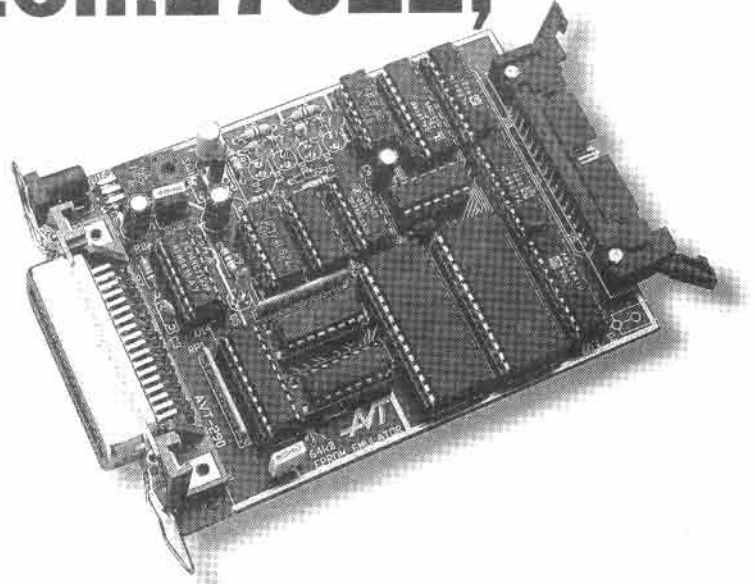


Symulator pamięci EPROM 2716...27512, część 1 kit AVT-270



Przedstawione w artykule urządzenie pozwala na emulację najbardziej popularnych, 8-bitowych pamięci EPROM, o pojemności od 2kB do 64kB, przy pomocy komputera PC wyposażonego w standardowy port drukarkowy. Sterowanie pracą urządzenia przy pomocy jednego przycisku, bez użycia specjalnego oprogramowania, czyni je prostym i jednocześnie funkcjonalnym w użyciu.

Temat podjęty przez nas w artykule doczekał się wielu opracowań, od prostych układów emulujących jeden rodzaj pamięci EPROM, do układów bardzo skomplikowanych, pozwalających na pracę w systemach wykorzystujących 16-bitowe „kości“ o pojemnościach rzędu setek kB. Wszystkie mniej lub bardziej skomplikowane konstrukcje łączą w sobie część sprzętową oraz współpracujący z nią program, który w bardziej lub mniej wymyślny sposób przesyła kod emulowanego programu z komputera do emulatora. Rzadziej spotykane konstrukcje, pracujące bez wykorzystania specjalnego oprogramowania, posiadają często wiele tzw. dip-przełączników, które pozwalają przed użyciem na ręczne ustalenie konfiguracji emulatora przez użytkownika.

Prezentowane urządzenie łączy w sobie prostotę sterowania z możliwością emulacji szerokiej gamy popularnych EPROMów.

Procedura przygotowania do emulacji jest bardzo prosta, należy po włączeniu emulatora do układu docelowego i zasileniu go, ustawić klawiszem typ emulowanego EPROMu, po czym za pomocą komendy DOS'u można przesać zawartość programu do emulatora. Przykładowa składnia polecenia podana została poniżej:

`COPY /b plik_binarny PRN,`
gdzie:

- plik_binarny - to nasz zbiór z "zawartością EPROM",
- PRN - to nazwa domyślnego portu drukarki (LPT1, przy innym porcie trzeba podać jego numer np. LPT2 itd.). Parametr /b to znak dla komendy COPY aby potraktować zbiór jako binarny,

Użytkownikom popularnego programu Norton Commander wystarczy komenda ukryta pod klawiszem F5, aby po chwili rozpocząć pracę urządzenia.

