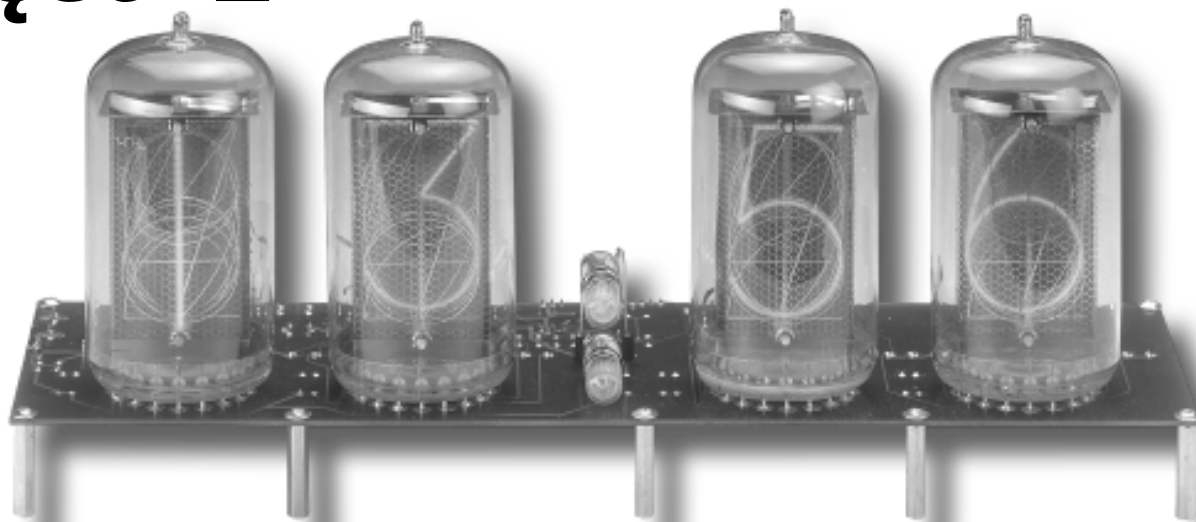


AVT Nixie Clock

VHDL i lampy, część 2



No tak, *Elektronika Praktyczna* wzięła się za „odgrzewanie kotletów”, pomyślą pewnie ci spośród naszych Czytelników, którzy pamiętają polską elektronikę z końca lat 80. Akceleracja, jakiej jesteśmy poddawani przez ostatnich 12 lat - chodzi głównie o dostęp do nowoczesnych podzespołów - spowodowała, że jeszcze niedawno traktowane pogardliwie lampy Nixie nabrały wyjątkowego „smaku”.

Pomysłem na ich zastosowanie dzielimy się z Wami w artykule.

Rekomendacje: projekt z popularnego na Zachodzie gatunku „Vintage Electronics”, czyli chwytające za serce (i zazwyczaj oko) praktyczne starocie. O tyle dopasowany do EP-owskiej doktryny nadążania za nowościami, że cała część cyfrowa została opisana w VHDL-u.

Montaż zegara

Płytką drukowaną zegara jest jednocześnie jego bazą mechaniczną, na której znajdują się wszystkie elementy elektroniczne, przełączniki, transformator, a także lampy Nixie oraz pomocnicza płytka drukowana, na której są zamontowane dwie neonówki spełniające rolę dwukropka. Ze względu na dużą średnicę szklanych baniek lamp wymiary płytki są dość znaczne (305 x 102 mm), ale jak pokazała praktyka, wygoda wynikająca ze zintegrowania wszystkich elementów zegara na jednej płycie rekompensuje dość wysoki koszt wykonania takiej płytki.

