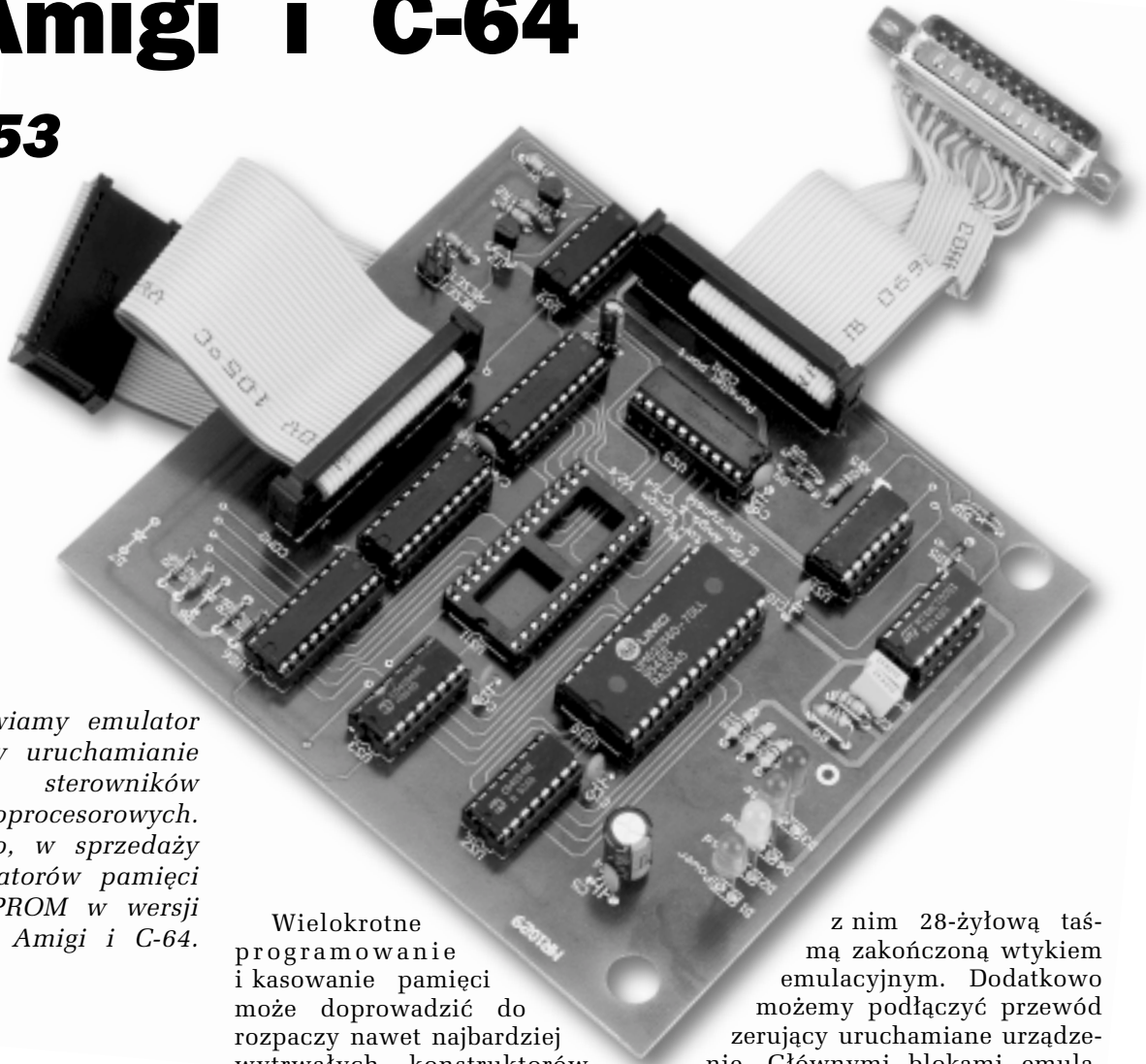


Emulator pamięci EPROM do Amigi i C-64

AVT-5053



Przedstawiamy emulator usprawniający uruchamianie sterowników mikroprocesorowych. Jak wiadomo, w sprzedaży nie ma emulatorów pamięci EPROM w wersji dla Amigi i C-64.

