

Pamięci EEPROM w systemach mikroprocesorowych, część 2

Tym artykułem kończymy prezentację sposobów programowania szeregowych pamięci EEPROM. Ponieważ największą popularnością cieszą się wśród użytkowników pamięci z interfejsem I2C, przedstawiamy przykładową procedurę ich obsługi, napisaną w asemblerze MCS-51.

Interesującym uzupełnieniem artykułu jest opis programatora szeregowych pamięci EEPROM, który znajduje się na str. 47.

EEPROM-y szeregowy z magistralą 3-przewodową

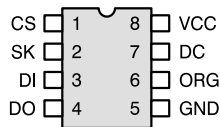
Powierzchnia płytki półprzewodnikowej zawierająca strukturę pamięci stanowi ułamek powierzchni obudowy z jej wieloma wyprowadzeniami adresowymi i danych. W celu zmniejszenia miejsca zajmowanego przez pamięć i redukcji liczby połączeń niezbędnych do komunikacji między systemem a pamięcią, opracowane zostały EEPROM-y, w których adres i dane przesyłane są szeregowo, bit za bitem.

Do rodziny układów z magistralą 3-przewodową należą EEPROM-y serii 93C46/57/66 o pojemnościach 1/2/4kb. Schemat wyprowadzeń tych układów pokazano na rys. 6, a strukturę blokową na rys. 7.

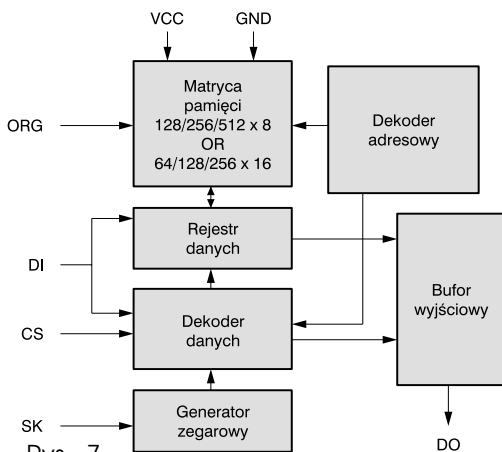
Redukcja liczby wyprowadzeń i szeregowy sposób transmisji spowodowały komplikację struktury układów w porównaniu z EEPROM-ami równoległymi. Także współpraca z pamięcią jest bardziej skomplikowana. W pamięci dane wejściowe i wyjściowe są rozdzielone i pojawiają się na różnych wyprowadzeniach układu. Poszczególne wyprowadzenia mają następujące funkcje:

- VCC i GND - zasilanie układu;
- CS - wybór układu, stan wysoki na tym wejściu uaktywnia pamięć;
- SK - wejście zegarowe, synchronizujące wprowadzanie i wyprowadzanie danych z pamięci;
- DI - wejście bitów rozkazu, adresu i danych;
- DO - wyjście danych i statusu pamięci;
- ORG - wejście sterujące wewnętrzną organizacją danych;
- DC - wejście niepodłączane.

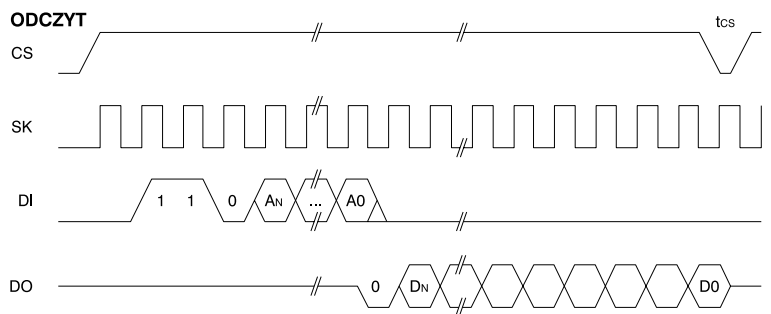
Wyjaśnienia wymaga funkcja wyprowadzenia ORG. Otóż układy serii 93Cxx traktują dane w sposób zależny od potencjału dołączonego do wyprowadzenia ORG. Jeżeli wyprowadzenie jest zwarte do masy, dane są traktowane jako słowa 8-bitowe i pojemność pamięci 1K wynosi 128 słów. Jeżeli wyprowadzenie podłączone zostanie do napięcia zasilającego, dane zapisywane są jako słowa 16-bitowe i pojemność pamięci zmniejsza się do 64 słów. Wybór trybu powoduje zmniejszenie lub zwiększenie przestrzeni adresowej pamięci i zmianę liczby bitów adresu (dla 93C46 i formatu słowa 8-bitowego, adres składa się z 7 bitów, a dla 16-bitowego formatu da-



