

Rys. 10. Różne możliwości sterowania wejść mikrokontrolera

niem wskutek uszkodzenia tranzystora - zwarcie w tranzystorze w obwodzie kolektor-baza spowoduje pojawienie się napięcia zasilającego obciążenie na wyprowadzeniu mikrokontrolera. Napięcie to może być wyższe od napięcia zasilania układu mikrokontrolera, co może doprowadzić do uszkodzenia napięciowego linii lub - jeśli na danym wyprowadzeniu będzie stan niski - może dojść do przeciążenia prądowego. Zastosowany rezystor ograniczy ten prąd oraz działał będzie jak dzielnik i ograniczy napięcie na wyprowadzeniu portu. Na **rys. 9c** przedstawiono sposób sterowania wejściami cyfrowych układów scalonych. Dla układów CMOS rezystor podciągający jest niezbędny, natomiast dla układów TTL nie jest on konieczny, lecz na pewno nie zaszkodzi (zwłaszcza, że ktoś w urządzeniu może zastosować odpowiednik układu wykonany w technologii CMOS). Port P0 może wysterować według danych producenta do ośmiu wejść układów TTL, co pozwala obciążyć każdą końcówkę w stanie niskim prądem około 13 mA i z powodzeniem wystarcza do sterowania diodą LED (jak np. na **rys. 9d**).

Wykorzystując linie portu P0 jako wejścia musimy także rozważyć konieczność zastosowania w razie potrzeby rezystorów podciągających. Rezystory takie są niezbędne jeżeli do mikrokontrolera chcemy podpiąć elementy stykowe lub układy cyfrowe (czy pojedyncze tranzystory) pracujące z wyjściami typu otwarty kolektor (dren). Sytuację taką przedstawiono na **rys. 10a**). Wartości rezystorów R1 i R2 powinny być tak dobrane, aby nie przeciążać wyjść dołączonych układów. Zastosowany rezystor R4 zapewnia stan niski na wejściu P0.4 gdy przycisk SW1 jest zwolniony, natomiast rezystor R3 zabezpiecza mikrokontroler przed przypadkowym zwarciem mogącym nastąpić, gdyby linia P0.4 została wyzerowana programowo. Jak widać wciśnięcie przycisków SW1 i SW2 wywołuje różne stany na wejściach mikro-

kontrolera - w zależności od potrzeby możemy wywoływać stan wysoki (SW1) lub niski (SW2). Rezystorów podciągających nie musimy stosować gdy układ współpracuje z elementami posiadającymi wyjścia komplementarne (układy cyfrowe, komparatory), co pokazano na rysunku 9b). Warto wspomnieć, że port P0 pracujący jako wejście bez zastosowania rezystorów podciągających posiada bardzo dużą impedancję wejściową (zbliżoną do impedancji tranzystora MOS), co umożliwi sterowanie ze źródeł o bardzo małej wydajności.

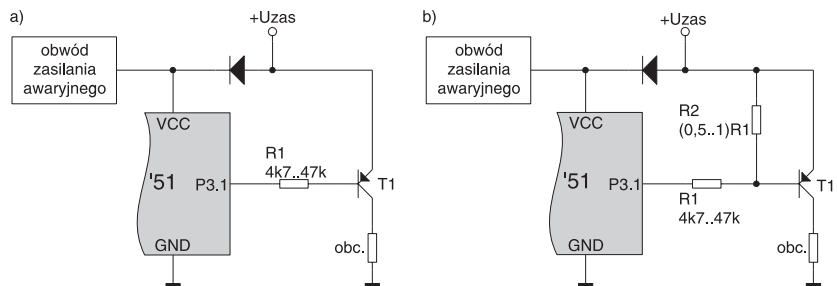
Pozostałe porty mikrokontrolera (P1 do P3) poza faktem posiadania wewnętrznych rezystorów podciągających (w rzeczywistości są to tranzystory MOS pracujące jako źródła prądowe - można je traktować jako rezystory podciągające o wartości 50...100 kΩ), posiadają także mniejszą wydajność prądową, odpowiadającą wysterowaniu czterech wejść układów TTL (maksymalny prąd w stanie niskim to około 6,5 mA). Schematy dołączeń układów peryferyjnych nie różnią się od tych podanych dla portu P0 (rys. 9 i 10), za wyjątkiem braku konieczności stosowania rezystorów podciągających.