Wielokrotne programowanie i kasowanie pamięci może doprowadzić do rozpaczy nawet najbardziej wytrwałych konstruktorów. Emulator może ułatwić życie, jest bowiem takim przyrządem w pracowni elektronika, jak stetoskop w gabinecie lekarza. Emulator zastępuje pamięci ROM i EPROM podczas uruchamiania systemów mikroprocesorowych z tymi pamięciami. Pamięci tego rodzaju spełniają w nich na przykład funkcje generatora znaków. Emulator może symulować pamięci do 64kB, zarówno w wersji MOS jak i CMOS.

Opis układu

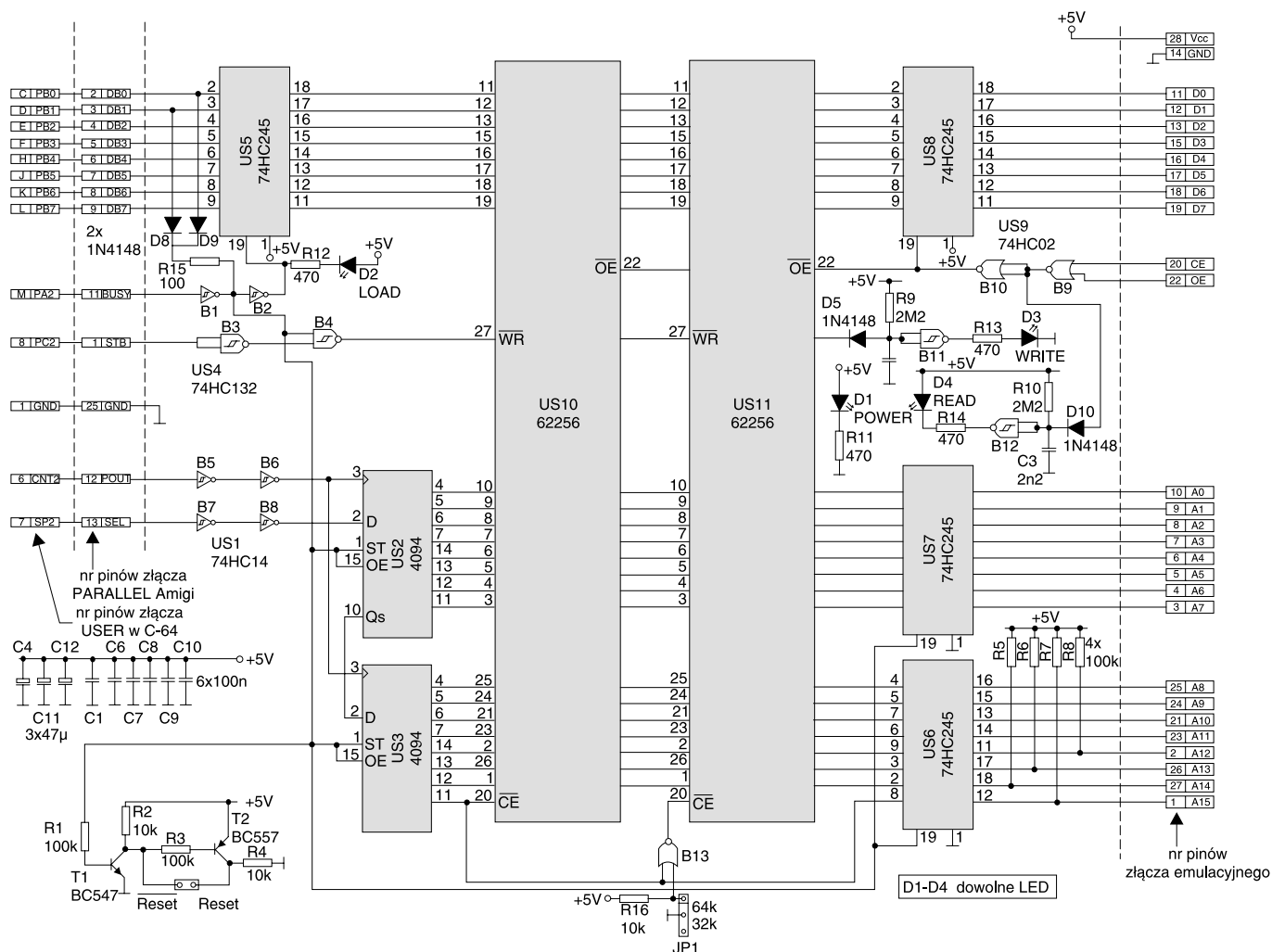
Schemat elektryczny emulatora pokazano na **rys. 1**. Współpracuje on z komputerami za pośrednictwem portu PARALLEL w Amidzie lub USER w C-64. Zasilanie emulatora jest pobierane z uruchamianego systemu. Emulator łączymy

