

Ćwiczenie 30

Techniki mikroprocesorowe
Programowanie w języku Asembler
mikrokontrolerów rodziny '51

Cel ćwiczenia

Poznanie architektury oraz zasad programowania mikrokontrolerów rodziny '51, aby zapewnić podstawy do ich zastosowania w praktyce inżynierskiej.

Spis treści:

1. Architektura mikrokontrolerów rodziny 51'
 - 1.1. Charakterystyka zasobów poszczególnych generacji mikrokontrolerów rodziny '51
 - 1.2. Opis wyprowadzeń mikrokontrolera 8253
 - 1.3. Schemat blokowy mikrokontroler 8253
 - 1.4. Wewnętrzne układy licznikowo-zegarowe mikrokontrolera 8051
2. Organizacja pamięci mikrokontrolerów rodziny 51' na przykładzie układu 8253
 - 2.1. Obszary pamięci wewnętrznej
 - 2.2. Tryby adresowania pamięci przez mikrokontroler rodziny 51'
3. Rejestry specjalne (SFR)
 - 3.1. Wykaz rejestrów specjalnych mikrokontrolera 8253
 - 3.2. Opis rejestrów
4. Podstawowe instrukcje
 - 4.1. Instrukcja MOV
 - 4.2. Instrukcje operacji bitowych
 - 4.3. Instrukcje operacji arytmetycznych
 - 4.4. Instrukcje skoku warunkowego
 - 4.5. Instrukcje skoku bezwarunkowego
 - 4.6. Wywołanie procedur
 - 4.7. Pozostałe instrukcje adresowe
5. Obsługa portów we/wy
6. Obsługa przerwań
7. Przykładowa struktura programu w języku assembler
8. Wyświetlacz 7-segmentowy zestawu EASY 8051v6
9. Reset mikrokontrolera

1. Architektura mikrokontrolerów rodziny 51'

1.1. Charakterystyka zasobów poszczególnych generacji mikrokontrolerów rodziny '51

8051:

- ośmiobitowa jednostka centralna z wbudowanym procesorem operacji bitowych;
- 255 instrukcji, w tym rozkazy szybkiego mnożenia i dzielenia liczb 8-bitowych;
- Flash wewnętrzny 4kB (kiedyś ROM);
- SRAM wewnętrzny 128 bajtów;
- 4 porty / 32 linie wejścia/wyjścia;
- wbudowane dwa 16-bitowe układy licznikowo-zegarowe;
- wbudowanym dwukierunkowy port transmisji szeregowej;
- zasilanie od 4 V do 5,5 V;
- maksymalna częstotliwość zewnętrznego sygnału taktującego do 33 MHz.

8052:

- Flash wewnętrzny 8kB (kiedyś ROM), dodatkowe 4 kB względem 8051;
- SRAM wewnętrzny 256 bajtów, dodatkowe 128 bajtów względem 8051;
- wbudowane trzy 16-bitowe układy licznikowo-zegarowe, dodatkowy licznik względem 8051;
- zasilanie od 4 V do 5,5 V;
- maksymalna częstotliwość zewnętrznego sygnału taktującego do 33 MHz.

8053:

- Flash wewnętrzny 12kB, dodatkowe 4 kB względem 8052;
- szeregowy interfejs urządzeń peryferyjnych (SPI);
- zasilanie od 4 V do 6 V;
- maksymalna częstotliwość zewnętrznego sygnału taktującego do 24 MHz.

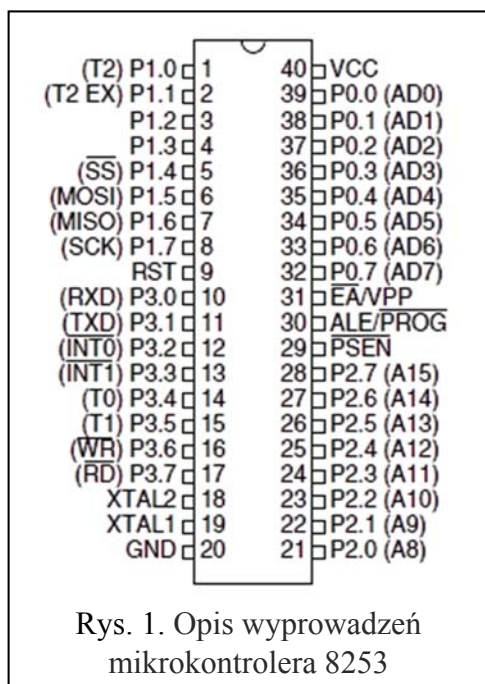
8252:

- Flash wewnętrzny 8kB;
- EEPROM wewnętrzny 2kB;
- zasilanie od 4 V do 6 V;
- maksymalna częstotliwość zewnętrznego sygnału taktującego do 24 MHz.

8253:

Zgodnie z opisem przedstawionym w dokumentacji mikrokontrolera AT89S8253 firmy ATMEL układ wyposażony jest m.in. w:

- ulotną wewnętrzną pamięć danych SRAM o pojemności 256 bajtów,
- nieulotną wewnętrzną pamięć danych EEPROM o pojemności 2kB,
- nieulotną wewnętrzną pamięć programu Flash o pojemności 12kB,
- cztery porty wejścia-wyjścia,
- system przerwań z układem priorytetów,
- wbudowane trzy 16-bitowe układy licznikowo-zegarowe,
- szeregowy interfejs urządzeń peryferyjnych,
- asynchroniczny port transmisji szeregowej,
- zasilanie od 2,7 V do 5,5 V;
- maksymalna częstotliwość zewnętrznego sygnału taktującego do 24 MHz.