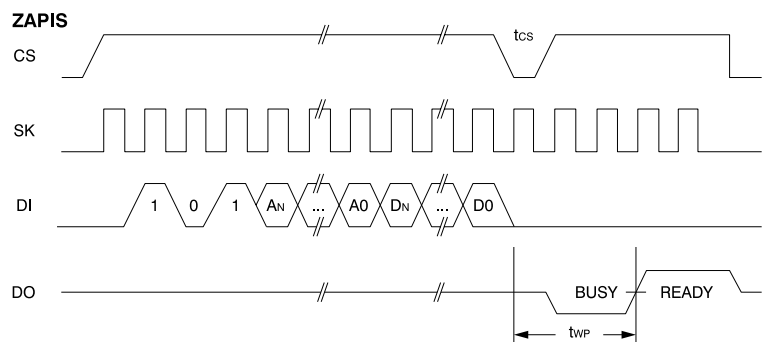
Rys. 6.



Rys. 7.



Rys. 8.



Rys. 9.

nych adres składa się z 6 bitów).

Działaniem pamięci steruje siedem instrukcji. Każdej instrukcji przypisany jest 2-bitowy kod i adres, wysyłane szeregowo do pamięci poprzez wyprowadzenie DI. W tabeli 1 zawarto zestawienie poszczególnych instrukcji, ich kodów i powiązanych z instrukcjami adresów.

Najprostszą operacją jest odczyt danych z pamięci. Schemat przebiegów na liniach sterujących podczas realizacji tej instrukcji pokazano na rys. 8. Przed uaktywnieniem układu pamięci przez podanie stanu wysokiego na wyprowadzenie CS, linie SK i DI powinny być na poziomie niskim. Następnie jest generowany cykl rozbiegowy zegara, a potem na wejściu DI pojawia się obowiązkowo stan 1. Bit jedynki musi poprzedzać każdą instrukcję wysyłaną do pamięci! Zapis bitów z linii DI jest dokonywany narastającym zboczem impulsu zegarowego. Po zapisaniu bitu 1 do pamięci jest wysyłany kod instrukcji, w tym przypadku 10, a następnie siedem bitów adresu komórki pamięci, której zawartość chcemy odczytać. Bity adresu są wysyłane od najstarszego do najmłodszego. Wpisanie ostatniego bitu adresu spowoduje pojawienie się na wyjściu DO poziomu niskiego, poprzedzającego wysłanie zawartości odczytywanej komórki pamięci EEPROM. Następnie, synchronicznie z narastającym zboczem impulsu zegarowego, pojawi się 8 bitów danych, począwszy od bitu najstarszego. Kolejne bity danych na wyjściu DO pojawiają się z opóźnieniem kilkuset nanosekund w stosunku do zbocza impulsu zegarowego. Po zakończeniu odczytu danych wyjście CS, co najmniej na 1µs, powinno znaleźć się na poziomie niskim przed inicjacją kolejnej instrukcji. Częstotliwość impulsów zegarowych nie powinna przekroczyć 1MHz.