z nim 28-żyłową taśmą zakończoną wtykiem emulacyjnym. Dodatkowo możemy podłączyć przewód zerujący uruchamiane urządzenie. Głównymi blokami emulatora są dwie pamięci o pojemności 32kB każda. Ponieważ emulator przewidziano do uruchamiania systemów z 8-bitową szyną danych, 32kB jest przeważnie maksymalną wielkością pamięci EPROM/ROM.

Niektórych Czytelników zdziwi może to, że połączenia układu US11 z US6, 7, 8 od strony pamięci RAM nie są oznaczone (brak etykiet). Nie jest to błąd w druku. Po prostu wyprowadzenia te są połączone z układami US2, 3, 4, 5 znajdującymi się w lewej części schematu. Taki sposób przedstawiania niektórych połączeń przyjął się w schematach bardziej skomplikowanych urządzeń, np. mikroprocesorowych. Tak narysowany schemat jest bardziej zwarty i bardziej czytelny.

Charakterystyka emulatora

1. Praca z każdym komputerem Amigą (OS2.04+) i C-64, za pośrednictwem portu równoległego.
2. Szybkość transferu z komputera 4kB/s.
3. Program transmisji danych napisany w assemblerze.
4. Pełna kontrola błędów programowych.
5. Praca programu w trybie wielozadaniowym.
6. Wbudowany port Arexxa.
7. Zabezpieczenie od strony złącza emulacyjnego buforami 74HCT245.
8. Zasilanie z uruchamianego systemu, podtrzymanie pamięci z komputera.



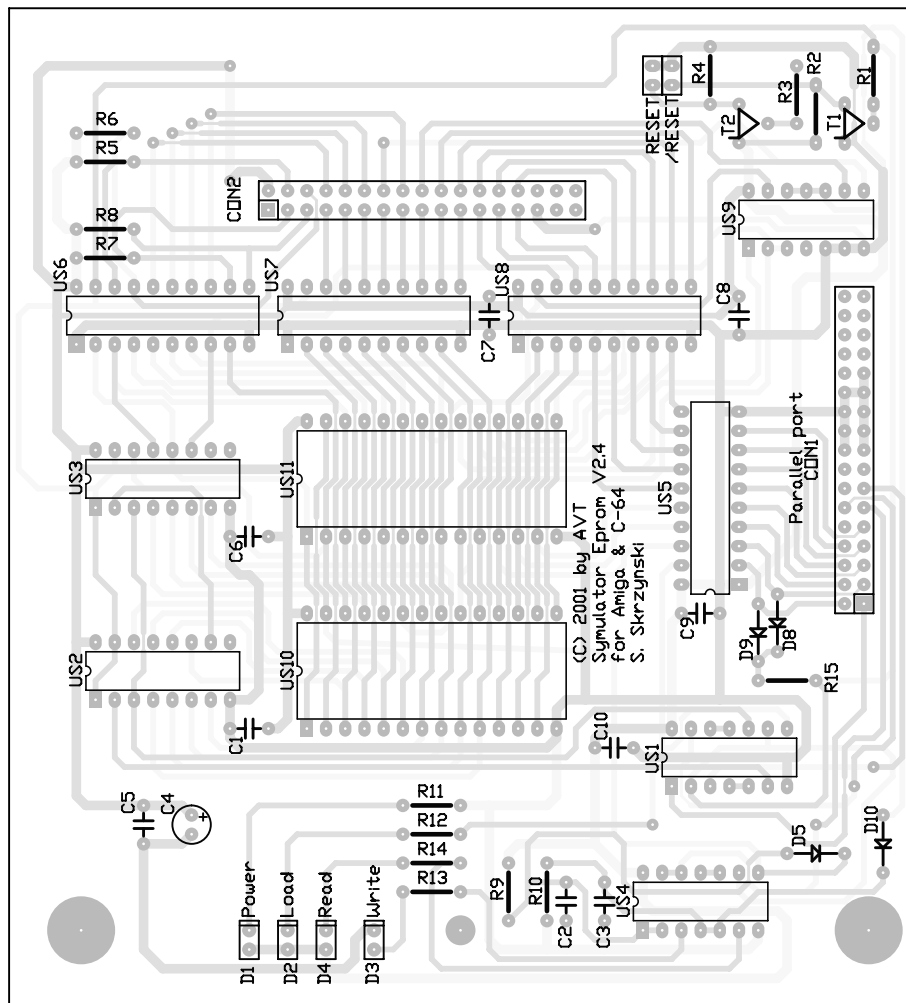
Rys. 1. Schemat elektryczny emulatora pamięci.

