera.com/literature/ds/dsbyte.pdf (opis klasycznej wersji ByteBlastera) - obydwie są zamienne.

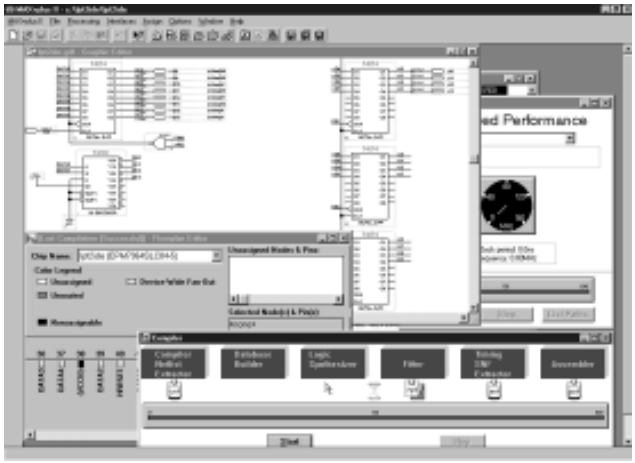
Podczas programowania układu US1 ByteBlaster musi być dołączony do złącza JP1.

WYKAZ ELEMENTÓW

Rezystory
R1...R4: 680Ω
R5...R12: 1kΩ
R13...R15: 2,2kΩ
Kondensatory
C1...C5: 100nF
C6: 10µF
Półprzewodniki
M1: dowolny mostek prostowniczy >200mA/50V
US1: EPM7064SLC84-10(84)
US2: SN74HCT14
US3: 7805 z radiatorem
Różne
Gn1: Gniazdo DC
J1: DB25M
ZI1: IDC40
JP1: ZWS10

Można także inaczej

Ponieważ praktycznie każdy producent układów PLD oferuje swój własny system projektowy, wymiana danych źródłowych pomiędzy nimi nie jest możliwa bez specjalnych zabiegów. Tak więc, wykonanie podobnego projektu z układem innego producenta niż Altera, zmusza projektanta do ponownego „budowania“ opisu struktury układu, co wiąże się m.in. z ryzykiem popełnienia błędu i w związku z tym ponownej symulacji.