1.2. Opis wyprowadzeń mikrokontrolera 8253

VCC - napięcie zasilania;

GND - masa zasilania;

XTAL1\2 - wejścia wewnętrznego układu oscylatora;

RST - wejście sygnału zerującego,

ALE/PROG - wyjście sygnału sterującego dostępem do zewnętrznej pamięci;

PSEN - wyjście impulsu odczytu z zewnętrznej pamięci;

EA\VPP - sygnał wejściowy wymuszający pobieranie rozkazów tylko z zewnętrznej pamięci programu;

P0.x- port 0 - ośmiobitowy, dwukierunkowy port z wyjściami typu otwarty dren, podczas dostępu do pamięci zewnętrznej służący do multipleksowanego przesyłania mniej znaczącego bajtu adresu i bajtu danych, w trakcie programowania i weryfikacji służy do wprowadzania i odczytu kodu instrukcji;

P1.x- port 1 - ośmiobitowy, dwukierunkowy port z wewnętrznymi rezystorami drenowymi podłączonymi do zasilania; w trakcie programowania i weryfikacji służy podaniu mniej znaczącego bajtu adresu; w przypadku 8253 bity zerowy i pierwszy mają dodatkowe funkcje:

P1.0 / T2 - zewnętrzne wejście impulsów licznika 2;

P1.1 / T2EX - zewnętrzne wejście wyzwalające licznik 2;

P1.4 / linia SS (ang. Slave Select) - synchroniczna transmisja szeregową, wybór aktywnego układu peryferyjnego;

P1.5 / MOSI - (ang. Master Output Slave Input) - synchroniczna transmisja szeregową, dane do układu peryferyjnego;

P1.6 / MISO - (ang. Master Input Slave Output) - synchroniczna transmisja szeregową, dane z układu peryferyjnego;

P1.7 / SCLK - (ang. Serial CLock) - synchroniczna transmisja szeregową, sygnał zegarowy;

P2.x - port 2 - ośmiobitowy, dwukierunkowy port z wewnętrznymi rezystorami drenowymi podłączonymi do zasilania, podczas dostępu do pamięci zewnętrznej służący do przesyłania bardziej znaczącego bajtu adresu; w trakcie programowania bardziej znaczącego bajtu adresu;

P3.x - port 3 - ośmiobitowy, dwukierunkowy port z wewnętrznymi rezystorami drenowymi podłączonymi do zasilania; alternatywny zestaw funkcji:

P3.0 / RxD - wejście asynchronicznego portu szeregowego;

P3.1 / TxD - wyjście asynchronicznego portu szeregowego;

P3.2 / INT0 - wejście przerwania zewnętrznego 0;

P3.3 / INT1 - wejście przerwania zewnętrznego 1;

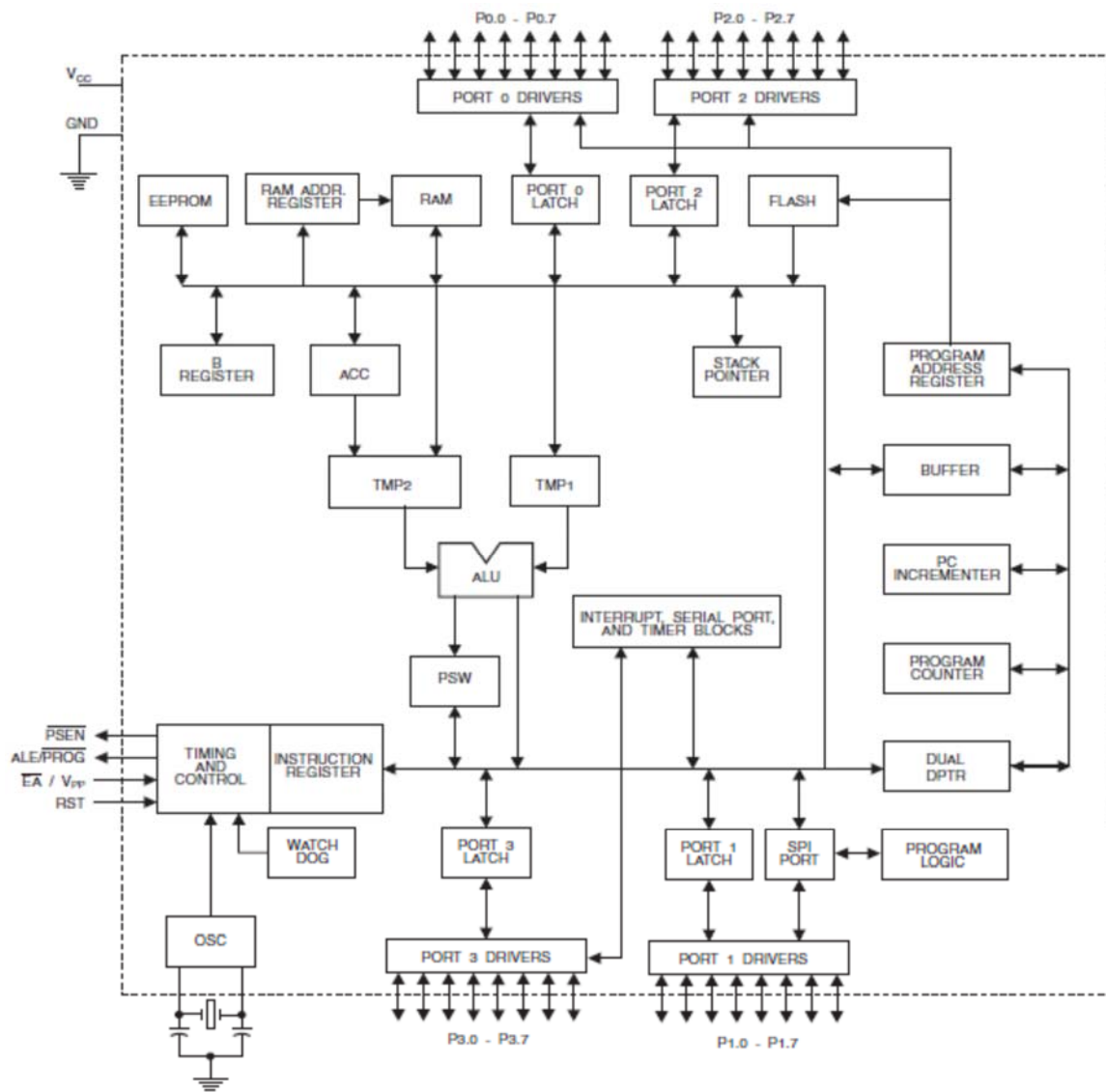
P3.4 / T0 - wejście impulsów licznika 0;

P3.5 / T1 - wejście impulsów licznika 1;

P3.6 / WR - sygnał zapisu do zewnętrznej pamięci lub innych układów;