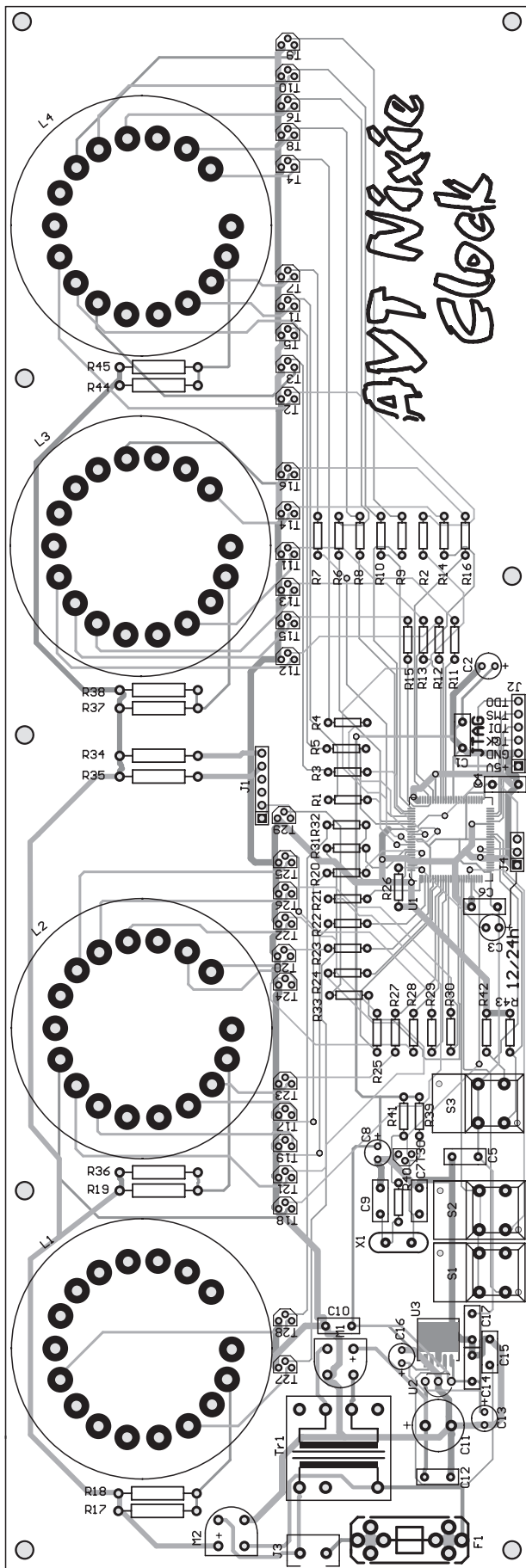
Schemat montażowy płytki głównej pokazano na **rys. 8**. W urządzeniu zastosowano dwa układy scalone przystosowane do montażu powierzchniowego - U1 oraz stabilizator napięcia 3,3 V - U3. O ile montaż stabilizatora nie sprawi żadnej trudności przeciętnie wyposażonemu elektronikowi, to do montażu układu U1 jest niezbędna lutownica z cienkim grotem oraz taśma rozlutownicza Wick. Można także podjąć próbę zamontowania tego układu w wyspecjalizowanej firmie zajmującej

się montażem elementów SMD, ale ze względu na trudną dostępność i wysoką cenę takiej usługi zachęcam do podjęcia próby samodzielnego przylutowania układu.

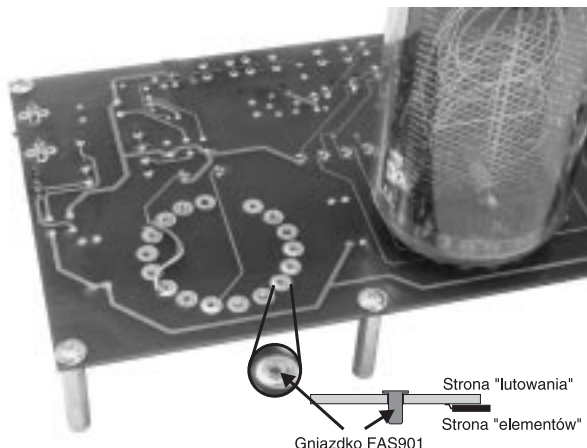
Zaczynamy od posmarowania spodniej części obudowy układu U1 odrobiną nieprzewodzącego kleju (doskonale sprawdził się butapren). Następnie kładziemy układ na miejscu dla niego przeznaczonym, zwracając oczywiście uwagę na właściwą orientację pierwszego wyprowadzenia układu. Delikatnymi ruchami należy możliwie dokładnie umieścić wyprowadzenia układu na polach lutowniczych, na co przez kilka minut pozwala elastyczność kleju zastosowanego do przymocowania układu. Teraz należy odczekać co najmniej godzinę, co spowoduje, że klej przestanie być elastyczny, a obudowa układu trwale przymocowana do płytki drukowanej.