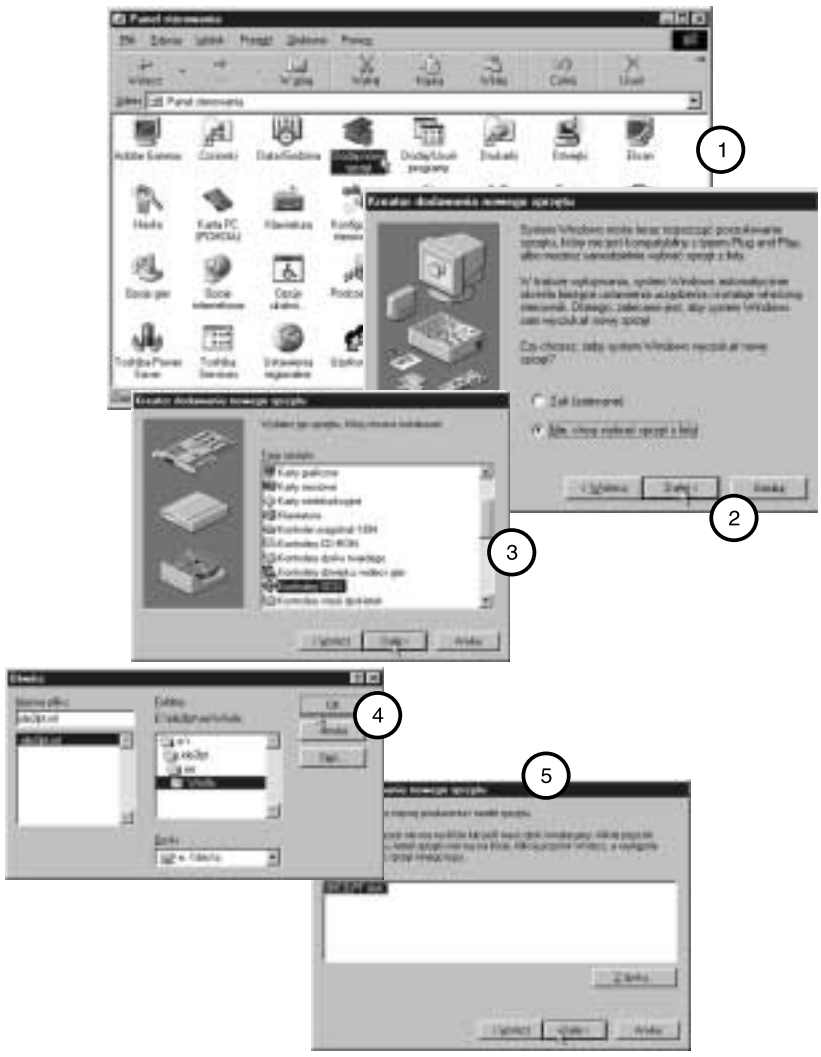


Rys. 4. Wygląd okna programu Max+Plus II

Rys. 5. Wygląd okna programu WebPack ISE

List. 1. Opis interfejsu IDE2LPT w języku VHDL (pominięto w nim niektóre fragmenty, komplet plików publikujemy na CD-EP6/2002B)

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity ide2lpt_main is port (
data: in std_logic_vector(7 downto 0);
hd: buffer std_logic_vector(15 downto 0);
li: out std_logic_vector(3 downto 0);
ha: buffer std_logic_vector(2 downto 0);
hrd, hwr, reset: buffer std_logic;
rlwr, rhwr, hreset, rcwr: in std_logic;
cs3fx, cs1fx, oe_dr: buffer std_logic
);
end ide2lpt_main;

architecture behav of ide2lpt_main is
component rejestr8 port (
   data:  in std_logic_vector(7 downto 0);
   wy: buffer std_logic_vector(7 downto 0);
   clk, oe: in std_logic
);
end component rejestr8;

component rejestr4 port (
   data:  in std_logic_vector(3 downto 0);
   wy: buffer std_logic_vector(3 downto 0);
   clk: in std_logic
);
end component rejestr4;

component dekoder port (
   we: in std_logic_vector(1 downto 0);
   oe: out std_logic_vector(3 downto 0)
);
end component dekoder;

signal trw, rcw, csxfx, dummy: std_logic;
signal sel, int0, int1, int2, int3:
             std_logic_vector(3 downto 0);

begin
reset <= not hreset;
csxfx <= cs3fx nand cs1fx;
hwr <= trw nand csxfx;
hrd <= rcw nand csxfx;
oe_dr <= hrd nand trw;
dr_low: rejestr8 port map (
   data => data,
   wy => hd(7 downto 0),
   oe => oe_dr,
   clk => rlwr
);

dr_high: rejestr8 port map (
   data => data,
   wy => hd(15 downto 8),
   oe => oe_dr,
   clk => rhwr
);

dek_sel: dekoder port map (
   we => data(1 downto 0),
   oe => sel
);

conf_reg: rejestr8 port map (
   data => data,
   wy(2 downto 0) => ha(2 downto 0),
   wy(3) => cs3fx,
   wy(4) => cs1fx,
   wy(5) => rcw,
   wy(6) => trw,
   wy(7) => dummy,
   oe => '0',
   clk => rcwr
);

hd_r0: rejestr4 port map (
   data => hd(3 downto 0),
   wy => int0,
   clk => rcw
);

.... - pominieto przypisania dwoch rejestrow!

hd_r3: rejestr4 port map (
   data => hd(15 downto 12),
   wy => int3,
   clk => rcw
);

with sel select
  li <= int0 when "0001",
        int1 when "0010",
        int2 when "0100",
        int3 when "1000",
      int0 when others;

end behav;
```

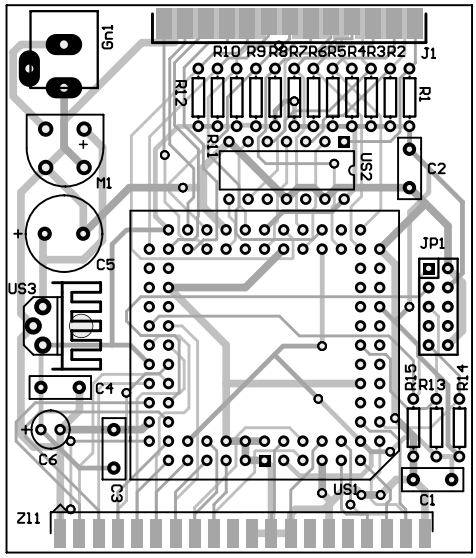


Rys. 6. W ten sposób instaluje się sterownik IDE2LPT w systemie Windows 98

Sytuację uprościło upowszechnienie się zestandardyzowanych języków opisu sprzętu (*HDL - Hardware Description Language*), jak VHDL czy Verilog. Kompilatory, programy syntezy logicznej i symulatory dla tych języków są udostępniane bezpłatnie, np.:

- Altera udostępnia pakiet Max+Plus II Baseline + Leonardo Spectrum lub (znacznie gorsza synteza VHDL) Max+Plus II Student Edition. Do kompilacji projektów opisanych w języku VHDL można także wykorzystać pakiet Quartus II, ale w wersji bezpłatnej nie obsługuje on układów z rodziny MAX7000S (programowanych w systemie).
- Xilinx udostępnia pakiet WebPack ISE, w ramach którego dostarczany jest niezły kompilator VHDL-a oraz symulator ModelSIM.

Także inni producenci układów PLD oferują bezpłatne narzędzia, ale ze względu na dostęp-



Rys. 7. Rozmieszczenie elementów na płytce drukowanej

ność układów skupimy się na dwóch wymienionych.

Aby ułatwić implementację interfejsu IDE2LPT z dowolnymi układami PLD został przygotowany jego hierarchiczny opis w języku VHDL. Na **list. 1** pokazano projekt *ide2lpt_main*, który zawiera opis interfejsu, przy czym należy zwrócić uwagę na odwołania do zewnętrznych plików, jak np.:

```
component rejestr4 port (
  data: in std_logic_vector(3
downto 0);
  wy: buffer std_logic_vector(3
downto 0);
  clk: in std_logic
);
end component rejestr4;
```

zawierających bloki funkcjonalne położone niżej w hierarchii. Dzięki takiemu zapisowi, wielokrotnie wykorzystywany w projekcie rejestr 4-bitowy został opisany tylko raz. Komplet plików źródłowych opublikowaliśmy na CD-EP6/2002B oraz na naszej stronie internetowej w dziale *Download*. Prezentowany projekt był kompilowany i weryfikowany za pomocą pakietu WebPack ISE 4.2 WP0 (**rys. 5**) oraz symulatora ModelSim XE 4.2. Projekt zmieścił się w układzie XC9572-LC84.

Oprogramowanie interfejsu

Działanie modelowego egzemplarza przetestowano na komputerach z systemem operacyjnym DOS oraz Windows 95/98. Z powodu braku odpowiednich sterowników, nie jest obecnie możliwa współpraca interfejsu z komputerami wyposażonymi w system operacyjny Windows w jakiejkolwiek wersji pochodnej NT.

W przypadku komputerów z DOS do pliku *config.sys* należy dopisać linię:

```
device=[ścieżka]\i2l4.exe
```

i zrestartować komputer. Program można uruchamiać z linii poleceń z dodatkowymi opcjami, np.:

- i2l4.exe **/t** uruchamia procedurę wykrywania dołączenia interfejsu do złącza Centronics,
- i2l4.exe **/l:xxxh** umożliwia określenie bazowego adresu portu Centronics (np. 378h),
- i2l4.exe **/h** wyświetla krótką instrukcję do programu,
- i2l4.exe **/l** powoduje włączenie obsługi dużych dysków LBA,
- i2l4.exe **/g:SEC:HEAD** umożliwia ustalenie własnej geometrii dysku twardego,
- i2l4.exe **/v** pozwala odczytać parametry dysku twardego dołączonego do interfejsu.

Sterownik można uruchamiać z wieloma parametrami jednocześnie.

W przypadku korzystania z komputera z zainstalowanym systemem operacyjnym Windows 95/98 należy zainstalować inny sterownik, napisany specjalnie dla Windows. Instalacja przebiega w typowy sposób - w *Panelu Sterowania* należy wybrać *Dodaj nowy sprzęt*. Kolejne kroki podczas instalacji pokazano na **rys. 6**.

Montaż i uruchomienie

Schemat montażowy dwustronnej płytki interfejsu pokazano na **rys. 7**. Podczas montażu należy zwrócić szczególną uwagę na sposób przylutowania gniazda Zl1 (40-stykowe gniazdo na kabel ATA) - położenie jego pierwszego styku wyraźnie zaznaczono na płytce drukowanej. Gniazda J1 i Zl1 są montowane na krawędzi płytki, natomiast pozostałe elementy na jej powierzchni.

Do uruchomienia interfejsu konieczny jest zasilacz dostarczający dwóch napięć: +5 i +12V, o wydajności prądowej wystarczającej do zasilenia dysku twardego. Zasilanie na wejście interfejsu należy podać z linii +12V, jest ono bowiem stabilizowane przez stabilizator wbudowany w interfejs. Pobór prądu przez sam interfejs nie przekracza 100mA, a średnio wynosi 80...90mA.

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

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: ***http://www.ep.com.pl/?pdf/czerwiec02.htm*** *oraz na płycie CD-EP06/2002B w katalogu* ***PCB***.