Zależnie od stanu linii BUSY uaktywniane są rejestry US2, US3 (poziom niski BUSY) lub bramy US6, US7 (poziom wysoki BUSY). Przyjmijmy, że BUSY znajduje się w stanie „0”. Na wyjściu bramki B1 pojawi się wówczas poziom wysoki, wskutek czego wyjscia buforów US6, US7 znajdują się w stanie trzecim, natomiast wyjscia rejestrów US2, US3 są aktywne. Liniami POUT i SEL przesyłany jest do rejestrów adres wpisywanej komórki pamięci. Następnie do postu wpisywana dana. Powoduje to automatyczną zmianę (na chwilę) poziomu linii STB na niski, co spowoduje, za pośrednictwem B2 i B3, pojawienie się impulsu na linii WR pamięci. Dzięki bramce B2 układ US5 jest w stanie aktywnym. Następstwem tego jest wpisanie danej do pamięci RAM. Następnie wysyłany jest do rejestrów kolejny adres (przeważnie o jeden większy), wysyłany daną do portu

itd. Gdy wpisujemy wszystkie dane, zmieniamy poziom linii BUSY na wysoki. Spowoduje to przejście rejestrów w stan trzeci, natomiast bufor US6, US7 zostaną uaktywnione. Brama US5 zostanie również ustawiona w stan trzeci. Bramka B4 nie pozwoli na wpisywanie do pamięci RAM (impulsy z STB nie będą przepuszczane). Adres generowany przez mikroprocesor, z uruchamianego systemu, zostanie przeniesiony na wejście adresowe pamięci RAM. Bufory, poza funkcją odcięcia linii adresowych od procesora, likwidują zakłócenia mogące pojawić się w długim kablu sondy emulacyjnej. Następuje to za sprawą bramek Schmitta wbudowanych w układ. Ustawienie na liniach CE i OE poziomu niskiego, za pośrednictwem bramek B9 i B10, uaktywni wyjscia pamięci RAM (linia OE) oraz bufora US8. Po czasie określonym w katalogach jako *czas*

dostępu, na wyjściach danych pamięci pojawi się bajt danych. Inne kombinacje wartości sygnałów OE i CE powodują przejście US8 w stan trzeci. Dzięki temu emulator zachowuje się jak „prawdziwa” pamięć EPROM. Bramki B2, B11 i B12 sterują diodami LED. Odczyt pamięci przez mikroprocesor powoduje zaświecenie diody „Read”. Ponieważ są to krótkie impulsy, zastosowano kondensator C3, który powoduje przedłużenie czasu świecenia LED. Dzięki temu obserwujemy ciągle świecenie pełną jasnością. Dioda oznaczona jako „Write” świeci podczas wysyłania danych przez port. Kondensator C2 zastosowano w tym samym celu co C3.

Dioda „Load” świeci, gdy na linii BUSY jest poziom niski, a więc w czasie, gdy jest możliwy zapis danych do RAM. Obwód z tranzystorami T1 i T2 generuje sygnał „Reset” do uruchamianego



Rys. 2. Rozmieszczenie elementów na płycie drukowanej emulatora.

systemu. Dostępne jest wyjście z aktywnym poziomem niskim lub wysokim.