W zależności od średnicy grota posiadanej lutownicy można zastosować dwa sposoby lutowania wyprowadzeń:

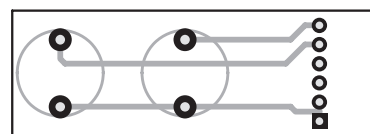
- jeżeli grot jest „gruby” (średnica końca większa niż 1,2 mm) najlepiej jest delikatnie przygrzewać i jednocześnie dociskać



Rys. 8. Rozmieszczenie elementów na płytce drukowanej (widok zmniejszony do 80%)



Rys. 9. Sposób wykonania podstawek do lamp



Rys. 10. Rozmieszczenie elementów na płytce drukowanej „dwukropka”

do pół lutowniczych po kolei grupy po kilka wyprowadzeń (ich liczba zależy od średnicy/szerokości grot) - ale bez podawania dodatkowego topnika! Jeżeli wyprowadzenia układu zostaną dobrze rozgrzane i dociśnięte, cyna znajdująca się na polach lutowniczych zwiąże się z nimi zapewniając dobre przewodzenie prądu.

- jeżeli mamy do dyspozycji „chudy” grot i oczywiście wprawna ręka można podjąć ryzyko podania odrobiny dodatkowego topnika. Zwiększy to prawdopodobieństwo dokładnego i trwałego przyłutowania wyprowadzeń, ale grozi powstaniem zwarcia pomiędzy sąsiednimi wyprowadzeniami układu. Lekarstwem na to jest miedziana taśma rozlutownicza (np. Wick), za pomocą której można odciągnąć nadmiar cyny, co w większości przypadków spowoduje usunięcie zwarcia.

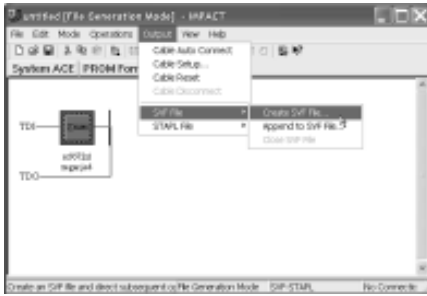
Kolejnym krokiem montażowym jest wlutowanie w płytke drukowaną gniazdek ze sprężystymi stykami (FAS901), które

spełniają rolę podstawki dla lamp Nixie. Ponieważ lampy są montowane od strony „lutowania” (czyli przeciwnej niż pozostałe elementy), to gniazdzka należy wlutować otworami od strony „lutowania”, jak to pokazano na rys. 9. Ze względów oszczędnościowych gniazdzka należy wlutować wyłącznie pod te wyprowadzenia lamp, które są wykorzystywane w aplikacji. Nie dotyczy to lampy L1 (są na niej wyświetlane dziesiątki godzin), której zaledwie trzy wyprowadzenia są wykorzystywane. Aby zapewnić jej odpowiednią stabilność mechaniczną należy zwiększyć liczbę gniazd, przy czym sensowne minimum zapewnia takie ich rozmieszczenie, jak w przypadku lampy L3 (są na niej wyświetlane dziesiątki minut).

Od strony „lutowania” montowane jest także gniazdo J1, które służy do zamontowania płytki drukowanej z dwiema neonówka-



Rys. 11. Widok okna konfiguracji programatora Uprog HS48 w trybie JTAG



Rys. 12. Widok okna konfiguracji programu iMPACT

mi (LP1 i LP2, wkręcone w oprawkę), które spełniają rolę dwukropka sygnalizującego odmierzenie czasu. Schemat montażowy tej płytki pokazano na rys. 10.

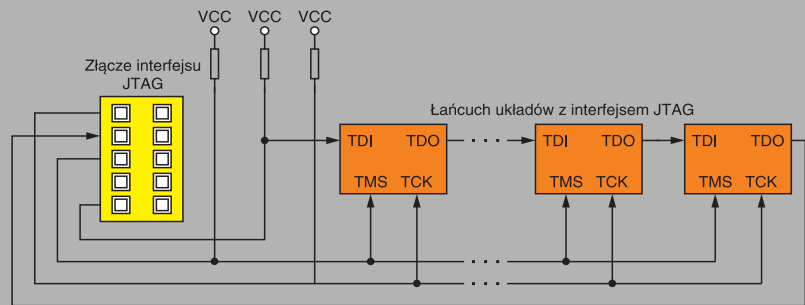
Montaż pozostałych elementów nie sprawi żadnych trudności Czytelnikom z choćby niewielkim doświadczeniem, nie będziemy się wobec tego nim zajmować.