Wspomniany zbiór wejściowy musi być w formie binarnej, bowiem emulator w trakcie przyjmowania danych ładuje każdy bajt po kolei, za każdym razem zwiększając adres w pamięci o 1. Dla tych, którzy pracują na zbiorach w popularnych formatach, np. In-

Podstawowe dane techniczne:

- ✓ emulacja pamięci EPROM: 2716, 2732, 2764, 27128, 27256, 27512, także w wersji CMOS,
- ✓ brak konieczności stosowania specjalistycznego oprogramowania do sterowania urządzeniem,
- ✓ do przesyłania danych wykorzystuje się standardowe komendy systemu operacyjnego DOS,
- ✓ "inteligentny" sposób wyboru typu układu za pomocą jednego przycisku,
- ✓ sygnalizacja typu emulowanego EPROM za pomocą LED,
- ✓ funkcja automatycznego resetu z możliwością wyboru jego poziomu: 0 lub 1,
- ✓ sterowanie przez standardowy Centronics z wykorzystaniem kabla od drukarki,
- ✓ możliwość zasilania emulatora z uruchamianego układu.

```
(* Program zamienia plik w formacie Intel-HEX na postac binarna
i przesyła go na drukarkę lub do innego pliku.
Program należy skompilować pod Turbo Pascal'em.
Wywołanie: INTELBIN.EXE <zbiór_wyjsciowy> [<zbiór_wyjsciowy>]
W wypadku nie podania drugiego parametru przyjmowany jest PRN.
*) {$I-}
Program INTELBIN;
uses crt,dos,printer;
const
  nagl_wiersza = ' ';
  max_data = 16; { maksymalna liczba bajtów w rekordzie }
var
  { pliku Intel-HEX }
  a,adres: word;
  pusta_dana,blad: byte;
  plik_we: text;
  plik_wy: file of byte;
  rekord: record
    znak: char;
    ile: byte;
    adres: word;
    info: byte;
    data: array[0..max_data-1] of byte;
    suma: byte;
  end;

Function Daj_rekord(var kod_bledu:byte): boolean;
{ procedura czyta linie w pliku i }
var s: string;
    kod: integer;
    i: byte;
begin
  daj_rekord:=false;
  kod_bledu:=1; { koniec pliku }
  if not eof(plik_we) then
    readln(plik_we,s) else exit;
  kod_bledu:=2; { blad odczytu }
  if IOResult<>0 then exit;
  kod_bledu:=3; { blad w wierszu }
  if s[1]<>nagl_wiersza then exit;
  val('$'+s[2]+s[3],rekord.ile,kod);
  if kod<>0 then exit;
  val('$'+s[4]+s[5]+s[6]+s[7],rekord.adres,kod);
  if kod<>0 then exit;
  val('$'+s[8]+s[9],rekord.info,kod);
  if kod<>0 then exit;
  for i:=1 to rekord.ile do
  begin
    val('$'+s[10+(i-1)*2]+s[11+(i-1)*2],rekord.data[i-1],kod);
    if kod<>0 then exit;
  end;
  val('$'+s[10+rekord.ile*2]+s[11+rekord.ile*2],rekord.suma,kod);
  if kod<>0 then exit;
  kod_bledu:=4;
  { sprawdzenie sumy kontrolnej }
  i:=rekord.ile+hi(rekord.adres)+lo(rekord.adres)+rekord.info+rekord.suma;
  for kod:=1 to rekord.ile do i:=i+rekord.data[kod-1];
  if i<>0 then exit;
  kod_bledu:=0; daj_rekord:=true; { operacja odczytu rekordu OK! }
end;

Procedure Koniec(b:byte);
begin
  case b of
    1: writeln('Konwersja przebiega pomyslnie. ');
    2: writeln('Blad odczytu zbioru wejsciowego. ',#7);
    3: writeln('Blad w wierszu o adresie: ',rekord.adres,#7);
    4: writeln('Blad sumy kontrolnej w pliku wejsciowym. ',#7);
    5: writeln('Emulator nie jest gotowy. ',#7);
  end; [case]
  if b=1 then close(plik_we);
  halt;
end;

BEGIN
  writeln; writeln('Program konwersji "Intel-HEX -> Binary"...');
  adres:=0; pusta_dana:=255;
  if paramcount=0 then
  begin
    if paramcount>1 then
    begin
      assign(plik_wy,paramstr(2));
      rewrite(plik_wy);
    end;
    assign(plik_we,paramstr(1));
    reset(plik_we);
    if IOResult=0 then
    begin
      repeat
        if Daj_rekord(blad) then
        begin
          if adres<rekord.adres then
            repeat
              if paramcount>1
                then write(plik_wy,pusta_dana)
                else write(lst,chr(pusta_dana));
              inc(adres)
            until adres=rekord.adres;
          for a:=1 to rekord.ile do
            if paramcount>1
              then write(plik_wy,rekord.data[a-1])
              else write(lst,chr(rekord.data[a-1]));
            if IOResult<>0 then Koniec(5);
            adres:=adres+rekord.ile;
          end else Koniec(blad);
          until false;
        end else
          writeln('Blad otwarcia pliku: ',paramstr(1),#7);
          halt;
        end else
          writeln('Uzycie: INTELBIN.EXE <zbiór_wyjsciowy>
          [<zbiór_wyjsciowy>]');
        END.
    end;
  end;
end;
```