Obsługa pamięci, przynajmniej na początku, nie należy do najłatwiejszych. Pewne czynności, które układy równoległe EEPROM wykonują automatycznie, tutaj muszą być przeprowadzone przez system mikroprocesorowy przy pomocy odpowiednich instrukcji. Tak jest w przypadku wszelkich manipulacji związanych z zapisem do pamięci. Po włączeniu zasilania pamięć automatycznie wchodzi w tryb ochrony przed zapisem. Dlatego, zanim cokolwiek będzie można zapisać do pamięci,

Listing 2.

```

;*****
;* Procedura I2C obsługi pamięci EEPROM 24C02
;* dla procesora z rodziny '51 zegar do 12MHz
;* bajty i bity ustawiane przed wywołaniem
;* procedury I2C
;* ADRES - adres układu do zapisu/odczytu
;* przez procedurę
;* SUBADR - subadres komórki, od której zacznie
;* się zapis/odczyt
;* BUF_ADDR -adres bufora danych w pamięci RAM
;* procesora
;* BUF_SPACE - długość bufora danych w pamięci
;* RAM procesora
;* R_W - bit 0 procedura zapisze dane
;* z bufora do pamięci EEPROM
;* 1 procedura odczyta dane z pamięci EEPROM do
;* bufora po zakończeniu procedury do
;* akumulatora i rejestru R7 wpisywana jest
;* wartość:
;* 00h -gdy procedura zakończyła się sukcesem
;* FFh -gdy wystąpił błąd
;* bajty i bity używane przez procedurę:
;* end_trans - bit flaga zakończenia
;* transmisji gdy 1
;* subadr_trans - bit flaga nadania subadresu
;* gdy 1
;* wait - bajt pętla 10ms
;* counter - bajt licznik bitów
;*****
I2c:
mov r1,buf_adr
mov r2,buf_space
jnb r_w,i2cw
jmp i2cr

; zapis do pamięci eeprom
i2cw: clr end_trans
;wsk.zakończenia transmisji
clr subadr_trans ;wsk.nadania subadresu
call i2cwl
ret

i2cwl: call i2ctest
jc blad
clr c
call i2cstart
iw31: clr a
cjne a,wait,iw22
jmp blad
iw22: jnb subadr_trans,iw24
;czy nadany subadres?
mov a,@r1 ;kolejny znak z bufora do A
mov counter,#9
iw27: jnb scl,iw27
jb end_trans,iw28
inc r1
djnz r2,iw3
setb end_trans
jmp iw3

iw28: call i2cend
ret
iw24: mov a,subad ;transmisja subadresu
setb subadr_trans
mov counter,#9
iw26: jnb scl,iw26
iw3: call sir3
jmp iw31

blad: setb sda ;błąd zapisu/odczytu układu
setb scl
mov a,#0ffh