Analizując właściwości portów P1 i P3, w które są wyposażone mikrokontrolery AT89C051 należy przedstawić kilka istotnych różnic wpływających na sposób ich wykorzystania do obsługi urządzeń zewnętrznych. Przede wszystkim należy wspomnieć o właściwościach

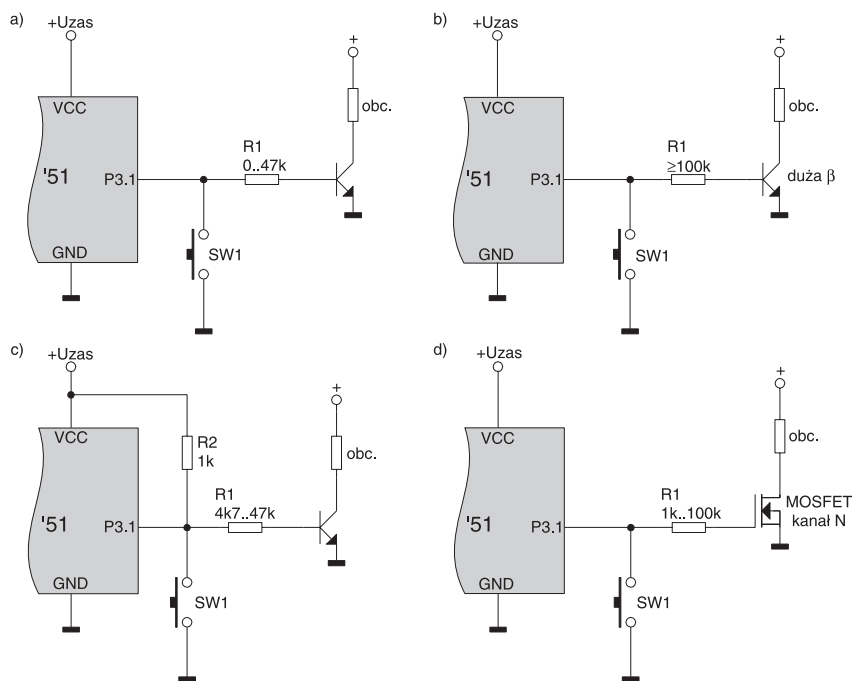
prądowych portów: zarówno linie portu P1 jak i P3 mogą w stanie niskim przeprowadzić prąd do 20 mA, z tym, że według zaleceń producenta, całkowity prąd wpływający jednocześnie do wszystkich wykorzystywanych linii portów nie powinien przekroczyć 80 mA - powyżej tej wartości firma Atmel nie gwarantuje deklarowanej wydajności prądowej. Producent przewidział możliwość sterowania przez port P1 katodami wyświetlaczy LED, jednak także port 3 może z powodzeniem sterować diody LED pod warunkiem zapewnienia sumarycznego prądu wszystkich wyprowadzeń P1 i P3 (będących w stanie niskim) na poziomie nie przewyższającym 80 mA. W stanie wysokim porty mają wydajność taką jak porty w „dużych“ Atmelach. Dodatkowo trzeba pamiętać, że dwie linie portu P1: P1.0 i P1.1 nie posiadają wewnętrznych rezystorów podciągających, gdyż zakłócałyby one pracę wewnętrznego komparatora podłączonego do tych linii. Jeżeli chcemy linie P1.0 i P1.1 wykorzystywać jako zwykłe we/wy cyfrowe, to należy stosować zasady dołączania zewnętrznych urządzeń takie jak w przypadku portu P0. W spisie wyprowadzeń układów AT89C051 nie znajdziemy linii P3.6 - linia ta jest wewnętrznie połączona z wyjściem komparatora i może być tylko odczytywana (stan odzwierciedla stan komparatora).