Listing 1. Postać źródłowa programu służącego do konwersji danych.

tel-HEX lub innych, przyda się małeńki programik do zamiany takiego właśnie zbioru na postać binarną. Jego listing w Pascal'u zamieszczony jest w artykule (list.1). Wzbogaconą o możliwość konwersji zbiorów w formatach: Intel, Millenium, Motorola, RCA, Tektronix, Texas-Instruments oraz TMS7000, wersję programu można także nabyć na dyskietce w dziale obsługi czytelników pod nazwą: „HXBNCONV“.

Opis układu

Schemat blokowy urządzenia przedstawia rys.1. Podstawową rolę pełnią pamięci statyczne RAM U5, U6 o łącznej pojemności 64kB, czyli odpowiadającej wielkości największej EPROM typu 27512. Do zmiany adresu aktualnie ładowanego bajtu w czasie przesyłania danych z komputera PC służy 16-bitowy licznik złożony z układów U3, U4. Pomiędzy wtykiem emulacyjnym, bezpośrednio dołączonym

do podstawki w układzie docelowym, a pamięciami RAM znajdują się trzy 8-bitowe bramy. Ich zadaniem jest „odcięcie” układów pamięci RAM od sygnałów dochodzących z wtyku emulacyjnego w czasie ładowania zbioru z programem do emulatora. Blok oparty na układzie U2 pozwala na odpowiednie sterowanie i konfigurację emulatora w zależności od typu emulowanej pamięci EPROM.

Wszystkie sygnały sterujące pochodzą z łącza równoległego komputera PC (Z1) i są zgodne ze standardem Centronics. W komplecie z emulatorem znajdują się dwie sondy emulacyjne. Pierwsza, 24-pinowa służy do emulacji układów EPROM typu: 2716 i 2732, druga 28-pinowa dla pozostałych (2764...27512).

O ile od strony „emulowanej EPROM” sprawa sterowania jest dość prosta, o tyle komplikuje się gdy przyjrzymy się stronie łącza

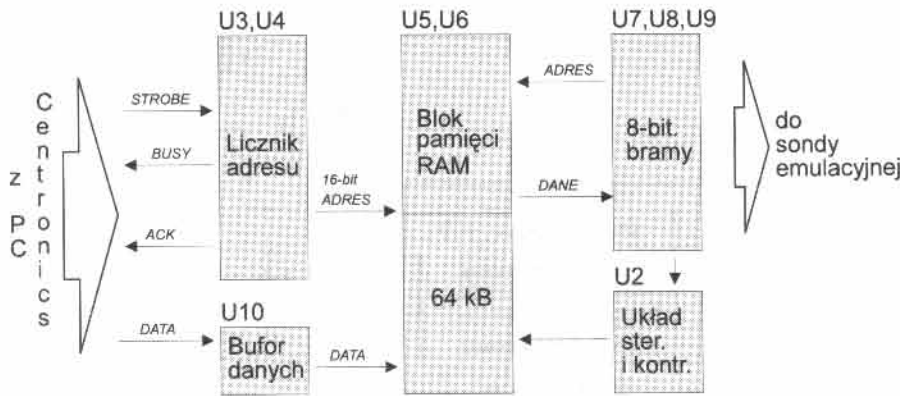
równoległego Centronics. Podczas wysyłania danych z komputera do drukarki transmisja przebiega zgodnie z rys.2.