;procedura wraca z wartością FFh w A i R7
mov r7,a
ret

;odczyt z pamięci
i2cr: setb end_trans
clr subadr_trans
call i2cwl ;adres i subadres do eeprom
cjne a,#0ffh,i2cr2
ret

i2cr2: call i2c10ms

call i2ctest
jc blad
setb c
call i2cstart
clr a
cjne a,wait,ir22
jmp blad
ir22: mov counter,#8
ir24: clr scl ;SCL 1
call i2ctakt

setb scl ;SCL h
call i2ctakt
mov c,sda
rlc a
djnz counter,ir24
inc r1
djnz r2,ir4
setb scl ;SCL h
call i2cend
ret

ir4: clr sda ;ACK
call i2ctakt
setb scl ;SCL h.9
call i2ctakt
clr scl ;SCL 1
call i2ctakt
setb sda ;SDA h
call i2c10ms
jmp ir22

i2ctakt: ;opóźnienie
nop
nop
nop
ret

i2ctest:
mov wait,#080h
i2ct1: jnb sda,i2ct3 ;SCL i SDA wolne?
jnb scl,i2ct3
i2ct5: clr c ;SCL i SDA wolne
ret
i2ct3: mov counter,#0ffh
djnz counter,$
djnz wait,i2ct1
jnb sda,i2ct4
jnb scl,i2ct4
jmp i2ct5
i2ct4: setb c ;błąd SCL i SDA nadal zajęte
ret

i2cstart:
mov a,adres
rlc a
mov counter,#9
clr sda ;start
call i2ctakt
sir3: clr scl ;SCL 1
call i2ctakt
rlc a
djnz counter,sir2 ;adres układu
jmp sir21
sir2: mov sda,c
call i2ctakt
setb scl ;SCL h
call i2ctakt
jmp sir3

sir21: setb sda ;SDA h
call i2ctakt
setb scl ;SCL h
call i2ctakt
mov wait,#080h
sir23a: jnb sda,sir22
mov counter,#0ffh
djnz counter,$
djnz wait,sir23a
sir22: ret

i2cend:
call i2ctakt
clr scl ;SCL 1
call i2ctakt
clr sda ;SDA 1 ACK
call i2ctakt
setb scl ;SCL h
call i2ctakt
setb sda ;END
call i2ctakt
clr a
mov r7,a
ret

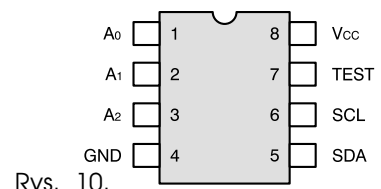
i2c10ms:
mov wait,#20 ;pętla 10ms
i10m: mov counter,#0ffh
djnz counter,$
djnz wait,i10m
ret

end

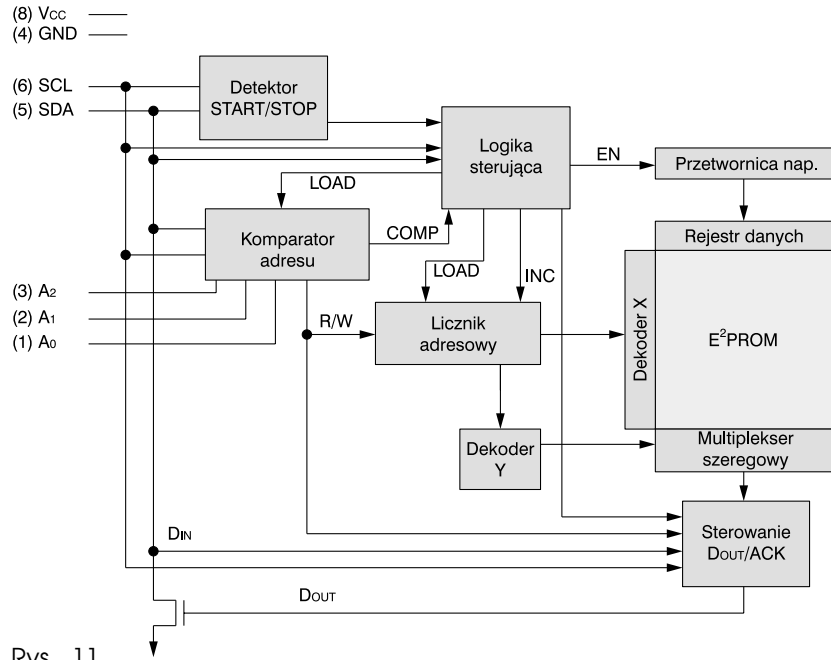
```

musi być wprowadzona instrukcja EWEN, dopiero wtedy można zmienić zawartość pamięci. Jednak zapis nowych danych do komórki, w której już coś jest, wymaga wymazania jej zawartości. Wiąże się to z użyciem (wprowadzeniem) instrukcji ERASE. W opisywanych pamięciach równoległych podobne wymazanie zawartości komórki przed nowym

zapisem jest wykonywane automatycznie. W dodatku, każda zmiana zawartości pamięci wymaga pewne-

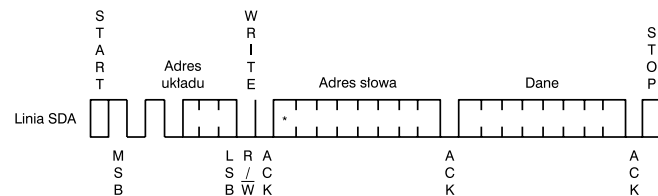


Rys. 10.

