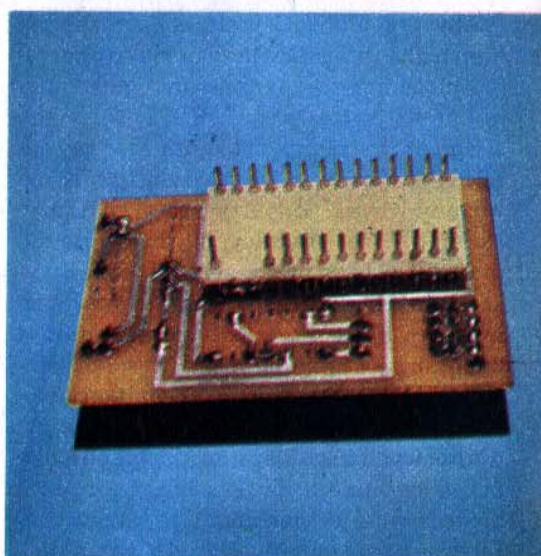
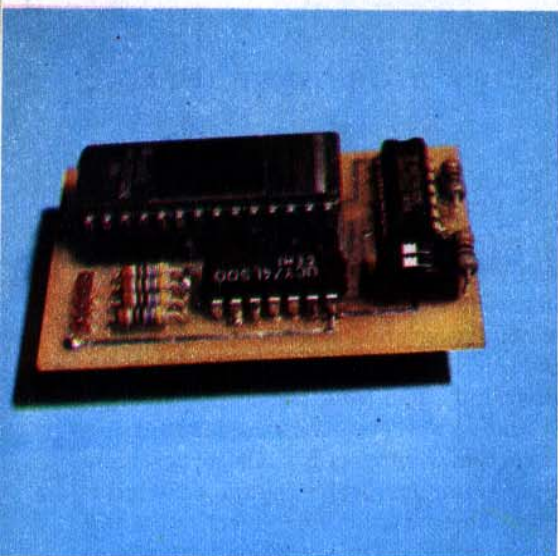


## DWA BIOSY W PECETA...

Pracujący komputer osobisty klasy IBM PC/XT, czyli tzw. gwarowo „pecet”, można wyobrazić sobie jako zespół czterech współpracujących ze sobą zależnych hierarchicznie warstw: aktualnie wykonywanego programu, systemu operacyjnego (najczęściej MS DOS, ale niekoniecznie), programu BIOS (ang. Basic Input/Output System) i sprzętu. Ten przedostatni jest zawartym w pamięci EPROM programem uruchamianym bezpośrednio po włączeniu komputera do sieci i jest odpowiedzialny za wstępne ustawienie konfiguracji sprzętu (pamiętajmy, że specjalizowane peryferyjne układy scalone trzeba zaprogramować, aby realizowały požądane przez nas funkcje – jest to cena ich uniwersalności) i jednocześnie stanowi warstwę pośrednią pomiędzy sprzętem a systemem operacyjnym. Ta druga funkcja powoduje, że sprzęt widziany jest przez system operacyjny zawsze (no, prawie zawsze...) tak samo, bez względu na swoją rzeczywistą konfigurację.