P3.7 / RD - sygnał odczytu z zewnętrznej pamięci lub innych układów;

1.3. Schemat blokowy mikrokontroler 8253a (źródło dokumentacja techniczna AT89S8253)



Rys. 2. Schemat blokowy mikrokontrolera 8253 (źródło dokumentacja techniczna AT89S8253)

W układzie scalonym umieszczono m.in.:

- rdzeń mikroprocesora CPU z 8-bitową jednostką arytmetyczno–logiczną (ALU), której odczyt/zapis możliwy jest jedynie pośrednio przez akumulator (ACC);
- 4 uniwersalne dwukierunkowe porty wejścia/wyjścia, gdzie możliwy jest zarówno odczyt rejestru portu (PORT LATCH), jak i stanu poszczególnych linii portu (PORT DRIVERS);
- programowany szeregowy port transmisji dwukierunkowej, który może np. służyć do komunikowania się z komputerem wyposażonym w złącze RS232C (SERIAL PORT);
- szeregowy, synchroniczny interfejs urządzeń peryferyjnych (SPI);
- układ generowania przerwań systemowych (INTERRUPT);
- wbudowane trzy 16-bitowe układy licznikowo-zegarowe (TIMER BLOCK);
- układ wewnętrznego oscylatora (TIMING AND CONTROL), który ogranicza do minimum konieczność stosowania zewnętrznych elementów do pojedynczego rezonatora kwarcowego (wejście OSC) oraz dwóch dodatkowych kondensatorów ceramicznych;
- pamięć do przechowywania danych i wyników obliczeń: RAM oraz EEPROM;
- wewnętrzną pamięć typu FLASH do przechowywania programu.

1.4. Wewnętrzne układy licznikowo-zegarowe mikrokontrolera 8051

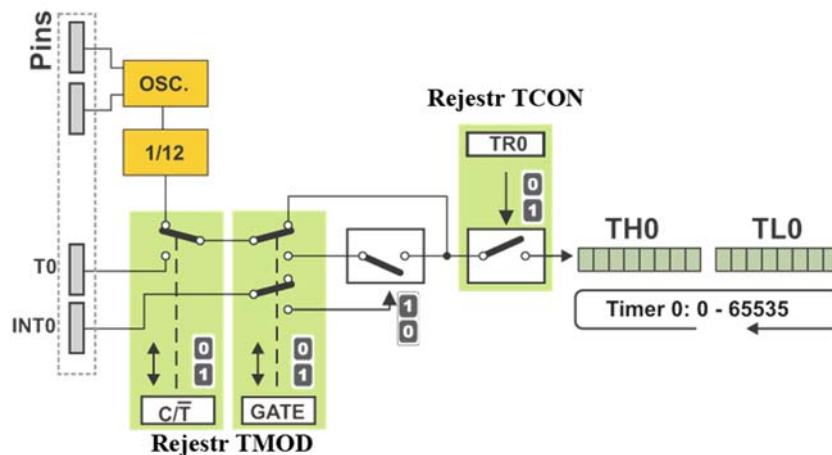
Podstawowe układy licznikowo-zegarowe T0 i T1 mogą pracować w jednym z czterech przedstawionych poniżej trybach pracy. W każdym z nich układ może pracować jako czasomierz taktowany wewnętrznym sygnałem zegarowym lub licznik zdarzeń zewnętrznych zwiększający swoją wartość przy każdorazowym wykryciu zbocza opadającego na wejściu Tx.

Tryb 0

W tym trybie licznik pracuje w konfiguracji 13-bitowej. Do rejestrów przechowujących wartość licznika (TH0 i TL0) można wpisać dowolną wartość pamiętając, że 3 najmłodsze bity słowa TL0 będą ignorowane. Licznik po uruchomieniu będzie zliczał od wartości wpisanej na początku (może to być także wartość 0) do wartości maksymalnej, czyli 8191, po czym się wyzeruje, zgłaszając przerwanie informujące program o tym fakcie.

Tryb 1

Tryb ten jest bardzo podobny do trybu 0, z tym że do zliczania wykorzystywane są wszystkie 16-bitów licznika - pojemność licznika w tym trybie wynosi 65535, po czym następuje przepełnienie, czyli wyzerowanie z ustawieniem znacznika zgłoszenia przerwania. Schemat blokowy licznika pracującego w tym trybie przedstawiono na rysunku 3.



Rys. 3. Schemat blokowy licznika T0/ T1 pracującego w trybie 16-bitowym

TCON i TMOD - rejestry sterujące pracą licznika,

T0 - wejście zliczanych impulsów, INT0 - wejście sterujące zliczaniem,

OSC. - źródło sygnału zegarowego, gdy licznik wykorzystywany do pomiaru czasu.

Tryb 2

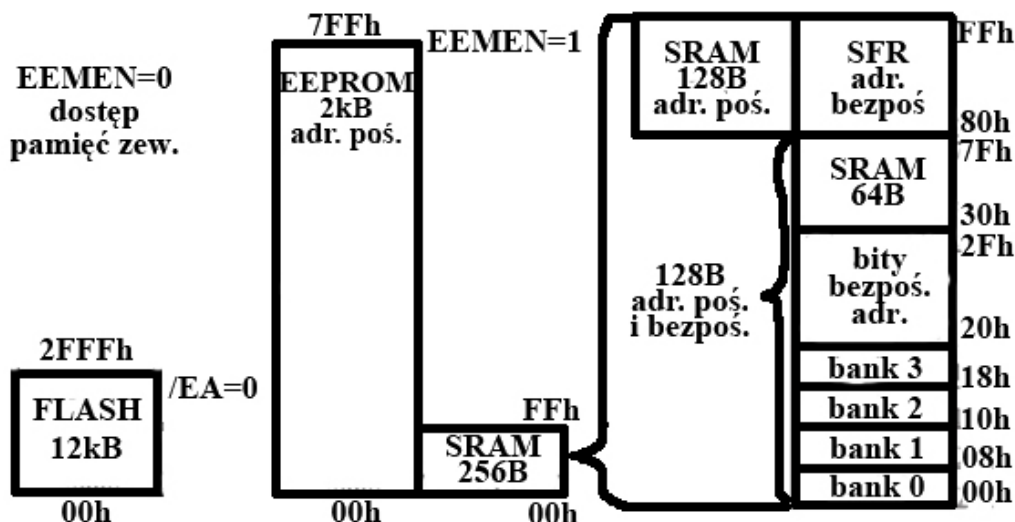
W trybie tym do zliczania wykorzystany jest tylko 8-bitowy rejestr TL0 licznika (TL1 dla licznika T1). Ośmiobitowy licznik TL0 zlicza w górę, aż do wartości maksymalnej czyli 255, po czym automatycznie zostaje przepisana do niego wartość początkowa z rejestru TH0.