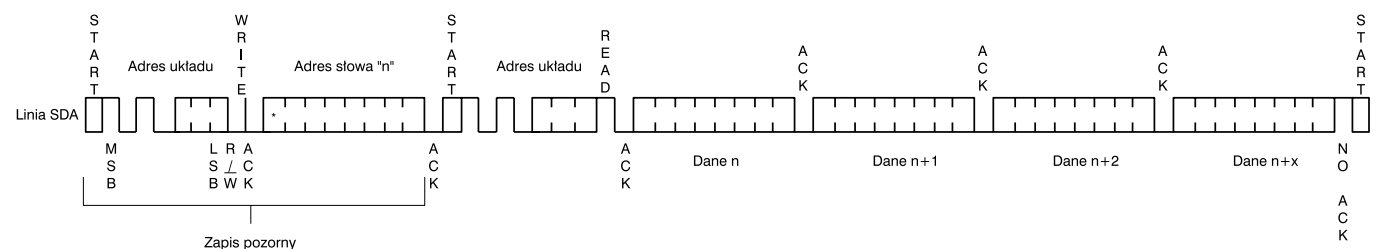


Rys. 11.

go czasu, związanego z cyklem zapisu i może trwać do 10ms. Przed upływem tego czasu pamięć nie może wykonać kolejnej instrukcji. W celu stwierdzenia, czy cykl zapisu został zakończony, po każdej instrukcji zmieniającej zawartość pamięci (ERASE, WRITE, ERAL, WRAL) i podaniu na wejście CS impulsu ujemnego jest badany stan wyjścia DO. Jeżeli stan tego wyjścia będzie niski, to oznacza, że cykl zapisu nie został jeszcze zakończony. Pojawienie się poziomu wysokiego oznacza, że pamięć może wykonać następną instrukcję. Dopiero wtedy można zapisać dane do pamięci przy pomocy instrukcji WRITE. Przebiegi czasowe impulsów związanych z tą instrukcją pokazano na rys. 9.



Rys. 12.



Rys. 13.

EEPROM-y szeregowe z interfejsem I²C

Standard magistrali I²C odniósł prawdziwy sukces na rynku elektroniki. Coraz większa liczba układów scalonych jest wyposażana w interfejs umożliwiający im współpracę z tą szeregową magistralą, wykorzystywaną do sterowania i wymiany danych. Do zalet należy mała liczba linii sterujących (linia SCL zegara i SDA danych), dobrze opisany i przejrzysty standard, a także możliwość dołączania do magistrali wielu układów.

Z tych powodów powstała i rozwija się rodzina pamięci EEPROM z interfejsem I²C. Należą do niej popularne układy z rodziny 24C01/04/08/16/64, gdzie liczba po literze „C” określa pojemność pamięci w tysiącach kilobitów.

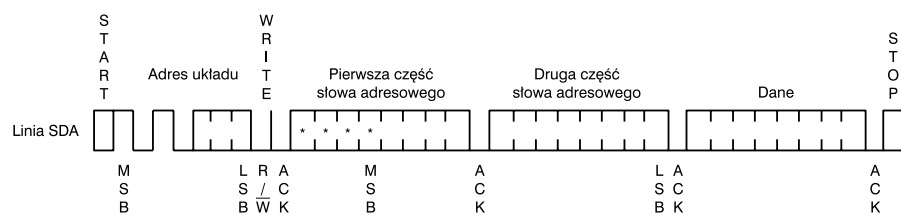
Tabela 1.