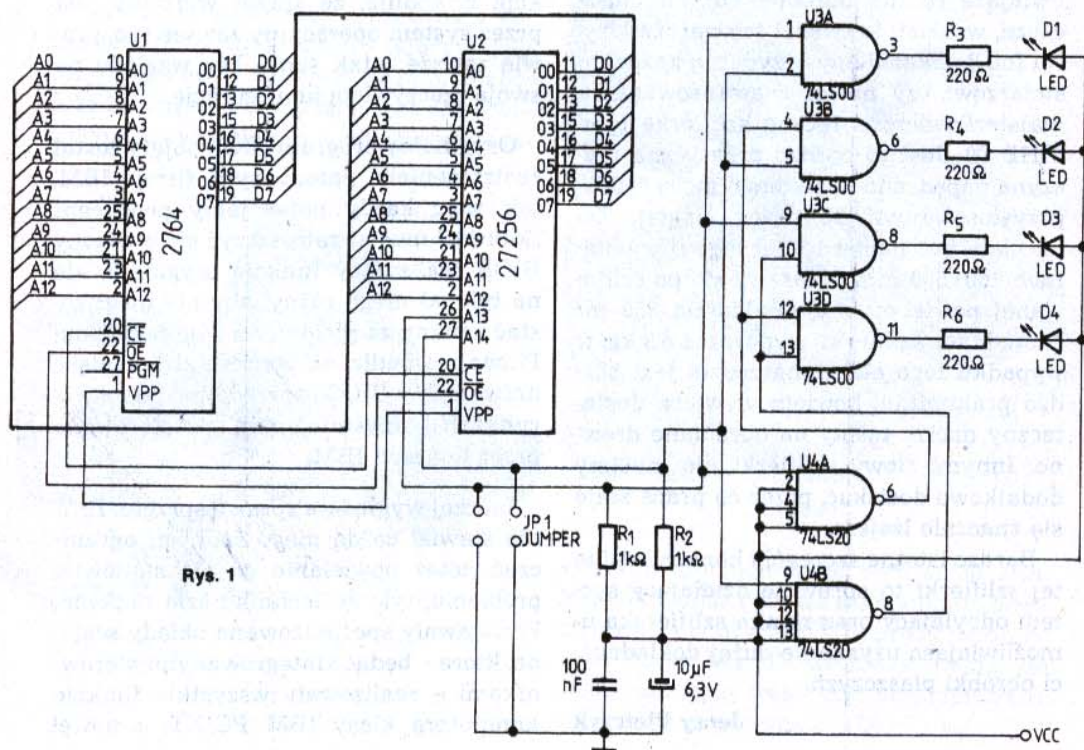
Oryginalny program BIOS objęty został zastrzeżeniem patentowym firmy IBM, tak więc każdy potencjalny producent „klonów” musiał zatroszczyć się o własny BIOS realizujący funkcje oryginału, ale na tyle od niego różny, aby nie mógł zostać uznany za plagiat. Aż w końcu firma Phoenix wpadła na pomysł, aby napisać uniwersalny BIOS i sprzedawać go na warunkach korzystniejszych niż oferowane przez koncern IBM.

Inaczej wyglądała sprawa sprzętu. IBM nie stawiał co do niego żadnych ograniczeń, toteż powielanie go nie stanowiło problemu, tyle że technika szła naprzód. Powstawały specjalizowane układy scalone, które – będąc zintegrowanymi sterownikami – realizowały wszystkie funkcje komputera klasy IBM PC/XT, a nawet



więcej tak, że wystarczyło dołączenie do nich procesora, układów pamięci ROM (zawierających BIOS) i RAM (oczywiście

o pojemności 1 Mb) i ewentualnie koprocesora, aby uzyskać kompletną płytę główną komputera.



Rys. 1

```

program bios_to_file;

uses dos;

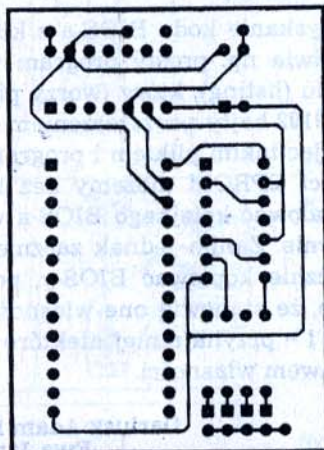
var bios_file: file;
    bios_buf: array [1..8192] of byte;
    filename: string;
    i: integer;

begin
    if ParamCount = 0 then
        begin
            writeln ('Brak nazwy pliku docelowego.');
            halt
        end;
        filename := Paramstr (1);
        for i:=0 to $ffff do bios_buf [i+1] := mem [$fe00:i];
        filename := filename + '.BIN';
        assign (bios_file, filename);
        {$I-}
        rewrite (bios_file, 8192);
        {$I+}
        if IOresult <> 0 then
            begin
                writeln ('Niedobrze - wykonywanie programu przerwane.');
                halt
            end;
        blockwrite (bios_file, bios_buf,1);
        close (bios_file);
        writeln ('BIOS zapisany do pliku ',filename)
    end.

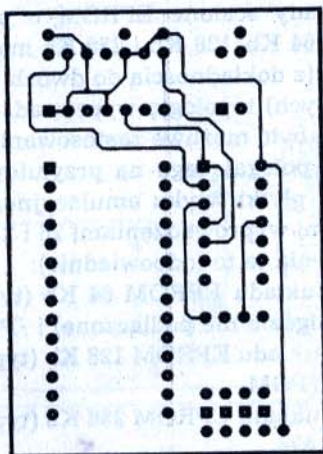
```

Jeżeli dołożymy do tego sytuację panującą na rodzimym rynku komputerowym, gdzie każdy podzespół pochodził z innego

(czytaj: możliwie jak najtańszego) źródła, nie dziwny się, że całość miewała niekiedy kłopoty z prawidłową współpracą.