Tryb 3

Licznik T1 jest zatrzymany i nie pracuje. Oba rejestry licznika T0: TH0 i TL0 są wykorzystywane jako dwa niezależne 8-bitowe liczniki, przy czym istnieją pewne ograniczenia co do ich funkcji, a mianowicie: TL0 może liczyć impulsy z wejścia T0 lub pracować jako czasomierz zliczając impulsy wewnętrzne, TH0 może zostać wykorzystany tylko jako czasomierz, czyli zliczać impulsy wewnętrzne.

2. Organizacja pamięci wewnętrznej mikrokontrolerów 51' na przykładzie układu 8253

Pamięć wewnętrzna układu 8253 składa się z: pamięci ulotnej SRAM (niepodtrzymywanej po odłączeniu zasilania) do przechowywania danych lub wyników obliczeń oraz nieulotnej EEPROM (zapis do 100 000 razy) do przechowywania danych oraz tylko do przechowywania kodu programu pamięci FLASH (zapis do 10 000 razy - wymaga zatrzymania wykonywania programu).



Rys. 4. Organizacja pamięci wewnętrznej mikrokontrolerów 51' na przykładzie układu 8253

2.1. Obszary pamięci wewnętrznej

- pamięć programu Flash 12 kB (do 64kB), odczyt /EA=0 dostępna przez rozkaz MOVC.
- wewnętrzna pamięć danych EEPROM 2 kB (do 64 kB) dostępna przez rozkaz MOVX, gdy bit EEMEN rejestru EECON jest ustawiony na 1.
- zewnętrzna pamięć danych dostępna przez rozkaz MOVX, gdy bit EEMEN rejestru EECON jest ustawiony na 0.
- wewnętrzna pamięć danych SRAM 256B, 128B adresowane bezpośrednio i pośrednio oraz 128B adresowane tylko pośrednio, dostęp przez rozkaz MOV.
- adresowane bezpośrednio rejestry specjalne (SFR - Special Function Registers).
- każdy bank od 0 do 3 zawiera 8 rejestrów pomocniczych od R0 do R7, wybór banku roboczego dokonywany jest przez ustawienie odpowiednich bitów rejestru stanu (PSW). Wśród rejestrów pomocniczych dwa pełnią szczególną rolę – rejestry R0 i R1 służą do adresowania pośredniego.
- Obszar pamięci od 20h do 2Fh - bity bezpośrednio adresowalne, do których można się odwoływać oraz wykonywać operacje bitowe: bity o adresach 0 ... 127 (00H ... 7FH). Bit o adresie 00H to najmniej znaczący (zerowy) bit komórki 20H, co można zapisać jako 20H.0. Bit o adresie 01H to pierwszy bit komórki 20H, czyli 20H.1. Bit o adresie 08H to z kolei zerowy bit komórki 21H, czyli 21H.0; bit 30H to bit 0 komórki 26H, itd. Ostatni bit z tego zakresu, czyli bit o adresie 127 (7FH), to najbardziej znaczący (siódmy) bit komórki 2FH, czyli bit 2FH.7. Każda komórka z tego zakresu zawiera 8 bitów; komórek jest 16, więc razem 128 bitów np. setb 30H (adres bitu), setb 26H.0 (adres bajta oraz pozycja bitu w bajcie). Takie bity mogą służyć jako znaczniki.

2.2. Tryby adresowania pamięci przez mikrokontroler rodziny 51'

- rejestrowy,
 - bezpośredni,
 - natychmiastowy,
 - pośredni, zawartością rejestru,
 - pośredni, sumą zawartości rejestru bazowego i indeksowego.
- W adresowaniu rejestrowym, operandem (argumentem instrukcji) jest jeden z rejestrów R0...R7 (oczywiście z aktywnego banku rejestrów), akumulator ACC, rejestr B, wskaźnik danych DPTR lub wskaźnik przeniesienia CY (traktowany umownie jako akumulator dla operacji na pojedynczych bitach): **MOV A, Rn**
 - W adresowaniu bezpośrednim operand jest jawnie podanym, adresem rejestru SFR, bajtu dolnego obszaru wewnętrznej pamięci RAM lub bitu adresowalnego bezpośrednio: **MOV A, adres**
 - W adresowaniu natychmiastowym operand jest stałą umieszczoną w pamięci programu: **MOV A, #dana**
 - Adresowanie pośrednie zawartością rejestru polega na określaniu adresu za pomocą zawartości rejestrów R0 lub R1 (znajdujących się w aktywnym banku rejestrów). Rejestry te są 8-bitowe, tak więc adresowanie pośrednie wykorzystuje zawartość rejestru jako wskaźnik bajtu znajdującego się w wewnętrznej pamięci RAM lub 256-bajtowym bloku danych pamięci zewnętrznej. Jediną formą adresowania pośredniego wykorzystującą wskaźnik 16-bitowy, umożliwiającą adresowanie zewnętrznej pamięci danych w zakresie pełnych 64 kB, są operacje z wykorzystaniem wskaźnika DPTR: **MOV A, @Ri**
 - Adresowanie pośrednie sumą zawartości rejestru bazowego i indeksowego umożliwia odczytanie bajtu pamięci programu, którego adres jest określony tą sumą: **MOVC A, @A + DPTR**

3. Rejestry specjalne (SFR)

3.1. Wykaz rejestrów specjalnych mikrokontrolera 8253 (22 główne rejestry z 8051)

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW					SPCR		D7
C8	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2		CF
C0								C7
B8	IP	SADEN						BF
B0	P3						IPH	B7
A8	IE	SADDR	SPSR					AF
A0	P2						WDTRST	WDTCON
98	SCON	SBUF						9F
90	P1						EECON	97
88	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	CLKREG
80	P0	SP	DP0L	DP0H	DP1L	DP1H	SPDR	PCON
								87