Nasz emulator, z punktu widzenia komputera, zachowuje się jak zwykła drukarka. Dzięki temu możliwe jest załadowanie pamięci emulatora za pomocą standardowych narzędzi systemu DOS.

Kluczową rolę obok 8-miu sygnałów danych D0...D7 pełnią 3 sygnały wyjściowe, a mianowicie:

- STROBE, komputer informuje drukarkę o gotowych do odbioru danych
- BUSY, drukarka informuje komputer, iż jest w danej chwili zajęta
- ACK, drukarka informuje komputer krótkim ujemnym impulsem o odczycie danych ze złącza i gotowości do przyjęcia następnego bajtu.

Ważne jest przy tym zachowanie odpowiedniej długości gene-



Rys. 1. Schemat blokowy symulatora.

rowanych przez obie strony sygnałów, szczególnie przez stronę odbiorczą. Niespełnienie warunków z rys.2 może prowadzić do błędów danych bądź zawieszenia transmisji, co objawia się najczęściej na ekranie komputera odpowiednim komunikatem o błędzie.

W stanie „jałowym“ na liniach sterujących STROBE, BUSY i ACK panują odpowiednio poziomy logiczne: 1, 0 oraz 1. W momencie rozpoczęcia transmisji danych do drukarki wysyłane bajty po kolei zostają „zatrzaśnięte“ w rejestrze danych portu równoległego PC. Za każdym razem komputer sygnalizuje ten fakt drukarce ujemnym poziomem na linii STROBE, co prawie jednocześnie powoduje odpowiedź z drugiej strony w postaci wysokiego poziomu na linii BUSY, jest to sygnałem dla komputera o wstrzymaniu transmisji. Kiedy drukarka odczyta dane, potwierdza to krótkim ujemnym impulsem na linii ACK, którego koniec jednocześnie przywraca stan niski na linii BUSY, zezwalając komputerowi na wysłanie następnego bajtu danych. Cykl ten przebiega aż do ostatniego bajtu.

Dokładny schemat ideowy emulatora przedstawia rys.3. Poza wspomnianymi układami pamięci RAM, licznikiem adresu oraz układami separującymi zastosowano dodatkowe układy, generujące odpowiednie sygnały sterujące wszystkimi blokami urządzenia oraz transmisją z komputerem. Układ U14a, zajmuje się generacją sygnału oznaczonego na rysunku „MODE“, który przyjmuje stan w zależności od trybu pracy emulatora, czyli ładowania programu użytkownika bądź emulacji EPROM. Dodatkowy przerzutnik U15a poprawia ostrość zboczy

tego sygnału, co jest konieczne do prawidłowej pracy układu. Drugi uniwersalny w postaci U14B generuje sygnał wykorzystywany jako ACK podczas transmisji danych z PC. Dodatkowo sygnał ten powoduje inkrementację licznika adresu (układy U5 i U6). W układzie U10 dana z PC jest dodatkowo zapamiętywana na czas zapisu jej do pamięci RAM. Aby dopasować często „kapryśne“ poziomy logiczne portu Centronics oraz dodatkowo zabezpieczyć wejścia układów naszego emulatora zastosowano bramki-bufory w postaci układu U13.

Blok sterowania konfiguracją symulatora składa się z licznika moduło 8, w postaci popularnego układu U1 oraz zaprogramowanego układu GAL U2, który spełnia 2 zadania. Po pierwsze steruje sygnalizacją wyboru typu EPROM'u (diody LED D1...D4), po drugie „blokuje“ odpowiednie linie adresowe pamięci w zależności od rodzaju emulowanego układu. Dioda D5 sygnalizuje tryb emulacji, natomiast bramki U13F, U12a i U12b podają sygnał resetu układu docelowego. Sygnał ten dostępny jest na gnieździe G2

i może być dodatkowo wykorzystany do resetowania uruchamianego układu podczas ładowania do emulatora danych. Należy tylko przed tym ustalić jego polaryzację za pomocą przełącznika SW3.