Przedstawiony wyżej problem stosowania rezystorów podciągających to tylko jeden z wielu możliwych błędów popełnianych przy podłączaniu do mikrokontrolera układów współpracujących. Na **rys. 11, 12, 13 i 14** przedstawiono najczęściej popełniane błędy konstrukcyjne, które uniemożliwiają poprawną pracę układu lub pogarszają jego pewność działania. Rysunki oznaczone literami **a** przedstawiają błędnie zaprojektowany układ, a opisane jako **b** pokazują poprawne rozwiązanie lub sposób naprawy błędnie zaprojektowanego układu.

Na **rys. 11a** przedstawiono układ sterowania tranzystorem PNP z wyjścia mikrokontrolera wyposażonego w awaryjny układ zasilania. Po głębszej analizie łatwo zauważyć, że układ może działać niestabilnie, objawiając się niecałkowitym wyłączeniem tranzystora. Następować to będzie wskutek obecności diody należącej do obwodu zasilania awaryjnego. Wskutek spadku napięcia na tej diodzie, napięcie wyjściowe w stanie



Rys. 11. Przykłady błędnie zaprojektowanych wzmacniaczy wyjściowych



Rys. 12. Kolejne przykłady błędnie zaprojektowanych obwodów wyjściowych

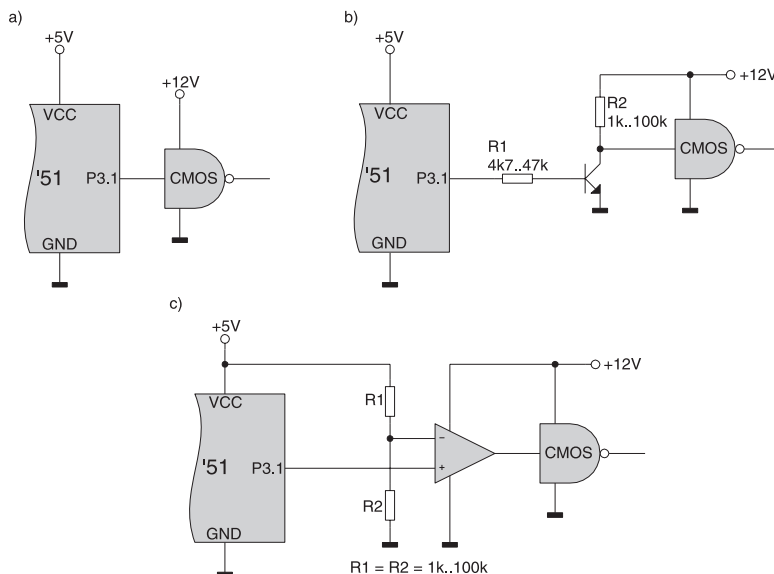
wysokim nie będzie miało wartości napięcia zasilania, lecz będzie obniżone o 0,6...0,8 V, co może doprowadzić do przewodzenia złącza baza-emiter tranzystora i jego otwarcia. Przy małych prądach obciążenia wystarczy to zapewne do nasycenia tranzystora i sytuacji, w której nie będzie można wyłączyć programowo sterowanego w ten sposób urządzenia. Rozwiązanie problemu może przynieść zamiana diody krzemowej na diodę Schottky'ego (mniejszy spadek napięcia), lepiej jednak zastosować dodatkowy rezystor R2 o wartości mniejszej od R1, polaryzujący bazę tranzystora. Układ taki przedstawiono na rys. 11b. Rezystory R1 i R2 będą tworzyły dzielnik napięcia obniżając o połowę (lub więcej) napięcie na bazie tranzystora i nie dopuszczając do jego przewodzenia wskutek różnicy napięć wywołanych spadkiem na diodzie.