Montaż i uruchomienie

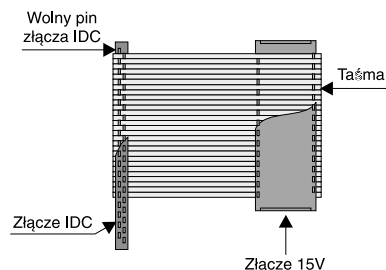
Schemat montażowy płytki emulatora pokazano na rys. 2. W pierwszej kolejności montujemy wszystkie podstawki i elementy bierne. Następnie zaciskamy złącza i sondę symulacyjną na taśmie 28-przewodowej. Zastosowano taką taśmę, aby istniała możliwość symulowania pamięci 2716 oraz mniejszych, czy też innych, nietypowych. Typ symulowanej pamięci zależy od zastosowanego kabla. Przy kablu zaproponowanym w tym emulatorze istnieje możliwość symulowania pamięci 2764, 27128, 27256, 27512 w typowych układach aplikacyjnych. Kabel należy zacisnąć tak, aby jego pierwsza żyła była połączona z drugim wyprowadzeniem złącza IDC (rys. 3). Taśma nie powinna być zbyt długa - jej

maksymalna długość nie powinna przekraczać 30cm.

Do połączenia emulatora z komputerem użyjemy typowego przedłużacza DB25 (w przypadku Amigi) lub kabla przejściówki (w przypadku C-64).

Wtyk DB25 zaciskamy na kablu i łączymy z płytą emulatora za pomocą złącza IDC34. Jeśli nie posiadamy wtyku DB25 zaciskanego na kabelu, można taśmę przylutować bezpośrednio do styków złącza DB25. Sposób połączenia przewodów ze złączem przedstawiono na rys. 4.

Po doprowadzeniu zasilania, ale bez zamontowanych układów, sprawdzamy napięcie na ich wyprowadzeniach zasilaniowych. Następnie wkładamy wszystkie układy scalone. Uruchamiamy program *EmulatorTest*. Postępując zgodnie z jego poleceniami testujemy urządzenie. Przeprowadzone testy są kolejno wyświetlane na oknach z opisami. Przykładowe



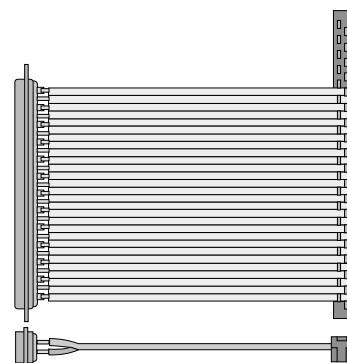
Rys. 3. Sposób wykonania kabla emulacyjnego.

okno pokazano na zrzucie ekranu na rys. 5. Program poinformował o wysłaniu danej \$5555 do rejestrów US2 i US3 (4094). Należy wówczas sprawdzić stany logiczne na poszczególnych wyprowadzeniach układów scalonych.

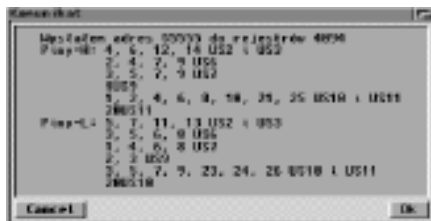
Okno podzielono na dwie części. W tym przykładzie wyszczególniono pierwsze wyprowadzenia, na których powinien być poziom H, w drugiej części wyszczególniono wyprowadzenia, na których powinien być poziom L. Jeśli sprawdzone poziomy są prawidłowe naciskamy przycisk „OK“, a jeśli nie, wycofujemy się z testu naciskając „Cancel“.

Testy ułożone są w określonym porządku. Najpierw sprawdzane są linie BUSY, POUT, SEL (układy US1, US4). Jeśli wynik jest pozytywny sprawdzane są rejestry US2 i US3, następnie brama US5, a na koniec obwód zapisu RAM (bramki B3, B4 układu US4).

Jeśli więc jakiś test wypadł niepomyślnie, procedurę należy przerwać i usunąć usterkę. Gdybyśmy testy przeprowadzali dalej, z pewnością ich wyniki będą negatywne, bo np. gdy uszkodzony będzie US1, nie ma mowy o poprawnym działaniu US2 i US3. Dzięki wyświetlonym podpowie-



Rys. 4. Sposób wykonania kabla połączeniowego.