Jako licznik adresu zastosowano dwa układy typu 74HCT590. Są to 8-bitowe liczniki asynchroniczne z zatrzaśkiem wyposażonym w trójstanowe wyjścia. Ponieważ układy te są mało popularne w kraju, dla czytelników nie posiadających katalogu zamieścimy krótki opis wyjaśniający znaczenie poszczególnych wejść i wyjść tej jakże funkcjonalnej „kostki“.

Wejścia układu:

CCLK - wejście zliczanych impulsów

CCLKEN - wejście blokowania zliczania

CCLR - wejście kasowania licznika

RCLK - narastające zbocze podane na te wejścia powoduje zatrzaśnięcie stanu licznika w wewnętrznym rejestrze

G - wejście wyboru układu

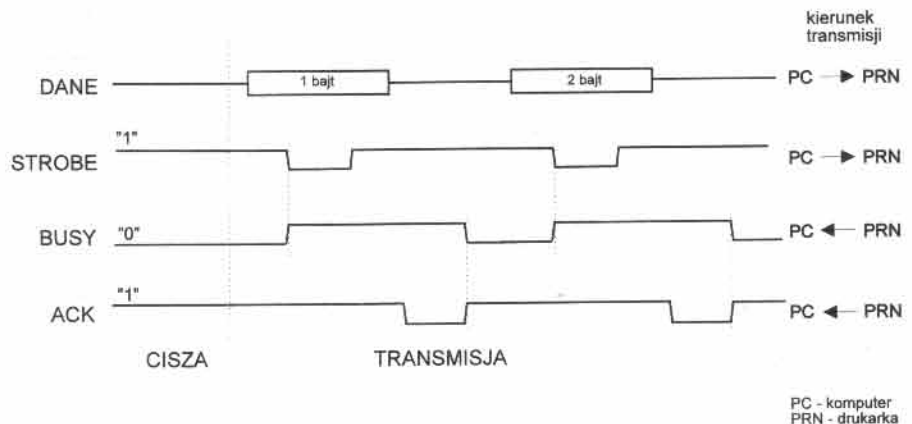
Wyjścia układu:

QA...QH - wyjście rejestru licznika

RCO - wyjście przeniesienia licznika

Pewnej uwagi wymaga też wspomniany układ sterowania U2. Zastosowany programowany układ logiczny GAL zmniejsza ilość potrzebnych elementów w układzie i upraszcza konstrukcję. Jego działanie najłatwiej prześledzić na podstawie tabeli 1.

Wyboru typu emulowanego EPROM'u dokonuje się klawiszem SW1. Jego sekwencyjne wciskanie powoduje zmianę stanu na wyjściach licznika U1 (QB...QD -



Rys. 2. Typowe przebiegi na złączu Centronics.

Tabela 1.

WEJŚCIA				WYJŚCIA				
Stan ENWT	Rodzaj sondy		stan U1	wyjścia adresowe				świecenie
	WT24	WT28	IC IB IA	QA15	QA14	QA13	QA11	LED D1...D4
1	1	0	X 0 0	0	0	0	A11	D1 - 2764
1	1	0	X 0 1	0	0	A13	A11	D2 - 27128
1	1	0	X 1 0	0	A14	A13	A11	D3 - 27256
1	1	0	X 1 1	A15	A14	A13	A11	D4 - 27512
1	0	1	X X 0	0	0	0	0	D1,D2 - 2716
1	0	1	X X 1	0	0	0	A11	D3,D4 - 2732
0	X	X	0 0 0	0	0	0	A11	D1 - 2764
0	X	X	0 0 1	0	0	A13	A11	D2 - 27128
0	X	X	0 1 0	0	A14	A13	A11	D3 - 27256
0	X	X	0 1 1	A15	A14	A13	A11	D4 - 27512
0	X	X	1 0 0	0	0	0	0	D1,D2 - 2716
0	X	X	1 0 1	0	0	0	A11	D3,D4 - 2732
0	X	X	1 1 0	0	0	0	A11	D1 - 2764
0	X	X	1 1 1	0	0	A13	A11	D2 - 27128
1	1	1	X X X	0	0	0	0	wygaszone

