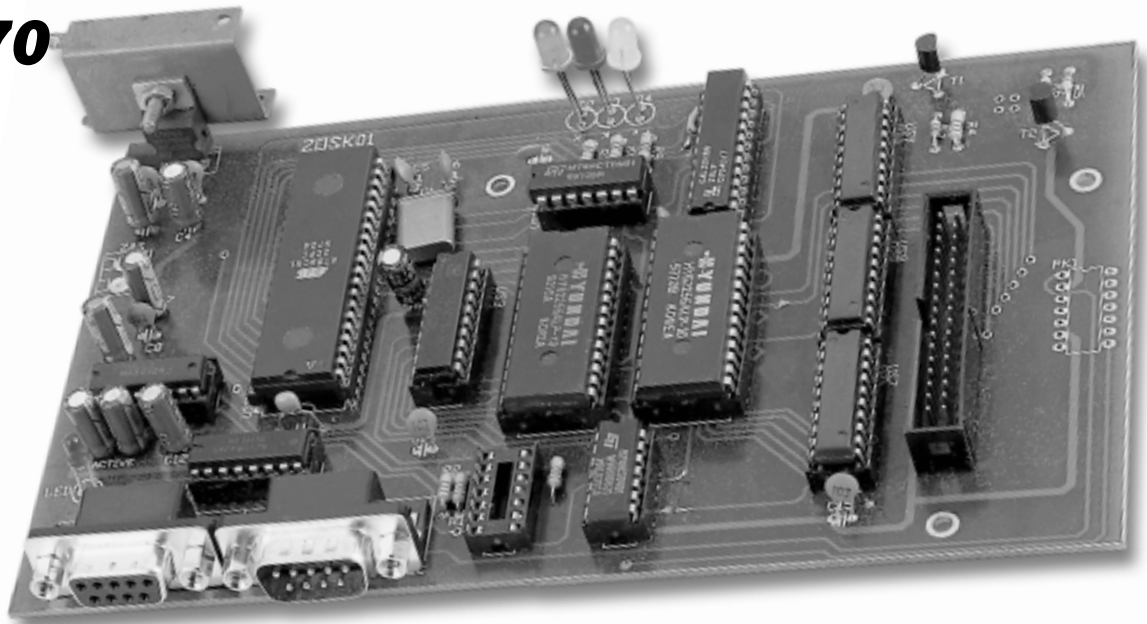


Symulator EPROM/EEPROM do wszystkich typów komputerów, część 1

AVT-870



Opisów takich emulatorów pamięci publikowaliśmy wiele.

Czym wyróżnia się ten emulator? Tym, że wyposażono go w przelotowy port RS232. Koniec „wachlowania” wtyczkami! Bez problemu można podłączyć modem czy inne urządzenie z przelotowym portem RS232 - cóż za wygoda!

Symulator współpracuje z każdym komputerem wyposażonym w port RS232C. Nie jest wymagane żadne specjalne oprogramowanie! Wystarcza systemowa komenda COPY! Warto zaznaczyć, że na pomysł budowy urządzeń współpracujących z każdym komputerem (RS-232C i COPY) wpadł już ktoś inny. Ale na przelotowy port nie, a przynajmniej nic mi na ten temat nie wiadomo.

Kontrowersyjna może być proponowana przeze mnie emulacja pamięci EEPROM. Wynika to z faktu, że EEPROM-y mają wyprowadzenia zgodne z pamięciami RAM. Chodzi tu o linię adresową A14, która w EPROM-ach spełnia rolę linii A15. Czasem zachodzi potrzeba emulowania EEPROM, np. gdy budujemy system z tą pamięcią (bez EPROM). Jakie odniesiemy korzyści:

- unikniemy kłopotliwego przekładania pamięci z urządzenia do programatora;
- EEPROM ma ograniczoną liczbę zapisów, a więc będziemy ją oszczędzać;

- zyskujemy na czasie (w EEPROM kasowanie bajtu może trwać nawet 10ms);
- możemy bez kłopotu podejrzeć zawartość EEPROM (wskutek czego wiemy co program zapisuje w pamięci, co jest bardzo pomocne przy uruchamianiu systemu).