Rys. 5. Widok okna programu testowego.

dziom znalezienie ewentualnego błędu będzie łatwe. Po sprawdzeniu układu można uruchomić program. Pozostało jeszcze wyjaśnić rolę złącza JP1. Jeśli zastosujemy dwie pamięci (symulacja maks. 64kB) jumper należy ustawić tak, aby zwieriał piny 1-2 (jumper przy napisie 64kB). Przy logicznym „0” na wejście bramki NOR, neguje ona sygnał pochodzący z linii adresowej A15. Dzięki temu do adresu \$7FFF wybrany jest układ US10, a od adresu \$8000 układ US11. Gdy jumper ustawimy tak, aby zwieriał piny 2-3, na bramkę NOR zostanie podany poziom H (symulacja maks. 32kB). Wtedy bez względu na stan linii adresowej A15 zawsze jest wybrany układ US11. Podstawka pod US10 musi być pusta! W przeciwnym przypadku może dojść do kolizji danych pomiędzy US10 i US11 (jednoczesne wybranie obu pamięci). Mogłoby wydawać się, że JP1 jest niepotrzebny, ale tak nie jest. Załóżmy, że bramka B13 pracuje jako inwerter, a układu US11 nie ma w podstawce. Gdy np. symulowana jest pamięć 8KB, na piny 1 (Vpp) i 27 (PGM) podawany jest poziom H. Wówczas dane w emulatorze zostaną wpisane pod adresy: \$0000, \$2000, \$4000, \$6000 oraz w nieistniejący obszar: \$8000, \$A000, \$C000, \$E000.

System mikroprocesorowy wygeneruje adresy \$C000...\$FFFF (Vpp=A15, PGM=A14), co spowoduje wybranie układu US11, którego przecież nie ma w podstawce.

Oprogramowanie dla Amigi

Obsługa programu sterującego pracą emulatora (jego okno pokazano na rys. 6) jest łatwa, a ewentualne wątpliwości rozwieje instrukcja umieszczona na CD-EP02/2002B.

Program rozpoznaje pliki binarne i w formacie IntelHex. Wbu-

dowany skrypt Arexxa umożliwia sterowanie emulatorem z zewnętrznych programów. Dołączono także program wysyłający dane z Amigi do CA80 oraz programy rozdzielające dane 16-bit na bajty parzyste i nieparzyste lub bity parzyste i nieparzyste. Dodatkowo polecam kompilatory Frankenstein zawierające między innymi kompilatory dla 6502, 6805, 6809, 6811, 8048, 8051, 8096, Z8, Z80 oraz skrypty Arexxa ułatwiające kompilacje dla 6502, 8051 i Z80. Skrypty te po kompilacji powodują wysyłanie danych do emulatora EPROM i generowanie sygnału zerującego dla uruchamianego systemu. Dodatkowo na płycie zamieściłem skrypty umożliwiające wysyłanie danych do AVT-497, AVT-870, AVT-995, AVT-2250. Cały pakiet można ściągnąć z www.ep.com.pl lub www.home.mck.pl/~r-mik/kompilatory.htm. Jest on także dostępny na płycie CD-EP02/2002B. Szczegółowe informacje o sposobie instalacji pakietu kompilatorów znajdują się w pliku „Compiler.DokPL”.

Jeśli uruchamiany mikroprocesor nie wykorzystuje całej pamięci, to nieużywane linie adresowe należy połączyć z masą lub szyną zasilania. Unikniemy w ten sposób oscylacji na niewykorzystanych liniach adresowych. Program przy emulowaniu pamięci mniejszych od 27512 wpisuje dane we wszystkie możliwe bloki. Po ustawieniu typu pamięci 27128 i wysłaniu danej z RAM Amigi pod adres \$0000 zostanie ona zapisana pod adresami: \$0000, \$4000, \$8000, \$6000. Dzięki temu nie trzeba sobie „zawracać głowy” offsetami.

Oprogramowanie dla C-64

Program sterujący pracą emulatora współpracującego z C64 obsługujemy z klawiatury. Klawiszami F1, F3 ustalamy obszar w pamięci RAM komputera, który zostanie przepisany do emulatora pod adres ustawiony za pomocą F5. Klawiszem F8 jest uruchamiana transmisja.