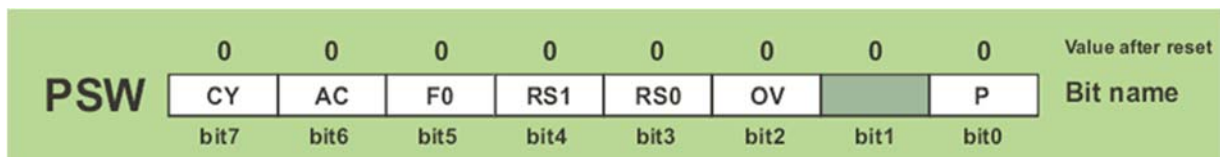
Rys. 5. Wykaz rejestrów specjalnych mikrokontrolera 8253

ACC - akumulator: 0E0H
 B - rejestr B: 0F0H
 PSW - rejestr stanu: 0D0H
 SP - wskaźnik stosu: 81H
 DPTR - dwubajtowy wskaźnik danych
 DPL - młodszy bajt: 82H
 DPH - starszy bajt: 83H
 P0 - Port 0: 80H
 P1 - Port 1: 90H
 P2 - Port 2: 0A0H
 P3 - Port 3: 0B0H
 IP - rejestr kontroli priorytetów przerwań: 0B8H
 IE - rejestr zezwoleń na przerywania: 0A8H
 TMOD - rejestr sterujący pracą liczników T0 i T1: 89H
 TCON - rejestr kontrolny liczników T0 i T1: 88H
 T2MOD - rejestr sterujący pracą licznika T2: C8
 T2CON - rejestr kontrolny licznika T2: C9
 TH0 - timer 0 starszy bajt: 8CH
 TL0 - timer 0 młodszy bajt: 8AH
 TH1 - timer 1 starszy bajt: 8DH
 TL1 - timer 1 młodszy bajt: 8BH
 SCON - sterownie transmisją szeregową: 98H
 SBUF - bufory transmisji szeregowej: 99H
 PCON - sterowanie trybami 'power down': 87H
 EECON - steruje dostępem do pamięci EEPROM: 96H

Podczas pisania programu można używać bezpośrednio adresów komórek z pamięci SFR zamiast ich symboli, np. zamiast pisać `mov A, P1`, można napisać `mov A, 90H`, zamiast pisać `mov P3, B`, można napisać `mov 0B0H, B` lub nawet `mov 0B0H, 0F0H` (0F0H to adres rejestru B), lecz jednak o wiele wygodniej jest używać nazw rejestrów, gdyż są one o wiele łatwiejsze do zapamiętania, niż ich adresy, a także program jest wtedy bardziej czytelny.

3.2. Opis wybranych rejestrów

PSW - rejestr stanu



PSW.7 - CY = znacznik przeniesienia 8 bit

Znacznik przeniesienia CY ustawiany jest w wyniku wykonania operacji arytmetycznych dodawania (ADD) lub odejmowania (SUBB), sygnalizując wystąpienie przeniesienia z 8-ego bitu wyniku (dodawanie) lub pożyczki z 9-tego bitu wyniku (odejmowanie).

PSW.6 - AC = znacznik przeniesienia pomocniczego do kodu BCD

PSW.5 - F0 = znacznik F0 użytkownika

PSW.2 - OV = znacznik przepełnienia

Znacznik nadmiaru OV ustawiany jest w przypadku, gdy w wyniku wykonanej operacji arytmetycznej wystąpiło przeniesienie na pozycję najstarszego bitu, który w przypadku liczb ze znakiem jest bitem znaku. Służy zatem do detekcji błędu określenia znaku wyniku w operacjach arytmetycznych na liczbach ze znakiem, np. zapisanych w kodzie (U2).

PSW.4 - RS1 = numer aktywnego banku rejestrów roboczych bit 1

PSW.3 - RS0 = numer aktywnego banku rejestrów roboczych bit 0

0 0 bank 0 adresy 00h-07h

0 1 bank 1 adresy 08h-0Fh

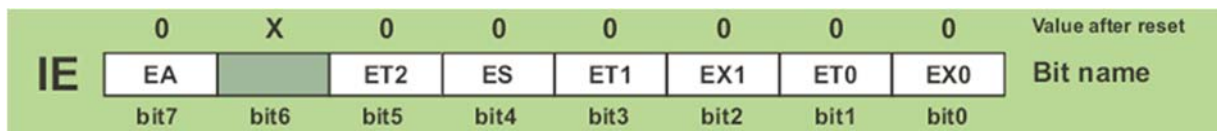
1 0 bank 2 adresy 10h-17h

1 1 bank 3 adresy 18h-1Fh

PSW.0 - P = bit parzystości akumulatora

Bit parzystości P odzwierciedla informację o ilości jedynek w akumulatorze.

IE rejestr zezwoleń na przerwania (0-przerwanie zablokowane, 1-przerwanie odblokowane)



EA - globalne zezwolenie na przerwania

ET2 - zezwolenie na przerwanie od T2

ES - zezwolenie na przerwanie od portu szeregowego (UART and SPI)

ET1 - zezwolenie na przerwanie od T1

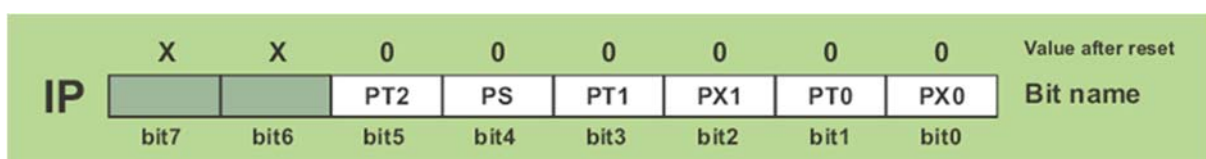
EX1 - zezwolenie na przerwanie zewnętrzne INT0

ET0 - zezwolenie na przerwanie od T0

EX0 - zezwolenie na przerwanie zewnętrzne INT1

mov IE, #10000010b ; globalne zezwolenie na przerwania oraz na przerwanie od T0

P - rejestr priorytetów przerwań (0-niższy priorytet, 1-wyższy priorytet)