Nazwa	Kod	Adres dla formatu słowa 8-bitowego
READ (odczyt 1 słowa danych)	10	A6..A0
EWEN (zezwolenie na zapis)	00	11xxxx
ERASE (wymaż komórkę pamięci)	11	A6..A0
WRITE (zapisz 1 słowo danych)	01	A6..A0
ERAL (wymaż całą pamięć)	00	10xxxx
WRAL (zapisz całą pamięć wzorem)	00	01xxxx
EWDS (zablokowanie zapisu)	00	00xxxx

(x -wartość bitu dowolna)

Rozkazy ERAL i WRAL odnoszą się do zawartości całej pamięci i pozwalają na jej globalne skasowanie i zapisanie określonym wzorem danych. Jest to pomocne podczas testowania pamięci. Ostatni z rozkazów EWDS powoduje sytuację analogiczną jak po włączeniu napięcia zasilającego: wszelkie zmiany w wartości pamięci EEPROM są niemożliwe.

Na rys. 10 i 11 pokazano rozkład wyprowadzeń i schemat blokowy tych pamięci. Na zewnątrz obudowy oprócz wyprowadzeń zasilania, linii magistrali SCL i SDA, wyprowadzono także trzy linie adresowe i końcówkę TEST (u niektórych producentów jest nieczynna). Redukcja linii sterujących i rozbudowany protokół transmisji spowodowały komplikację struktury wewnętrznej pamięci, która musi być wyposażona w układy logiki interpretującej dane napływające magistralą I²C. Na szczęście budowa wewnętrzna układu nie jest kłopotem użytkownika i programisty, tylko konstruktora układu scalonego.



Rys. 14.

W celu jednoznacznej identyfikacji układu EEPROM, do którego dane mają być zapisane lub odczytane, każda kostka pamięci dołączona do magistrali I²C ma przypisany unikatowy adres. Adres to słowo składające się z ośmiu bitów. Cztery najstarsze bity określają grupę, do której należy dany układ scalony (w tym przypadku pamięć), trzy kolejne bity adresują konkretną kostkę pamięci ze wszystkich dołączonych do magistrali, a stan bitu najmłodszego określa rodzaj operacji (0 - zapis, 1 - odczyt).

b7 b6 b5 b4 b3 b2 b1 b0
1 0 1 0 A2 A1 A0 R/W

Bit A2..0 są powiązane z nogami adresowymi o takich samych symbolach. Jeżeli wszystkie trzy wyprowadzenia adresowe układu zostaną zwarte do masy, pamięć będzie odpowiadać tylko w przypadku, jeżeli w bajcie adresu bity b3..1 będą wyzerowane itd. Wynika z tego, że do magistrali może być jednocześnie podłączonych 8 pamięci (choć istnieją pewne ograniczenia, o czym za chwilę).

Przebieg sygnałów na linii SDA w czasie operacji zapisu bajtu do pamięci EEPROM i odczytu sekwencji bajtów z pamięci pokazano na rys. 12 i 13. Z wyjątkiem sekwencji START i STOP sygnał na linii SDA może ulec zmianie tylko wtedy, gdy linia SCL jest na poziomie niskim. W czasie zapisu bajtu danych do pamięci jest wysyłana najpierw sekwencja START i adres pamięci z wyzerowanym najmłodszym bitem b0. Aktywna pamięć potwierdza fakt odebrania adresu sygnałem ACK. Robi to w ten sposób, że zwiera do masy linię SDA w czasie trwania pierwszego impulsu zegarowego po odbiorze adresu. Następnie jest transmitowany WORD ADRES lub SUBADRES będący binarnym adresem komórki pamięci, która ma być zapisana. Jeżeli, jak w przypadku EEPROM-ów 24C01A, pojemność pamięci wynosi tylko 128 bajtów, najstarszy bit subadresu jest ignorowany. Potwierdzeniem odbioru subadresu jest znów sygnał ACK genero-