Program napisano tak, aby współpracował z innymi programami. Można więc „przechodzić” między Turboassembler, Basic, Mo-



Rys. 6. Okno programu sterującego pracą symulatora.

onitor czy innym programem, a emulatorem. Program jest przygotowany w dwóch wersjach. Pierwsza - EMULATOR.EXE - zawiera monitor asemblera (na dane jest przeznaczony obszar \$3000...\$FFFF). Druga wersja - EMULATOR-A.EXE - nie zawiera monitora, wykorzystuje monitor w ActionReplay (na dane przeznaczony jest obszar \$1200...\$FFFF). Po uruchomieniu program zapamiętuje adres Basic. Dzięki temu przy wyjściu (klawisz strzałka w lewo) wraca do niego lub jeśli startowaliśmy z monitora (np. w Action), to do monitora. Naciśnięcie *Restore* spowoduje przejście do Turboassemblera (jeśli jest w RAM). Wyjście z programu przez CTRL+STOP powoduje restart Actionreplay, Final, Black Box. Wię-

WYKAZ ELEMENTÓW

Rezystory

R1, R3, R5...R8: 100kΩ
R2, R4, R16: 10kΩ
R9, R10: 2.2MΩ
R11...R14: 470Ω
R15: 100Ω

Kondensatory

C1: 100nF
C2, C3: 2.2nF
C4, C11, C12: 47μF/16V
C5...C10: 100nF

Półprzewodniki

D1, D4: LED zielona
D2: LED żółta
D3: LED czerwona
T1: BC547
T2: BC557

US1: 74HC14
US2, US3: 4094
US4: 74HC132
US5...US8: 74HC245
US9: 74HC02
US10, US11: 62256

Różne

CON1, CON2: gniazdo IDC34
JP1: listwa goldpin 1*3
Wtyki zaciskane IDC34 na kabel FLAT (2 szt.)
Wtyk DB25PIN-F na kabel FLAT
Wtyk emulacyjny 28DIP

cej szczegółów można znaleźć w instrukcji (READ ME.EXE) na dyskietce dołączanej do kitu.

Uwagi końcowe

Jeśli zastosujemy niepełne dekodowanie adresów, należy pamiętać o odpowiednim ustawieniu offsetu (przycisk „Adres EPROM“ dla Amigi lub klawisz F5 dla C-64). Przykładowo, jeżeli w uruchamianym systemie zastosujemy pamięć 27256 (ze względu na tę samą cenę co 27128, czy 2764), procesor nadal będzie obsługiwał tylko jej początkowe 16kB. Dla pamięci 27128 linia Vpp i PGM musi być zwarta z +5V. Wejście PGM dla 27128 jest na wyprowadzeniu, na którym jest bit adresowy A14 dla 27256. Procesor wystawiając adresy \$0000...\$3FFF będzie więc odwoływał się do adresów EPROM zawartych w zakresie \$4000...7FFF. W emulatorze należy więc ustawić offset = \$4000. O tym offsecie należy pamiętać

podczas programowania również pamięci w programatorze. Gdyby linia PGM była zwarta z masą, wtedy adresy dla 27256 zawarte byłyby w zakresie \$0000...\$3FFF. Tak jednak prawdopodobnie nigdy nie będzie, ponieważ niewykorzystane linie adresowe „większych“ pamięci, a co za tym idzie linie sterujące „mniejszych“ pamięci łączymy z +5V.

Emulator podłączamy do komputera przy wyłączonym zasilaniu komputera i emulatora. Sondę emulacyjną można umieszczać w uruchamianym systemie tylko przy wyłączonym zasilaniu uruchamianego urządzenia.

Odmienność kopii w stosunku do oryginału

Emulator różni się od prawdziwej pamięci EPROM kilkoma cechami:

1. Większa obciążalność wyjść emulatora dzięki buforom 74HCT245 od rzeczywistej dla pamięci EPROM.

2. Krótszy czas dostępu do pamięci emulatora (100ns) w porównaniu z EPROM (200ns).

3. Duża obciążalność dynamiczna, spowodowana długimi przewodami połączeniowymi.

4. Większy pobór prądu przez emulator w porównaniu do EPROM.

Cechy te mogą spowodować, że urządzenie będzie działać prawidłowo z emulatorem, przestanie po zainstalowaniu pamięci EPROM. W dotychczasowej pracy nie zauważyłem jednak jakichkolwiek nieprawidłowości w pracy emulatora. Został sprawdzony w wielu systemach opartych na procesorach Z80 jak i szybkim 6502.

Sławomir Skrzyński, AVT
slawomir.skrzynski@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/luty02.htm> oraz na płycie CD-EP02/2002B w katalogu PCB.