Wiem, że można tak zbudować urządzenie i napisać program ładujący, że procesor będzie wczytywał program z komputera do EEPROM w urządzeniu poprzez RS232. Ale, co zrobić, gdy interfejs ten jest

Charakterystyka symulatora:

- ✓ przelotowy port RS232C szybkość transmisji od 2400 do 57600 bodów,
- ✓ maksymalny czas transmisji (64KB) 35 sekund (przy 75600 bodów),
- ✓ współpraca z każdym komputerem wyposażonym w port RS (także AVT-2250!),
- ✓ przyjmowanie danych w formacie INTELHex i binarnym,
- ✓ symulacja pamięci EPROM 2716...27512,
- ✓ symulacja pamięci EEPROM 2816...28256,
- ✓ sygnał RESET do uruchamianego systemu (aktywny stan niski i wysoki),
- ✓ bufony na liniach adresowych i danych.

i wady. Wadą jest to, że urządzenie przyłączone za symulatorem odbierają dane nie przeznaczone dla nich, zaś zaletą to, że można przyłączyć kilka urządzeń do magistrali i sterować nimi równocześnie. No to „nowości“ mamy za sobą.

Tryb ŁADOWANIA

Zakładamy, że procesor jest przyłączony do magistrali. Po otrzymaniu sekwencji przełączającej w tryb ładowania, linia LOAD_EMUL zmieni stan na wysoki (w czasie emulacji, czyli po resecie jest poziom niski). Wyjścia buforów US7, US8, US9 i układu GAL będą przełączone w stan trzeci. Linia RD pamięci US4 i US5 zostanie podciągnięta do poziomu wysokiego przez rezystor R13. Wyjścia zatrasku US3 zostaną uaktywnione. Dzięki temu procesor US2 może zapisywać, odczytywać dane do pamięci RAM. Układy są traktowane jako zewnętrzna pamięć programu. Jak to się dzieje, że najpierw jest przesyłana młodsza część adresu, można poczytać w literaturze o 8051. Do RAM procesor ma dostęp instrukcjami *movx A,@dptr* i *movx @dptr,A*. Tryb ładowania jest włączony tylko w czasie odczytu/zapisu pamięci RAM przez procesor US2.

Tryb ŁADOWANIA jest włączony po otrzymaniu danych w formacie IntelHex i w czasie odczytu po instrukcji @READ lub @MON. W czasie wydawania innych instrukcji (np zmiana typu emulowanej pamięci) symulator znajduje się w trybie EMULACJI. Wyjątkiem jest instrukcja @RESET, po której na 0,5s. urządzenie przechodzi w tryb ŁADOWANIA. Dzięki temu (tak jak podczas zapisu RAM) tranzystory T1, T2 znajdują się w stanie aktywnym, wystawiając sygnał reset do uruchamianego systemu (T1 aktywny niski, T2 aktywny wysoki).

Z zerowaniem trzeba uważać. Jeśli uruchamiany system posiada zewnętrzny układ WATCHDOG lub RESETu ze specjalizowanym układem scalonym, to trzeba sprawdzić czy można dołączyć zewnętrzny układ zerowania. Nie każdy układ to umożliwi i możemy zniszczyć go (tranzystory T1, T2 raczej wytrzymają).