wany przez pamięć. Wreszcie wysyłany jest bajt danych, a po sygnale ACK transmisja jest kończona przez sekwencję STOP. W przypadku odczytu danych z pamięci procedura jest tylko trochę bardziej skomplikowana i składa się jak gdyby z dwóch części. Najpierw jest wysyłany do pamięci adres z subadresem określającym pierwszą komórkę, od której rozpocznie się odczyt. Następnie, bezpośrednio po sygnale ACK jest generowana sekwencja START i ponownie wysyłany adres pamięci, tym razem z ustawionym najmłodszym bitem b0. Zaadresowana do odczytu pamięć EEPROM potwierdza swoją gotowość sygnałem ACK i zaraz potem w takt impulsów zegarowych wysyła zawartość pierwszej komórki pamięci. Tym razem to urządzenie odczytujące wysyła sygnał ACK i wtedy możliwy jest odbiór kolejnego bajtu z pamięci. Liczba odczytywanych danych jest ograniczona tylko pojemnością pamięci. Zakończenie transmisji następuje przez pominięcie potwierdzenia ACK i wygenerowanie przez układ odczytujący sekwencji STOP.

Taki sposób zapisu i odczytu odnosi się do pamięci, których pojemność nie przekracza 256 bajtów i które mogą być bezpośrednio adresowane przez subadres. Jak jednak wygląda współpraca z pamięciami o większych pojemnościach? Otóż kostki te „udają”, że mają w środku większą liczbę pamięci o pojemności 256 bajtów, a do ich adresowania są używane bity b3..1 adresu. Z tego powodu w układzie 24C04 jego wyprowadzenie A0 musi pozostać niepodłączone, a liczba pamięci współpracujących z magistralą I²C jest ograniczona do 4. W przypadku układu 24C16 wszystkie wyprowadzenia A0..2 pozostają wolne, a urządzenie może adresować tylko jedną pamięć tego typu.

Producenci oferują już pamięci EEPROM (24C32/64), których pojemność wykracza poza opisane wcześniej możliwości adresowania. Z tej właśnie przyczyny WORD ADRES

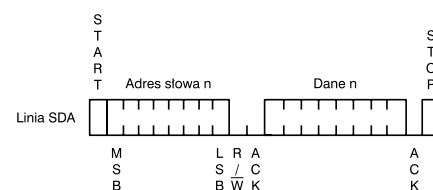
został podzielony na dwa bajty. Przebieg zapisu bajtu do tego typu pamięci pokazano na rys. 14.

Nie koniec jednak zamieszania wynikającego ze sposobu adresowania pamięci! Na rynku są dostępne także pamięci oznaczane jako 24C01 o pojemności 128 bajtów, których wyprowadzenia A0..2 pozostają nieaktywne. Zastosowanie takiego układu eliminuje możliwość dołączenia dodatkowych pamięci EEPROM do magistrali I²C, ponieważ sposób komunikacji z tym układem jest uproszczony i niekompatybilny z resztą rodziny.

Zapis bajtu do takiej pamięci pokazano na rys. 15. Wynika z niego, że po sekwencji START jest pomijany adres, a transmitowany zostaje bezpośrednio subadres wskazujący komórkę, do której zapisany zostanie bajt danych. Ze sposobu adresowania wynika, że pojedyncza pamięć EEPROM może być w układzie bez kłopotu zamieniona na pamięć o większej pojemności, z wyjątkiem układów 24C01, 24C32/64 różniących się sposobem komunikacji z otoczeniem.