Na rys. 12a przedstawiono układ, który według założenia powinien działać następująco: mikrokontroler steruje tranzystorem włączając i wyłączając jakieś urządzenie. Jeśli urządzenie jest włączone, to naciśnięcie przycisku ma je wyłączyć. Układ jednak nie będzie działał zgodnie z założeniami. Jeżeli mikrokontroler wyzeruje linię P3.1, to jasne jest, że urządzenie zostanie wyłączone. Jeżeli natomiast na wyjściu P3.1 pojawi się stan „1”, to tranzystor zacznie przewodzić i włączy urządzenie - tak układ działałby, gdyby program nie wykrywał wciśnięcia przycisku. W układzie jak na rysunku mikrokontroler włączy urządzenie natychmiast po włączeniu. Będzie to spowodowane faktem, że wewnętrzny rezystor podciągający ma dość dużą wartość (50...100 kΩ) i będzie tworzył z rezystorem w obwodzie bazy tranzystora dzielnik dający

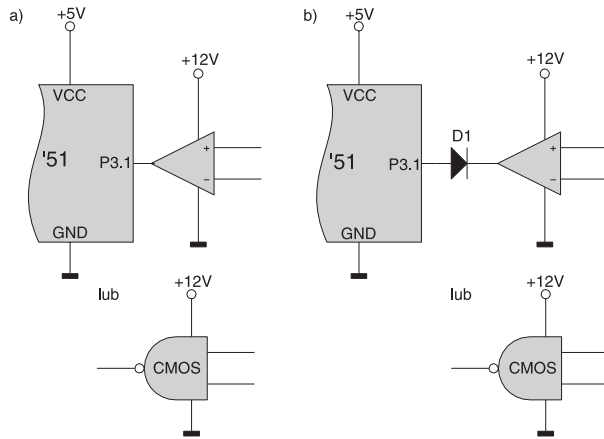
na wyprowadzeniu mikrokontrolera napięcie o wartości ok. 2 V (z uwzględnieniem napięcia baza-emiter) lub mniejsze, co będzie interpretowane jako stan „0” - czyli wciśnięcie przycisku. Problem rozwiązać można dobierając dużą wartość rezystora R1 - powyżej 100 kΩ, co wraz z napięciem baza-emiter tranzystora powinno dać na wyprowadzeniu mikrokontrolera napięcie około 3 V, które powinno być zinterpretowane jako „1” (rys. 12b). Wadą takiego rozwiązania jest konieczność zastosowania tranzystora o dużym wzmacnieniu (jeżeli chcemy uzyskać znaczny prąd obciążenia). Na rys. 12c przedstawiono lepsze rozwiązanie - zastosowanie dodatkowego rezystora podciągającego R2 o wartości znacznie mniejszej (przynaj-

mniej kilkukrotnie) niż rezystor R1. Takie rozwiązanie nie posiada wady jak poprzednie, ale konieczny jest dodatkowy (choć tani) element. Problem można rozwiązać jeszcze inaczej, stosując układ z rys. 11d. Zastosowano tutaj układ z tranzystorem MOSFET, który jest elementem sterowanym napięciowo i wartość rezystancji R1 nie odgrywa tutaj znaczącej roli (przy małych częstotliwościach przełączeń). Rezystor R1 pełni tutaj funkcję zabezpieczającą podobnie jak na rys. 9b. Wadą takiego rozwiązania jest nieco wyższa niż w przypadku tranzystorów unipolarnych cena tranzystorów MOSFET oraz fakt dość wysokiego napięcia progowego tranzystora, co praktycznie uniemożliwia stosowanie tego rozwiązania w układach z zasilaniem mikrokontrolera napięciem niższym niż 5 V.