Uruchomienie zegara

Układ U1 jest wyposażony w pamięć konfiguracji typu Flash, którą można wielokrotnie reprogramować w systemie za pomocą wbudowanego w układ interfejsu JTAG. Korzystanie z niego bardzo przypomina programowanie w systemie mikrokontrolerów PIC lub AVR, a do jego przeprowadzenia jest potrzebny mało skomplikowany interfejs portu drukarkowego, który opisaliśmy w EP4/2001. Przeprowadziłem także próby programowania układu U1 za pomocą programatora uniwersalnego UprogHS 48, który jest przystosowany do programowania w systemie układów wyposażonych w interfejs JTAG, co wymaga jedynie zastosowania 6-żyłowego kabla połączeniowego, którego końcówki są z jednej strony zaciskane w podstawie ZIF48 (rys. 11). Ponieważ projekt dla układu U1 powstał w oparciu o bezpłatne narzędzie WebPack ISE, za pomocą którego nie można wygenerować bezpośrednio pliku w formacie JAM-STAPL/SVF (które to formaty akceptuje UprogHS 48), trzeba dokonać konwersji pliku wynikowego w formacie JEDEC na SVF lub JAM-STAPL. Do tego celu służy program iMPACT (rys. 12), który należy przełączyć w tryb pracy *File Mode* (w menu opcja *Mode*). Format pliku wynikowego można wybrać w menu: *Output>SVF File* lub *Out-*

O JTAG-u słów kilka

Typowe dla JTAG-a procesy, tzn. testowanie i programowanie (konfigurowanie) układów z interfejsem JTAG przebiegają w podobny sposób. Najważniejsza różnica pomiędzy nimi polega na wykorzystaniu podczas testowania *rejestrów ścieżki krawędziowej*, a podczas programowania (konfigurowania) *rejestrów ISP*. Twórcy interfejsu JTAG przewidzieli możliwość jednoczesnego programowania lub testowania wielu układów. W takim przypadku należy je połączyć kaskadowo w łańcuch BST (ścieżki krawędziowej), jak to pokazano na poniżej. Każdy układ z interfejsem zgodnym ze standardem JTAG musi być wyposażony w 1-bitowy *rejestr obejściowy (bypass)*. To właśnie dzięki temu rejestrowi istnieje możliwość "operowania" na układach dowolnie wybranych z całego łańcucha.



Styk fizyczny interfejsu JTAG składa się z zaledwie 4 lub 5 pojedynczych, jednokierunkowych linii sygnałowych. Ciężar realizacji algorytmów sterujących wymianą informacji w łańcuchu JTAG jest rozłożony pomiędzy program sterujący pracą interfejsu oraz blok TAP wraz z elementami towarzyszącymi, w które wyposażono układy ISP. Dzięki temu typowe programatory-konfiguratorzy układów programowanych w systemie zawierają zazwyczaj tylko bufor zabezpieczające przed uszkodzeniem wyjścia portu równoległego komputera (LPT). Niektórzy producenci oferują także konwertery RS232 <->JTAG oraz nowocześniejsze konwertery USB<->JTAG, których jedną z zalet jest możliwość pracy *plug&play*.

put>STAPL File - w naszym przypadku obydwie formaty są równoważne.

Na fot. 13 pokazano ułożenie złącza JTAG na płytce zegara. Jak widać, wszystkie sygnały złącza J2 zostały wyraźnie opisane na płytce, co ułatwia dołączenie przewodów. Z lewej strony tego złącza widać także jumper oznaczony 12/24H, za pomocą którego można zmieniać tryb zliczania czasu przez zegar.