Tak jak w przypadku EEPROM-ów szeregowych, pamięci z rodziny 24Cxx wyposażono w możliwość zapisu stronicowanego. Oznacza to, że jednorazowo można zapisać do pamięci blok danych o wielkości 4 lub więcej bajtów. Wielkość strony zależy od pojemności pamięci i producenta. Nie mając dostępu do danych katalogowych można jednak bezpiecznie przyjąć, że dla pamięci powinno być możliwe jednorazowe zapisanie co najmniej 4 bajtów. Każdy cykl zapisu (bajtu lub strony) może trwać do 10ms. Przed upływem tego czasu EEPROM może nie wykonać prawidłowo kolejnych operacji zapisu lub odczytu.

Należy także wspomnieć o znaczeniu wyprowadzenia oznaczanego jako TEST. Czasami jest ono nieaktywne, czasami jednak pełni funkcję sprzętowej blokady zapisu do pamięci. Jeżeli zostanie ono podłączone do poziomu wysokiego, zapis danych do pamięci będzie niemożliwy.



Rys. 15.

Na list. 2 zamieszczona została procedura assemblerowa obsługi pamięci 24C02 przez procesor z rodziny '51, taktowany zegarem do 12MHz. Procedura nie jest może zbyt elegancko napisana i zoptymalizowana, ale jej zaletą jest to, że „żyje”, ponieważ pochodzi z funkcjonującego programu. Jeżeli chochlik drukarski czegoś nie popsuje, procedura po dokładnym skopiowaniu powinna od razu zadziałać. Można także łatwo ją zmodyfikować do obsługi innych pamięci z interfejsem I²C.

Przed wywołaniem procedury w programie trzeba zdefiniować porty procesora, które mają pełnić rolę końcówek SDA i SCL. Należy także zadeklarować kilka bajtów i bitów współpracujących z procedurą. Przepisanie wyjść oraz adresów bajtów jest dowolne i zależy od programu, w którym procedura ma pracować. Przykładowa deklaracja może wyglądać następująco:

```
scl      bit p1.3    ;linia SCL
sda      bit p1.2    ;linia SDA
```

```
adres      equ 30h
subadr     equ 31h
buf_adr    equ 32h
buf_space  equ 33h
wait       equ 34h
counter    equ 35h
r_w        bit 0h
end_trans  bit 1h
subadr_trans bit 2h
```

Wywołanie procedury poprzedza ustawienie parametrów określających jej pracę. Trzeba podać adres układu, subadres komórki, ustawić bit r_w określający rodzaj operacji (0 -zapis, 1 -odczyt) oraz podać początek bufora w pamięci RAM procesora i jego długość. Z bufora będą zapisywane dane do pamięci EEPROM lub podczas odczytu w buforze będą zapisywane dane z układu EEPROM. Przykładowe wywołanie procedury I2c z programu głównego może wyglądać następująco:

```
                ;odczyt danych z EEPROM
mov adres,#50 ;adres pamięci EEPROM
mov subadr,#0 ;odczytane będą dane począwszy od subadresu =0
```

```
mov buf_adr,#adres_bufora
                ;adres bufora w pamięci RAM procesora
mov buf_space,#8
                ;odczytanych będzie 8 bajtów
                ;z pamięci EEPROM
setb r_w        ;ustawienie flagi odczytu
call I2c        ;wywołanie procedury
```

Jeżeli odczyt się powiedzie, procedura wraca z wartością #0 w akumulatorze i rejestrze R7. Jeżeli nastąpił błąd (np. pamięć jest uszkodzona), w rejestrach znajdzie się wartość #FFh. Korzystanie z procedury musi uwzględniać ograniczenia związane z funkcjonowaniem pamięci EEPROM takie jak zapis do pamięci bajtu, strony, czas trwania cyklu zapisu itd. W przypadku niektórych assemblerów może być konieczna zmiana składni zapisu procedury w celu dostosowania do wymagań assemblera.

Ryszard Szymaniak, AVT

Program z list. 2 jest dostępny na internetowej stronie EP, pod adresem: www.avt.com.pl/avt/ep/ftp/.