Przy budowie nowych układów szczególną uwagę projektanta powinien zwrócić fakt dołączania do mikrokontrolera urządzeń zasilanych innym (zazwyczaj wyższym) napięciem. Niestety także tutaj można popełnić podstawowy błąd uwidoczniiony na rys. 13a. W układzie tym wyjście mikrokontrolera steruje układem cyfrowym zasilanym napięciem 12 V. Układ nie będzie działał z bardzo prostego powodu - w układach cyfrowych CMOS próg przełączania znajduje się mniej więcej w połowie wartości napięcia zasilania, co dla przedstawionego układu wyniesie 6 V, czyli więcej niż napięcie zasilania mikrokontrolera. Wskutek tego dowolny stan logiczny poprawny z punktu widzenia napięć zasilania na poziomie 5 V będzie przez wejście bramki rozpoznawany stale jako stan niski. Pułapką czyhającą na projektanta jest fakt, że taki sam układ, w którym zamiast 12 V zastosujemy zasilanie np. 9 V może działać - nie będzie to jednak działanie pewne i rozwiązanie to może



Rys. 13. Współpraca mikrokontrolera z układami zasilanymi napięciem o innej wartości



Rys. 14. Sterowanie wejść mikrokontrolera z wyjść układów zasilanych wyższym napięciem

przysporzyć sporych kłopotów podczas uruchamiania układu (okresy poprawnego i błędnego działania zależne od temperatury, poziomu zakłóceń itp.). Poprawiony układ przedstawiono na rys. 13b). Problem został rozwiązany przez zastosowanie układu inwertera z tranzystorem NPN łączącym obydwie obwody. Układ gwarantuje poprawną pracę, należy jedynie zwrócić uwagę, że neguje on podawany na bramkę sygnał, co należy uwzględnić przy pisaniu oprogramowania. Innym rozwiązaniem jest zastosowanie wzmacniacza operacyjnego (komparatora), na którego wejście odwracające podajemy sygnał z mikrokontrolera, a na wejście odwracające wartość napięcia progowego - w tym wypadku zgodnie ze standardem CMOS-5V jest to napięcie 2,5 V utworzone przez dzielnik R1 i R2 (rys. 13c).

Należy pamiętać, że wzmacniacz operacyjny musi być zasilany z tego samego źródła (o takim samym napięciu) co sterowana nim bramka. Zaletą tego układu jest fakt, że nie odwraca on fazy sygnału, należy jednak pamiętać o ograniczonym zakresie częstotliwościowym wzmacniacza operacyjnego (układy z pojedynczymi tranzystorami w tej konkurencji wygrywają - są dużo szybsze).

Podobnych zasad należy przestrzegać, gdy do wejścia mikrokontrolera dołączamy zasilany wyższym napięciem komparator lub inny układ z wyjściem komplementarnym (nie dotyczy układów z wyjściem typu otwarty kolektor/dren). Na rys. 14a przedstawiono błędny układ - połączenie bezpośrednie, w wyniku którego w najgorszym przypadku może dojść do uszkodzenia wejścia mikrokontrolera i/lub wyjścia układu sterującego. Na rys. 14b przedstawiono najprostszy sposób rozwiązania problemu - zastosowanie diody blokującej pojawienie się wysokiego napięcia na wejściu mikrokontrolera. Kiedy układ współpracujący z mikrokontrolerem jest w stanie wysokim, to dioda zostaje spolaryzowana zaporowo - nie przewodzi prądu. Na wejściu mikrokontrolera

utrzymuje się również stan wysoki wskutek wbudowanego (lub w razie konieczności dołączonego) rezystora podciągającego. Stan niski na wyjściu układu współpracującego powoduje polaryzację diody w kierunku przewodzenia i przepływ prądu - napięcie pojawiające się na wyprowadzeniu mikrokontrolera ma wartość napięcia wyjściowego układu współpracującego powiększonego o napięcie przewodzenia diody (warto tutaj zastosować diodę Schottky'ego ze względu na niż-

szy spadek napięcia). Problem można oczywiście rozwiązać w inny sposób, na przykład stosując omówiony wyżej układ z tranzystorem - układ z diodą jest jednak najprostszy.