Programowanie układu U1 nieśie ze sobą pewne niebezpieczeństwo, ponieważ zegar nie jest odizolowany galwanicznie od sieci energetycznej. W przypadku większości klasycznych komputerów PC, ich obudowy w przypadku źle wykonanej instalacji elektrycznej także mogą znaleźć się na potencjale sieci, co grozi porażeniem oraz uszkodzeniem zarówno

komputera, jak i elementów zastosowanych w zegarze. Zagrożenia tego nie można bagatelizować, dlatego gorąco zachęcam do zastosowania do zasilania zegara (lub PC) transformatora bezpieczeństwa 220 VAC/220 VAC, można pokusić się także o zastąpienie go UPS-em, ewentualnie zamiast komputera stacjonarnego można zastosować notebooka zasilanego podczas programowania z wbudowanej baterii.

Zaprogramowanie układu U1 jest w zasadzie jedyną czynnością, jaką trzeba wykonać podczas uruchamiania zegara. W zależności od parametrów zastosowanego kwarcu X1 może okazać się konieczne zmodyfikowanie pojemności kondensatora C7 - częstotliwość pracy generatora z tranzystorem T30 powinna być możliwie bliska 32,768 kHz.

List. 3. Opis połączeń pomiędzy elementami projektu w języku VHDL

```

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

entity zegar is port (
jm: inout std_logic_vector(9 downto 0);
dzm: inout std_logic_vector(5 downto 0);
dzh: inout std_logic_vector(2 downto 1);
jh: inout std_logic_vector(9 downto 0);
ster_dk, szybko_wy, wolno_wy: out std_logic;
clk_ref, res, mode, clk_set, res_a,
set_a: in std_logic
);
end zegar;

architecture behavioral of zegar is
component licznik_jm port (
clk: in std_logic;
clk_o: out std_logic;
res: in std_logic;
outp: inout std_logic_vector(9 downto 0)
);
end component licznik_jm;

component licznik_dzm port (
clk: in std_logic;
clk_o: out std_logic;
res: in std_logic;
outp: inout std_logic_vector(5 downto 0)
);
end component licznik_dzm;

component licznik_h port (
clk: in std_logic;
res, mode: in std_logic;
outp_j: inout std_logic_vector(9 downto 0);
outp_dz: inout std_logic_vector(2 downto 1)
);
end component licznik_h;

component mux_2_1 port (
in0, in1, a: in std_logic;
out_mux: out std_logic
);
end component mux_2_1;

component presk port (
szybko, wolno, wy_1_min, wy_1Hz,
wy_4Hz: out std_logic;
res, clk: in std_logic
);
end component presk;

signal a, d, clk_a, clk_min,
clk_min_set: std_logic;
signal clk_0, clk_1, int_1Hz, int_4Hz,
nie_clk_a: std_logic;

begin
licz_j_minut: licznik_jm port map (
clk => clk_min_set,
clk_o => clk_0,
res => res,
outp => jm
);

licz_d_minut: licznik_dzm port map (
clk => clk_0,
clk_o => clk_1,
res => res,
outp => dzm
);

licz_h: licznik_h port map (
clk => clk_1,
res => res,
mode => mode,
outp_j => jh,
outp_dz => dzh
);

mux_czasu: mux_2_1 port map (
in0 => clk_min,
in1 => clk_set,
a => a,
out_mux => clk_min_set
);

mux_dwukropka: mux_2_1 port map (
in0 => int_1Hz,
in1 => int_4Hz,
a => a,
out_mux => ster_dk
);

preskaler: presk port map (
szybko => szybko_wy,
wolno => wolno_wy,
wy_1_min => clk_min,
res => res,
clk => clk_ref,
wy_1Hz => int_1Hz,
wy_4Hz => int_4Hz
);

-- przerzutnik D adresujący multiplexersy
-- wybor trybu pracy: ustawianie/czas
d_ff: process (clk_a, res)
begin
if res = '1' then
a <= '0';
elsif clk_a = '1' and clk_a'event then
a <= d;
end if;
d <= not a;
end process d_ff;

-- przerzutnik RS likwidujący drgania styków
clk_a <= '0' when res_a = '1' else
'1' when set_a = '1' else
clk_a;

nie_clk_a <= '1' when res_a = '1' else
'0' when set_a = '1' else
clk_a;

end behavioral;