X - dowolny stan logiczny

logiczne "0" wybiera rodzaj sondy

wykorzystano część licznika modułu 8, wyjście QA jest wyjściem dzielnika przez 2), która pojawia się na wejściach Ia...IC układu GAL. Na tej podstawie U2 decyduje o zapaleniu odpowiedniej diody LED oraz odblokowaniu właściwych dla danej kości EPROM linii adresowych. Jest to konieczne, bowiem dla różnych pojemności emulowanej pamięci niektóre z tych wejść są niewykorzystane i powinny być zwarte

do masy. W wypadku emulacji założonych układów nerwalgicznymi liniami adresowymi są: A11...A15. Układ U2 przynosi sygnał danej linii adresowej z wtyku emulacyjnego do układu U7 (dalej do pamięci RAM), bądź podaje na odpowiadające danej linii wejście układu U7 logiczne „0”. Uważny czytelnik zauważy brak sterowania linią adresową A12. Nie jest to konieczne, bowiem linia ta, aktywna dla ukła-

dów 2764...27512, w przypadku emulacji pamięci 2716 i 2732 została zablokowana „mechanicznie” poprzez zwarcie końcówki jej odpowiadającej do masy na wtyku 24-pinowej sondy emulacyjnej.

Wejścia układu U2 WT28 i WT24 informują GAL o rodzaju dołączonej sondy emulacyjnej (24 lub 28-pinowej) i w zależności od jej typu pozwalają na wybór odpowiedniej konfiguracji układu. Np. w przypadku dołączonej sondy DIL-24 wciskanie SW1 powoduje naprzemienne wybieranie układu 2716 oraz 2732. Gdy dołączona jest sonda dla emulacji układów od 8kB wzwyż, możliwy jest wybór tylko tych 4 układów, czyli 2764, 27128, 27256 i 27512 (wybór 2716 i 2732 są zablokowane). Jeśli do złącza Z2 nie jest podłączona żadna sonda, wtedy diody D1..D4 nie świecą. Takie prawie „inteligentne” sterowanie nie pozwala na błędne wybranie układu przy niewłaściwej sondzie emulacyjnej, co mogłoby spowodować nieprawidłową pracę lub uszkodzenie układu. Poprzez dodatkowe wejście układu U2, oznaczone ENWT, zwierając jumper JP1 możemy zablokować rozpoznawanie rodzaju dołączonego wtyku emulacyjnego.

Sławomir Surowiński, AVT

WYKAZ ELEMENTÓW

Rezystory

RP1, RP2: 8x10kΩ drabinki rezystorowe

R1, R7: 20kΩ

R2..R5, R9: 300Ω

R6: 2kΩ

R8: 100kΩ

R10, R11: 10kΩ

Kondensatory

C1: 1μF/16V

C2: 4.7μF/16V

C3: 680 pF

C4: 10μF/16V

C5, C6: 100μF/16V

C7, C8: 100nF

Półprzewodniki

U1: 74HCT93

U2: zaprogramowany GAL16V8

U3, U4: 74HCT590 (HC590)

U5, U6: pamięci SRAM 62256

U7..U9: 74HCT541

U10: 74HCT574

U11: 74HCT32

U12: 74HCT04

U13: 74LS07

U14: 74HCT123

U15: 74HCT74

U16: 7805 lub podobny

D1...D4: LED zielone

D5: LED czerwona

D6: 1N4148

Różne

SW1: przycisk chwilowy

Z1: złącze DB36 drukarkowe do druku

Z2: złącze "z tapkami" 34-pinowe

G1: gniazdo zasilania z bolcem

G2 - gniazdo "reset" (dowolne)

SW2, SW3: przełącznik 2-pozycyjny

Dyskietka z oprogramowaniem

(opcja na zamówienie)

Uwaga: zamiast układów TTL serii

HCT można zastosować układy LS.