Założmy, że dane zostały wysłane/pobrane z RAM przez procesor i układ przejdzie w tryb EMULACJI. Wtedy to na linii LOAD_EMUL wystąpi poziom niski. Procesor ustawi poziom wysoki na porcie P2 („słabo“ podciągany do +5V) co można traktować jako stan trzeci. P0 pozostanie „pływający“ czyli w stanie trzecim. Wyjścia US3 znajdują się w stanie trzecim. Uruchamiany systemma dostęp do RAM za pośrednictwem buforów US7, US8, US9 i układu GAL. Linia YRD może sterować linią RD układu RAM. Trochę dłużej zatrzymamy się przy linii WR RAM. Jeśli emulowana jest pamięć EPROM, to dostęp do tej linii jest zablokowany (przez wewnętrzne bramki GAL-a). Jeśli natomiast emulujemy EEPROM, to pamięć może być zapisywana sygnałem WR (linia A14 dla EEPROM 2864/28256, linia A11 dla 2816). O buforach US8/US9 nie ma wiele do mówienia. Dzięki bramkom Schmita poprawiane są zbocza sygnałów przechodzących długimi (jak na technikę mikroprocesorową) przewodami. Trochę bardziej zagmatwane jest działanie bufora US7. Brama otwiera się, gdy na wejściu OE i CE sondy emulacyjnej pojawi się poziom niski (na wyjściu YCE GAL-a pojawia się stan niski przekazywany na wejście G bramy). Linia DRI bramy, sterująca kierunkiem transmisji jest sterowana sygnałem WR (zapisu do pamięci RAM). W trybie ŁADOWANIA poziom na wejściu G jest wysoki (za sprawą wyjścia YCE GAL-a), dzięki czemu brama jest nieaktywna i uruchamiany system nie fałszuje danych zapisywanych do RAM. GAL pełni jeszcze jedną ważną funkcję. Zależnie od stanu wejść A, B, C, D przepuszcza, bądź nie sygnały A11..A15 do pamięci RAM. I tak np przy emulowaniu pamięci 2764 na liniach A15, A14 i A13 pamięci jest poziom niski niezależnie od stanu na wejściach adresowych złącza emulacyjnego. Jeśli natomiast wybrany jest tryb emulacji pamięci EEPROM, zamieniana jest funkcja wejścia A15. Wtedy spełnia ono funkcję wejścia A14. Po prostu sygnał A15 ze złącza emulacyjnego jest przesyłany do wejścia A14 pamięci RAM, natomiast na A15 zawsze występuje poziom niski.

W przypadku emulowania układów 2716, 2732 i 2816 sondę emulacyjną należy umieścić w podstawce tak, aby cztery górne wyprowadzenia nie były do niczego podłączone.

Jak widać GAL spełnia wiele funkcji. Gdyby zastąpić go układami TTL trzeba by było ich około 10 szt. Na koniec wspomnę, że w układzie użyto dwie pamięci o pojemności 32KB. Aktywny układ wybiera tranzystor T3 pełniący funkcję inwertera sterując naprzemiennie wejścia CE. Układ US10 steruje diodami LED. Stała czasowa RC jest tak dobrana, że w czasie zapisu/odczytu diody D2, D3 nie migają, lecz świecą pełną jasnością.

Ślawomir Skrzyński

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/pcb.html> oraz na płycie CD-EP06/2000 w katalogu PCB.

WYKAZ ELEMENTÓW

Kondensatory

C1, C4, C7: 47μF
C2, C3, C8, C13..C18: 100nF
C5, C6: 27pF
C9..C12, C19: 10μF
C20, C21: 22nF

Rezystory

R1, R3, R5, R6, R14: 10kΩ
R2, R13: 1kΩ
R4: 2,2kΩ
R7..R10: 470Ω
R11, R12: 100kΩ

Półprzewodniki

D1..D4: LED
D5, D6: 1N4148
T1, T3: BC238
T2: BC308
US1: 7805
US2: AT89C51
US3: 74HC573
US4, US5: 62256
US6: GAL20V8
US7..US9: 74HC245
US10: 74HCT04
US11: 74HC157
US12: MAX232

Różne

PK1: EDR 101C-5V
Q1: 11.059MHZ
G1: DB9M kątowne do druku
G2: DB9F kątowne do druku
Z-WS34G