```

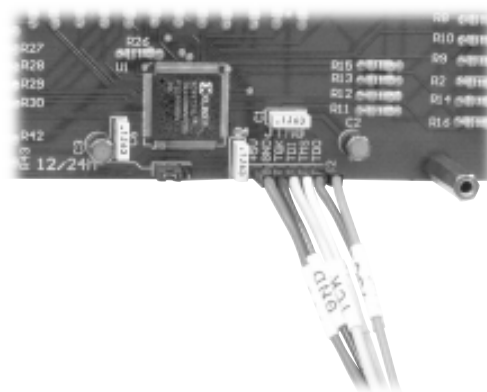
Kłopoty z generatorem

Wykonałem dwa egzemplarze zegara, podczas uruchamiania których okazało się, że generator sygnału wzorcowego jest bardzo czuły na parametry oscylatora kwarcowego X1. Praktycznie w obydwu przypadkach konieczne było wyselekcjonowanie kwarcu, który wzbudzał generator. Aby uniknąć problemów podczas uruchamiania powstał znacznie stabilniejszy generator, którego schemat elektryczny pokazano na **rys. 14**. Jest to - jak widać - klasyczny generator z niebuforowanym układem CMOS typu 4069. Sygnał z wyjścia generatora jest buforowany przez inwerter U1B, a pozostałe inwertery nie są wykorzystywane i mają wejścia zwarte do masy. Do regulacji częstotliwości pracy generatora służy trymer C3.

Schemat montażowy generatora oraz sposób zainstalowania na płytce drukowanej (za pomocą pięciu pojedynczych szpilek „gold-pin“ o długości 17 mm) pokazano na **rys. 15**. W przypadku zastosowania generatora z **rys. 14** na płytce głównej nie montujemy elementów: T30, R39, R40, R41, X1, C7 i C9.

Modyfikacje U1

Działanie układu U1 („serca“ zegara) zostało opisane w języku VHDL (komplet źródeł opublikowaliśmy na CD-EP2/2003B). Dzięki zastosowaniu niezwykle elastycznego układu PLD, którego działanie jest modelowane za pomocą języka opisu sprzętu, każdy z Czytelników może samodzielnie zmodyfikować jego działanie lub zwiększyć możliwości zegara. Pokażę teraz prosty



Fot. 13. Umieszczenie złącza JTAG na płytce zegara

przykład modyfikacji opisu, co spowoduje zmianę sposobu sygnalizowania przełączenia zegara w tryb ustawiania czasu.

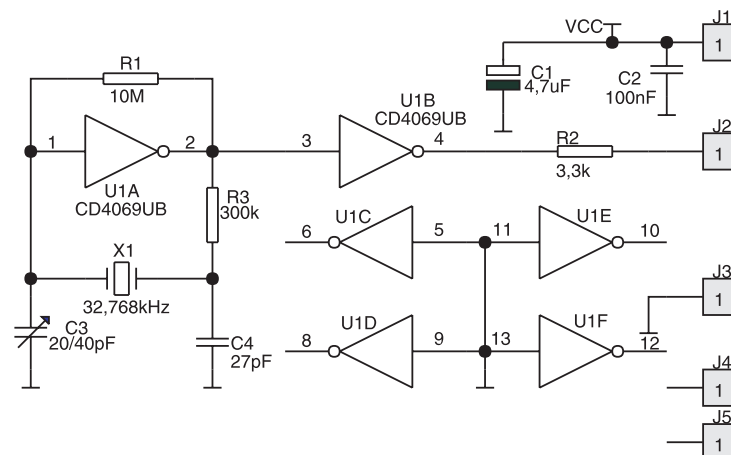
W modelowych egzemplarzach podczas normalnego zliczania czasu dwukropki miga z częstotliwością 1 Hz, natomiast po przełączeniu w tryb ustawiania czasu zaczynał migać z częstotliwością 4 Hz (zgodnie ze schematem blokowym z **rys. 6** - EP2/2003). Za przełączanie sygnału sterującego dwukropkiem odpowiada multiplexer *mux_dwukropka* (**list. 3**). Zwróćmy uwagę na poniższy fragment tego listingu:

```

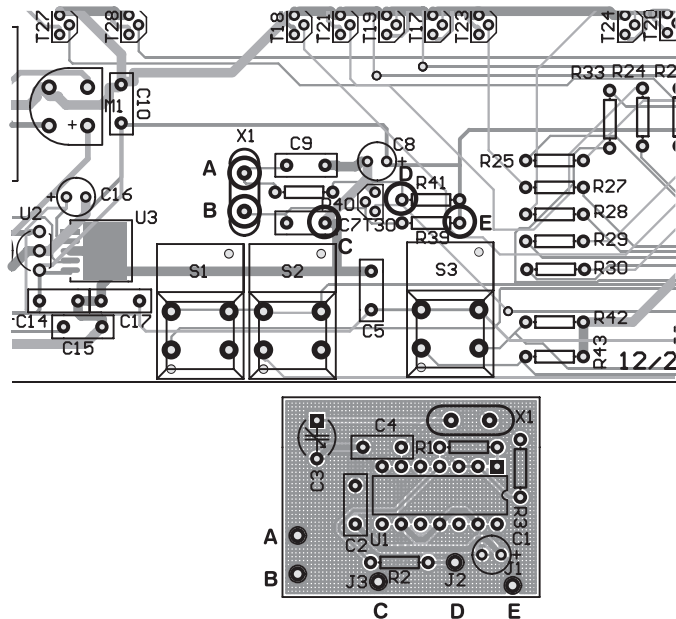
mux_dwukropka: mux_2_1 port
map (
in0 => int_1Hz,
in1 => int_4Hz,
a => a,
out_mux => ster_dk
);