Rys. 2



Stąd właśnie pochodzi geneza „dwóch BIOS-ów w peceta”. Aluzja do przysłowio- wych „dwóch grzybów w barszczu” jest nieco przewrotna, gdyż niekiedy możli- wość szybkiej zmiany BIOS-a jest bardzo przydatna. Rzecz w tym, że BIOS-y oferowane przez naszych dostawców nie za- wsze są w pełni dopasowane do tej możli- wości, a dostęp do oryginału jest trudny, jeżeli nie wręcz niemożliwy. Bywa że uru- chomienie programu jest niemożliwe przykładowo z BIOS-em firmy AMI, a możliwe z BIOS-em firmy Phoenix, mimo że ten ostatni pracuje wolniej i ma o wiele mniejsze możliwości. Można oczywiście mieć na taką okazję przygotowany komplet EPROM-ów, ale takie postępowanie na dłuższą metę nie wyjdzie płycie głównej na zdrowie.

W IBM PC/XT BIOS usytuowany jest w ostatnich ośmiu KB jednomegabajtowego obszaru adresowego procesora. Propono- wane rozwiązanie stanowi przystawkę wkładaną w miejsce oryginalnego układu EPROM. Schemat ideowy przystawki przedstawiony jest na rysunku 1. Na płytce przystawki umieszczona jest pamięć EPROM o pojemności maks. 256 Kb zawierająca do czterech programów BIOS o po- czątkach (względem jego początku) pod adresami 0, 8192, 16384 i 24576. Wybór pro- gramu dokonywany jest przełącznikami. Diody świecące stanowią optyczną sygna- lizację wybranego programu.

Układy scalone EPROM o pojemno- ściach 64 Kb, 128 Kb i 256 Kb mają jedna- kową (z dokładnością do dwóch bitów ad- resowych) topologię wyprowadzeń, dlate- go też było możliwe zastosowanie rozwią- zania polegającego na przylutowaniu do spodu płytki wtyku emulacyjnego z usu- niętymi wyprowadzeniami 26 i 27. Wypro- wadzenia te to (odpowiednio):

dla układu EPROM 64 Kb (typ 2764) – NC (nigdzie nie podłączone) i /PGM,

dla układu EPROM 128 Kb (typ 27128) – A13 i /PGM,

dla układu EPROM 256 Kb (typ 27256) – A13 i A14.

Przełącznik umieszczony na płytce przystawki zapewnia w zależności od swo- jej nastawy wymuszenie na nich stanów logicznych powodujących wybór odpow- iedniej ośmiokilobajtowej części pamię- ci „na stałe”, czyli odpowiedniego progra- mu BIOS. Układy U1 i U2 stanowią deko- der typu 1 z 4, sterujący diodami świecący- mi.

Dodatkowe szczegóły montażowe poka- zane są na fotografiach.

Na rysunku 2 pokazany jest schemat obwodu drukowanego przystawki. Należy tu jednak przestrzec przed automatycz- nym jego skopiowaniem – różnorodność o- becnych w kraju konfiguracji sprzęto- wych (rozmiary płyty głównej, rozmiary obudowy, wzajemne usytuowanie peryfe- riów) jest tak olbrzymia, że najpierw war- to sprawdzić, czy w danym sprzęcie płytka przystawki nie będzie kolidowała z dy- skiem sztywnym lub kartami rozszerzeń, a w przypadku kolizji przeprojektować ją.

Na zakończenie kilka uwag praktycz- nych. Zanim zdecydujemy się na wymia- nę BIOS-a, warto w nim trochę pogrzebać, aby przekonać się o jego rzeczywistej wartości. Do tego celu można posłużyć się pakietem SOURCER (firmy V COMMU- NICATIONS), który dokonuje deasembla- cji programów (w tym i BIOS-ów) wyposa- żając odzyskany listing źródłowy w boga- te komentarze.

Odzyskanie kodu BIOS-a z komputera umożliwia np. prosty program w Turbo Pascalu (listing), który tworzy plik o dłu- gości 8192 bajty z rozszerzeniem BIN. Dy- sponując takim plikiem i programatorem pamięci EPROM możemy bez kłopotów zainstalować kolejnego BIOS-a w naszej maszynie. Zanim jednak zaczniemy bez- krytycznie kopiować BIOS-y, pomyślimy chwilę, że stanowią one własność produ- centa, i – przynajmniej niektóre – objęte są prawem własności.

Dariusz Adam Przygoda  
Ewa Jarosiewicz