Porty mikrokontrolera od strony programowej

Odpowiednie połączenie elektryczne wyprowadzeń portów to nie wszystko - konieczna jest jeszcze odpowiednia obsługa programowa. Przede wszystkim należy zwrócić uwagę na fakt, że porty mikrokontrolera są obsługiwane w sposób synchroniczny - tzn. odczyt danych z portu czy zapis do niego odbywa się zgodnie z częstotliwością cyklu maszynowych mikrokontrolera (pod warunkiem wykonywania rozkazów zapisu/odczytu portu). Dlatego mikrokontroler nie jest w stanie wykryć impulsów wejściowych krótszych niż 1 cykl maszynowy (w przypadku niektórych rozkazów są to 2 cykle) oraz sam nie jest w stanie generować przebiegów o stanach krótszych niż 1 cykl ($f_{zegara}/12$).

Dla programisty zapis czy odczyt portu nie różni się od zapisu czy odczytu wewnętrznej pamięci danych. Możliwy jest także zapis lub odczyt pojedynczych linii portów z wykorzystaniem adresowania bitowego. Najczęściej wykonywane instrukcje odwołujące się do portów to:

```
MOV P1,#0F0H
;jednoczesny zapis ośmiu linii
;portu (np. 4 starsze ustawione na
;"1", 4 młodsze wyzerowane.

MOV P1,A
;wysłanie do portu P1 zawartości
;akumulatora

MOV A,P1
;odczytanie zawartości linii portu
;P1 i wpisanie jej do akumulatora

SETB P1.1
;ustawienie "1" na linii P1.1

CLR P1.1
;wyzerowanie linii P1.1
```

JB P1.2,etykieta
;wykonanie skoku warunkowego, gdy
;na linii P1.2 panuje stan wysoki

Te instrukcje nie stanowią pełnego zbioru rozkazów odwołań do portów. Najogólniej można powiedzieć, że do obsługi portów można użyć wszystkich rozkazów mogących wykonywać operacje na adresowanej bezpośrednio komórce pamięci lub pojedynczym bicie. Specyficzne działanie w stosunku do portów posiada pewna liczba rozkazów określanych jako instrukcje typu odczyt-modyfikacja-zapis. Są to:

- ANL - iloczyn logiczny,
- ORL - suma logiczna,
- XRL - suma modulo 2,
- JBC - skok warunkowy i wyzerowanie bitu,
- CPL - negacja bitu,
- INC - inkrementacja,
- DEC - dekrementacja,
- DJNZ - dekrementacja bajtu i skok jeśli nie zero,
- MOV P_.,C - przesłanie zawartości wskaźnika C na dowolną linię portu,
- CLR P_ - zerowanie linii portu,
- SETB P_ - ustawianie linii portu.

Instrukcje te działają w następujący sposób: najpierw odczytywany jest wewnętrzny stan rejestrów odpowiedzialnych za stany na wyprowadzeniach portów, potem następuje modyfikacja odczytanej zawartości, a na końcu następuje zapis nowej wartości do portu. Należy zwrócić uwagę, że odczyt rejestru portu nie ma nic wspólnego ze stanami na liniach zewnętrznych - jest to wartość jaka została wpisana tam programowo, np. dla linii wykorzystywanych jako wejścia powyższe instrukcje będą odczytywały stan „1” niezależnie od panującego na linii napięcia. Będzie to spowodowane tym, że chcąc korzystać z linii jako wejście programista musiał wcześniej wpisać do odpowiedniego rejestru jedynkę i to właśnie ona będzie odczytywana. Szczególną ostrożność należy zachować w przypadku nieprzemyślanego zastosowania instrukcji JBC P_.,etykieta. Także tutaj skok i wyzerowanie podanego bitu (pojedynczej linii portu) nastąpi jeśli w rejestrze portu będzie na danej pozycji stan „1”, niezależnie od faktycznego stanu występującego na wyprowadzeniu mikrokontrolera (w odróżnieniu od instrukcji JB, JNB i CJNE - one odczytują stany linii zewnętrznych). Nieprzemyślane wykonanie instrukcji typu odczyt-modyfikacja-zapis często prowadzi do trudnych do zlokalizowania błędów programowych. Mimo tego nie należy się bać i rezygnować z tych instrukcji - są one bardzo przydatne, należy jednak stosować je w sposób świadomy i zdawać sobie sprawę ze sposobu ich działania.

Paweł Hadam