PT2 - bit priorytetu przerwania od T2

PS - bit priorytetu przerwania od portu szeregowego

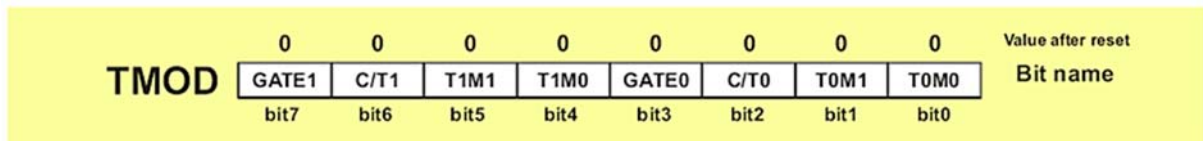
PT1 - bit priorytetu przerwania od T1

PX1 - bit priorytetu przerwania od INT1

PT0 - bit priorytetu przerwania od T0

PX0 - bit priorytetu przerwania od INT0

TMOD - rejestr sterujący pracą układów czasowo-licznikowych T0 i T1
 młodsze 4 bity odpowiadają licznikowi 0, starsze – licznikowi 1



GATE =1 - sterowanie zliczaniem przez wejście zewnętrzne INT (zliczani impulsów tylko, gdy INT = 1), zliczanie impulsów bez dodatkowego bramkowania, kiedy GATE=0
 C/T – określa rodzaj pracy układu czasowo-licznikowego (0 – odmierzanie czasu, 1 – licznik impulsów zewnętrznych)

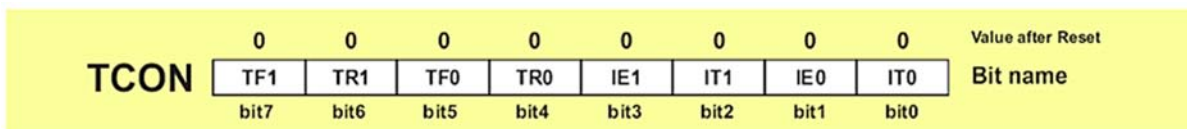
bity M0, M1 – ustalają trybu układu czasowo-licznikowego

TxM1 TxM0 Tryb

0	0	0	licznik 13-bitowy
0	1	1	licznik 16-bitowy
1	0	2	licznik 8-bitowy z autoprzeładowaniem
1	1	3	T0 i T1 jako osobne liczniki 8-bitowe

```
mov TMOD, #00010001b ; T0 i T1 liczniki 16-bitowe impulsów wewnętrznych.
mov TMOD, #00000001b ; T0 licznik 16-bitowy impulsów wewnętrznych lub setb T0M0.
```

TCON - rejestr kontrolny układów czasowo-licznikowych T0 i T1



TF0, TF1 – flaga przepelnienia licznika
 TR0, TR1 – flaga pracy licznika (0 – stop, 1 – zliczanie)

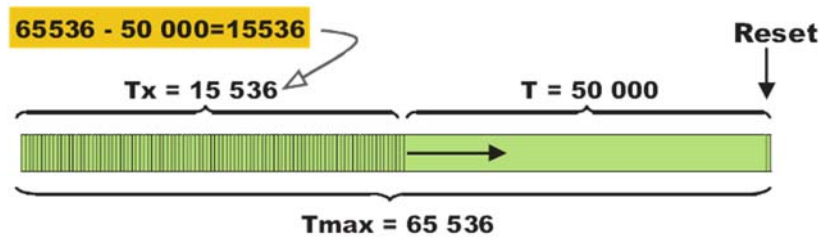
```
setb TR0 ; start licznika T0
```

Liczniki generują sygnał TFX przy przekroczeniu maksymalnej wartości – odpowiednio 2¹³, 2¹⁶, 2⁸. Jako, że są to liczniki zliczające "w górę", oraz ich inkrementacja następuje co jeden cykl zegarowy w celu kontrolowania odmierzanego przez nie czasu Tx jako wartość początkowa do TH | TL należy wpisać wartość:

$$X = \text{max zakres} - \frac{T_x}{12 * T_{osc}}$$

gdzie: max zakres w trybie 16 bit = 65536 impulsów, 8-bitowym = 256 impulsów,
 T_{osc} = 1/częstotliwość oscylatora.

```
Jeżeli licznik 16-bitowy ma zliczyć 50 000 impulsów to należy ustawić stan początkowy
licznika na: 65536-50000=15536 (3CB0h),
ponieważ zliczanie impulsów przez układy liczników T0 i T1 zawsze następuje "w górę":
mov TL0, #03Ch ; ustawianie wartości początkowej licznika T0 - 8 bitów rejestru TL0
mov TH0, #0B0h ; ustawianie wartości początkowej licznika T0 - 8 bitów rejestru TH0.
```



Jeżeli czas, który ma zostać odmierzony przez licznik przekracza dostępny zakres dla 16-bitów licznika należy zastosować jeden z rejestrów do zliczania kolejnych przepełnień licznika w celu odmierzenia wielokrotności czasu zliczania impulsów przez licznik.

T2CON- rejestr kontrolny licznika T2 (nie trzeba ustawić rejestru T2MOD)

Licznik T2 może pracować w jednym z trzech trybów pracy liczników T0, T1 (bez trybu 0 - 13 bit). W każdym z nich, w zależności od ustawienia bitu C/T2, układ może pracować jako czasomierz taktowany wewnętrznym sygnałem zegarowym (C/T2=0) lub licznik zdarzeń zewnętrznych. W każdym z wymienionych przypadków taktowania licznika T2 pracuje on jako licznik 16-bitowy. W trybie zapamiętania rejestrów licznika (ang. *capture mode*) każdorazowe przepełnienie licznika T2 powoduje ustawienie bitu TF2. **W trybie automatycznego przeładowania rejestrów licznika jeśli bit EXEN2=0, to przy przepełnieniu licznika T2 ustawiany jest bit TF2, a jednocześnie następuje przeładowanie rejestrów TL2 i TH2 licznika odpowiednio wartościami z rejestrów RCAP2L i RCAP2H, które wcześniej mogą być zapisane programowo.**

								Value after reset	
0 0 0 0 0 0 0 0									
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	Bit name
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	

TF2 - Znacznik przepełnienia licznika T2. Ustawiany sprzętowo przy przepełnieniu licznika.

