

Politechnika Poznańska
Wydział Elektroniki i Telekomunikacji

MIKROKONTROLER 8051

**Linie wejść/wyjść, porty oraz pamięć
wewnętrzna mikrokontrolera 8051**

Poznań 2009

Wstęp teoretyczny

Układ 8051 zawiera cztery 8-bitowe porty P0, P1, P2, P3. Wszystkie te porty mogą być adresowane jako całe bajty lub jako poszczególne ich bity. Do dyspozycji istnieją zatem 32 linie wejść/wyjść. Taka sytuacja występuje gdy program jest wpisany do wnętrza mikrokontrolera. Jeśli jest umieszczony w zewnętrznej pamięci EPROM (tak jak w systemie DSM-51) to do bezpośredniego sterowania pozostaje tylko port P1 i sześć linii portu P3.

W strukturze wewnętrznej mikrokontrolera porty umieszczone są w obszarze rejestrów specjalnych (SFR - Special Function Registers). Każdy rejestr posiada swój adres, który służy do jego identyfikacji. Rejestry specjalne zajmują wewnętrzny obszar pamięci RAM (256x8) od adresu 128 do 255, czyli od 80H do FFH. Rejestrom poszczególnych portów przyporządkowano następujące adresy P0 - 80H, P1 - 90H, P2 - A0H, P3 - B0H.

Poniżej w tabeli zostały zestawione wszystkie rejestry specjalne mikrokontrolera 8051.

adres	symbol	nazwa
0E0H	* ACC	akumulator
0F0H	* B	rejestr B
0D0H	* PSW	rejestr stanu
81H	SP	wskaźnik stosu
	DPTR	wskaźnik danych
82H	DPL	mniej znaczący bajt
83H	DPH	bardziej znaczący bajt
80H	* P0	port 0
90H	* P1	port 1
0A0H	* P2	port 2
0B0H	* P3	port 3
0B8H	* IP	rejestr priorytetu przerwań
0A8H	* IE	rejestr zezwoleń przerwań
89H	TMOD	rejestr trybu pracy timerów
88H	* TCON	rejestr sterujący timerów
8CH	TH0	Timer 0 bardziej znaczący bajt
8AH	TL1	Timer 0 mniej znaczący bajt
8DH	TH1	Timer 1 bardziej znaczący bajt
8BH	TL1	Timer 1 mniej znaczący bajt
98H	* SCON	rejestr sterujący portu szeregowego
99H	SBUF	bufor portu szeregowego
87H	PCON	rejestr sterujący zasilaniem mikrokontrolera

Po sygnale RESET wszystkie bity w portach mikrokontrolera 8051 są w stanie 1.

Wszystkie rejestry wewnętrzne mikrokontrolera 8051 są rejestrami 8-bitowymi.

W mikrokontrolerze 8051 umieszczono, oprócz rejestrów specjalnych 128 komórek pamięci RAM pod adresami 00 - 7FH. Początkowe 32 bajty pamięci stanowią 4 banki rejestrów.

Każdy bank zawiera 8 rejestrów: R0 - R7. Do tego obszaru można odwoływać się podając adres odpowiedniej komórki lub nazwę rejestru np.

MOV A,R0.

Odpowiedni bank rejestrów wybierany jest przez bity umieszczone w rejestrze stanu RS0 i RS1.

RS1	RS0	bank	adresy
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

Przy pomocy rejestrów R0 i R1 z dowolnego banku można adresować inne komórki pamięci, czyli realizować tzw. adresowanie pośrednie. Adresowanie pośrednie dotyczy tylko obszaru pamięci RAM (tj. 00H - 7FH), adresowanie bezpośrednie może być stosowane w obrębie całego obszaru pamięci.

W pamięci RAM za obszarem rejestrów (00 - 1FH) umieszczony jest obszar, do którego można odwoływać się również przez adresowanie poszczególnych bitów (20H - 2FH) podobnie jak do wybranych rejestrów specjalnych.

Przebieg ćwiczenia

1. Napisz program, który będzie powodował zaświecenie diody LED podłączonej do 7 linii w porcie P1 mikrokontrolera 8051.

Przed uruchomieniem napisany programu musi być poddany procesowi asemblacji. Do tego celu służy program DSM51ASS, (jako parametr należy podać nazwę pliku z kodem źródłowym). W wyniku przeprowadzonej asemblacji powstaje plik z listingiem programu i plik z kodem wynikowym w formacie HEX.

2. Zmodyfikuj program z polecenia 1.) tak by dioda była na przemian zapalana i gaszona. Wykorzystaj podprogram realizujący opóźnienie DELAY_100MS. Parametr zwiększający czas opóźnienia jest przekazywany do tego podprogramu przez rejestr A.
3. Przekształć przykład 2.) w ten sposób, by świecenie diody odbywało się na przemian z włączaniem brzęczyka podłączonego do linii piątej portu P1. Narysuj algorytm działania tego programu. Zastosuj rozkazy odwołujące się do poszczególnych bitów w porcie.
4. Napisz program realizujący te same funkcje co w punkcie 3.), ale wykorzystaj instrukcje operujące na całych bajtach, a nie na poszczególnych bitach.
5. Napisz program, który będzie wyświetlał na wyświetlaczu LCD numery kolejnych banków rejestrów (0 - 3), zapisane wcześniej w poszczególnych rejestrach tych banków np. R7. Do czyszczenia wyświetlacza LCD służy procedura LCD_CLR. Procedura WRITE_HEX powoduje wyświetlenie na wyświetlaczu LCD zawartości akumulatora.
6. Napisz program, który wykorzystując adresowanie pośrednie zeruje 10 kolejnych komórek pamięci, zaczynając od adresu 40H.
7. Napisz program zapisujący kolejne rejestry z banku 3 kolejnymi liczbami, np.: 0, 1, ..., 7. Skorzystaj z adresowania rejestrowego. Na wyświetlacz LCD wyślij wartość jednego z rejestrów wykorzystując adresowanie bezpośrednie.
8. Napisz program z zadania 7.) wykorzystując adresowanie pośrednie do zapisu danych do rejestrów.