```

Wejściu *in1* multiplexera jest przypisany sygnał o nazwie *int_4Hz*, pobierany z wyjścia *clk_4_Hz* preskalera częstotliwości wzorcowej. Proponuję zmienić sposób sygnalizowania przełączenia



Rys. 14. Schemat elektryczny zmodyfikowanego generatora wzorcowego



Rys. 15. Sposób montażu generatora wzorcowego

nia w tryb ustawiania - zamiast szybkiego migania dwukropki będzie się świecił na stałe. Wymaga to wprowadzenia jednej zmiany w opisie sposobu przyłączenia multiplexera *mux_dwukropka*, jak to pokazano na listingu poniżej:

```

mux_dwukropka: mux_2_1 port
map (
    in0 => int_1Hz,
    in1 => '1',
    -- ^Tu wprowadzono zmianę!
    a => a,
    out_mux => ster_dk
);
    
```

Każda wprowadzona zmiana wymaga oczywiście ponownego skompilowania opisu i zaprogramowania układu U1.

Likwidacja zakłóceń

Podczas eksploatacji zegara w warunkach domowych wpływ zakłóceń występujących w sieci energetycznej na jego działanie był trudny do wychwycenia. Smutna prawda objawiła się po przeprowadzeniu kilku prób w środowisku semi-przemysłowym, gdzie do linii zasilających są dołączone silniki dużej mocy, lampy sodowe i świetlówki z zapłonnikami bimetalicznymi, a także impulsowe regulatory mocy. Zakłócenia występujące w liniach zasilających powodowały, że zegar dość zmieniał swoje ustawienia, często układ U1 „zawieszał się“.

Okazało się, że doskonałym lekarstwem na problemy tego typu było zastosowanie w li-

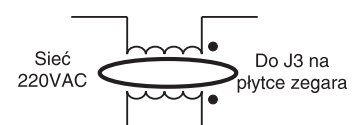
niach zasilających specjalnego dławika odkłócającego (rys. 16), który został wymontowany ze starego zasilacza z PC. Cechą charakterystyczną tego dławika jest to, że ma dwa uzwojenia nawinięte współbieżnie i symetrycznie na jednym rdzeniu, co znakomicie likwiduje zakłócenia impulsowe. Alternatywnym wyjściem może być zastosowanie gniazda sieciowego z wbudowanym filtrem LC (dostępne m.in. w Elfie i TME).

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

W artykule wykorzystano zdjęcia i informacje pochodzące ze stron:

- <http://www.webx.dk/oz2cpu/index.htm>,
- <http://www.arttec.net/art/Relevators.html>,
- <http://www.amug.org/~jthomas/iee-enix.html>,
- <http://w1.871.telia.com/~u87127080/ind/z560m.htm>,
- <http://www.mif.pg.gda.pl/homepages/frank/sheets/084/z/Z560M.pdf>.

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl?pdf/marzec03.htm> oraz na płycie CD-EP3/2003B w katalogu PCB.



Rys. 16. Sposób włączenia dławika odkłócającego w linię zasilania