Zerowanie musi być realizowane programowo.

EXF2 - Znacznik ustawiany sprzętowo i określający wystąpienie przeładowania nową wartością lub zapamiętania rejestrów roboczych TL2 i TH2 licznika T2 (w zależności od stanu bitu CP/RL2) spowodowanego wystąpieniem zbocza opadającego na wejściu T2EX przy ustawionym bicie EXEN2. W przypadku aktywnego przzerwania od licznika T2 powoduje przejście do wektora przzerwania związanego z tym licznikiem. Znacznik ten musi być zerowany programowo.

RCLK - Bit sterujący taktowaniem odbiornika portu szeregowego (dotyczy tylko trybu 1 i 3 pracy portu szeregowego): 0 - odbiornik portu szeregowego taktowany sygnałem przepełnienia licznika T1, 1 - odbiornik portu taktowany przepełnieniem licznika T2.

TCLK - Bit sterujący taktowaniem nadajnika portu szeregowego (dotyczy tylko trybu 1 i 3 pracy portu szeregowego): 0 - nadajnik portu szeregowego taktowany sygnałem przepełnienia licznika T1, 1 - nadajnik portu szeregowego taktowany przepełnieniem licznika T2.

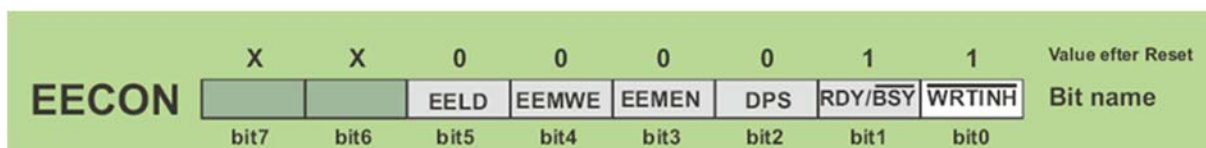
EXEN2 - Ustawienie tego bitu powoduje, że przy każdorazowym wystąpieniu zbocza opadającego na wejściu T2EX następuje przeładowanie rejestrów roboczych TH2 i TL2 licznika nową wartością odpowiednio z rejestrów RCAP2H i RCAP2L lub zapamiętanie wartości z rejestrów roboczych w rejestrach RCAP2H i RCAP2L (w zależności od stanu bitu CP/RL2) pod warunkiem, że licznik T2 nie jest używany do generowania prędkości transmisji portu szeregowego. Zerowy stan tego bitu - ignorowanie zdarzeń na wejściu T2EX.

TR2 - Bit sterujący pracą licznika T2: 0 - licznik zatrzymany, 1 - licznik pracuje.

C/T2 - wybór funkcji dla licznika T2: 0 - czasomierz taktowany wewnętrznym sygnałem zegarowym, 1 - licznik zdarzeń zewnętrznych.

CP/RL2 - Określa jeden z dwóch trybów pracy licznika **T2**: **0** - w przypadku wystąpienia przepełnienia licznika **T2** lub zbocza opadającego na wejściu **T2EX** (jeśli również **EXEN2=1**) rejestry robocze **TL2** i **TH2** są przeladowywane odpowiednio zawartością rejestrów **RCAP2L** i **RCAP2H**, **1** - zawartość rejestrów **TL2** i **TH2** jest przepisywana odpowiednio do rejestrów **RCAP2L** i **RCAP2H** przy każdorazowym wystąpieniu zbocza opadającego na wejściu **T2EX** (jeśli **EXEN2=1**). Kiedy co najmniej jeden z bitów **RCLK** lub **TCLK** jest ustawiony, stan tego bitu jest ignorowany, a licznik **T2** jest automatycznie przeladowywany nową wartością pobieraną z rejestrów **RCAP2L** i **RCAP2H** przy każdorazowym wystąpieniu jego przepełnienia.

EECON - rejestr sterujący dostępem do pamięci EEPROM



WRTINH - tylko do odczytu informuje o zbyt niskim napięciu do programowania pamięci,

RDY/BSY - tylko do odczytu informuje o trwającym zapisie do pamięci,

DPS: **0** - adres odczytu/zapisu znajduje się w rejestrach **DP0H** i **DP0L**, **1** - adres odczytu/zapisu znajduje się w rejestrach **DP1H** i **DP1L**,

EEMEN: **0** - instrukcja **MOVX** jest wykorzystywana do dostępu do pamięci zewnętrznej, **1** - instrukcja **MOVX** jest wykorzystywana do dostępu do pamięci EEPROM, jeśli adres wskazuje na przestrzeń ponad **2K**, mikrokontroler automatycznie wykona odczyt/zapis do pamięci zewnętrznej

EEMWE - zezwala na zapis do EEPROM instrukcją **MOVX**. Po zakończeniu zapisu musi być zerowany programowo, aby umożliwić odczyt z pamięci

EELD - aktywny zapis blokowy do pamięci po 32 bajty.

4. Podstawowe instrukcje

4.1. Instrukcja MOV

służy do przekazywania danych pomiędzy rejestrami procesora i pamięcią wewnętrzną.

MOV A, #dana – instrukcja załadowania 8-bitowej liczby „dana” do akumulatora

MOV A, #100 - załadowanie do akumulatora liczby 100

MOV A, Rn – do akumulatora zostaje załadowana zawartość rejestru **R0...7**, cykle: 1 bajty: 1
MOV R7, #0

MOV A, R7 - wyzerowanie akumulatora

MOV A, adres – do akumulatora zostaje załadowana zawartość komórki wewnętrznej pamięci RAM o adresie: „adres”, cykle: 1 bajty: 2

MOV A, 20h - załadowanie do A zawartości komórki o adresie 20h

MOV A, @Ri – do akumulatora zostaje załadowana zawartość komórki wewn. RAM, której adres znajduje się w rejestrze **R0** (**i=0**), lub **R1** (**i=1**), cykle: 1 bajty: 1

;przed wywołaniem instrukcji **R1** zawiera wartość 34h

MOV A, @R1 - załadowanie zawartości komórki o adresie 34h do akumulatora

MOV Rn, A – do rejestru Rn (R0...R7) zostaje załadowana zawartość akumulatora

MOV R3, A - przepisanie zawartości akumulatora do rejestru R3

MOV Rn, adres – do rejestru Rn (R0...R7) zostaje załadowana zawartość komórki o adresie „adres”, cykle: 1 bajty: 2

MOV Rn, #dana – do rejestru Rn (R0...R7) zostaje wpisana 8bitowa liczba, cykle: 1 bajty: 2

MOV R6, #10h - 10h szesnastkowo to 16 dziesiętnie

MOV adres, A – do komórki o adresie „adres” zostaje wpisana zawartość akumulatora, cykle: 1 bajty: 2 (kod instrukcji + adres)

MOV 00h, A - wpisanie zawartości akumulatora do komórki o adresie 0

MOV adres, Rn – do komórki o adresie „adres” zostaje wpisana zawartość rejestru Rn

MOV 30h, R5 ;wpisanie do komórki w wew. RAM zawartości rejestru ;R5.

MOV adres1, adres2 – przepisanie komórki o adresie „adres2” do komórki o adresie „adres1”

MOV 7Fh, 7Eh - przepisanie zawartości dwóch sąsiadujących komórek w wew. RAM

MOV adres, @Ri – do komórki o adresie „adres” zostaje wpisana zawartość komórki której adres znajduje się w rejestrze R0 (i=0) lub R1 (i=1)

MOV R1, #32

MOV 32h, @R1 - przepisanie zawartości komórki o adresie 32h na nią samą (fizycznie operacja ta nie ma efektu)

MOV adres, #dana – do komórki o adresie „adres” zostaje wpisana wartość stałą (8-bitowa liczba), cykle: 2 bajty: 3 (kod instrukcji + adres + dana)

MOV 45h, #100;komórce o adresie 45h zostaje nadana wartość 100

MOV @Ri, A – do komórki o adresie znajdującym się w rejestrze R0 (i=0) lub R1 (i=1)

zostaje wpisana zawartość akumulatora, cykle: 1 bajty: 1

CLR A - wyzerowanie akumulatora

MOV @R1, A ;wyzerowanie komórki o adresie w R1

MOV @Ri, adres – do komórki o adresie znajdującym się w rejestrze R0 (i=0) lub R1 (i=1) zostaje wpisana zawartość komórki o adresie „adres”, cykle: 2 bajty: 2 (kod instrukcji +adres)

MOV R0, #11h

MOV @R0,10h - przepisanie zawartości komórki o adresie 10h do komórki sąsiedniej o adresie 11h

MOV @Ri, #dana – do komórki o adresie znajdującym się w rejestrze R0 (i=0) lub R1 (i=1) zostaje wpisana wartość stała (liczba), cykle: 1 bajty: 2 (kod instrukcji + dana)

MOV @R0, #255 - jeżeli wcześniej rejestr R0 miał wartość np. 30h, to w ;efekcie tej operacji do komórki o adresie 30h zostanie wpisana liczba 255

MOV DPTR, #dana16 – instrukcja załadowania 16-bitowego, bezwzględnego adresu do wskaźnika danych DPTR, cykle: 2 bajty: 3 (kod instrukcji + starszy bajt + młodszy bajt liczby „dana16”) MOV DPTR,#0 ;wyzerowanie wskaźnika danych

NOP - opóźnienie procesora o jeden cykl maszynowy

4.2. Instrukcje operacji bitowych

CLR bit; - wyzeruj bit

CLR AFh - zablokowanie systemu przerwań za pomocą adresowania bitowego (EA = 0 w rejestrze IE) inaczej CLR EA

SETB bit; - ustaw bit, cykle: 1 bajty: 2 (instrukcja + adres bitu)

SETB P1.2 ;ustawienie wysokiego poziomu logicznego na wyprowadzeniu 2 portu P1 (pin 3)

CPL bit - zaneguj bit

CPL P1.0 - zmienia stan logiczny na wyprowadzeniu 0 portu P1 (pin 1) na przeciwny

ANL (iloczyn logiczny, np. ANL P2, A),

ORL (suma logiczna, np. ORL P1, #10),

XRL (suma logiczna bitów modulo 2, np. XRL P3, A),

4.3. Instrukcje operacji arytmetycznych

ADD – do wartości przechowywanej w akumulatorze dodawany jest wskazany argument, a wynik zostaje wpisany do akumulatora, znaczniki: C, AC i OV

ADD A , Rn - do akumulatora dodawana jest zawartość rejestru Rn: **ADD A , R2**

ADD A , adres - do akumulatora dodawana jest zawartość komórki w wewn. RAM o adresie „adres”: **ADD A , 2Fh** (dodanie do A zawartości komórki o adresie 2Fh)

ADD A , @Ri - do akumulatora dodawana jest zawartość komórki w wewn. RAM o adresie wskazywanym przez rejestr Ri (R0 lub R1):

ADD A , @R0 (dodanie do A zawartości komórki o adresie w R0)

ADD A , #dana - do akumulatora dodawany jest argument stały (8-bitowa liczba):

ADD A , #120 (dodanie do A liczby 120)

SUBB - od wartości przechowywanej w akumulatorze odejmowany jest wskazany argument oraz zawartość znacznika przeniesienia C, a wynik zostaje wpisany do akumulatora, znaczniki: C, AC i OV

SUBB A , Rn - od akumulatora odejmowana jest zawartość rejestru Rn oraz C: **SUBB A , R6**

SUBB A , adres - od akumulatora odejmowana jest zawartość komórki w wewn. RAM o adresie „adres” oraz znacznik C:

SUBB A , 45h (odjęcie od A zawartości komórki o adresie 45h i znacznika C)

SUBB A , @Ri - od akumulatora odejmowana jest zawartość komórki w wewn. RAM o adresie wskazywanym przez rejestr Ri (R0 lub R1) oraz C:

SUBB A , @R1 (odjęcie od A zawartości komórki o adresie w R1 oraz C)

SUBB A , #dana - od akumulatora odejmowany jest argument stały (8-bitowa liczba) oraz C:

SUBB A , #86h (odjęcie od A liczby 86h i znacznika C)

INC - do wskazanego argumentu jest dodawana jedynka

INC A - do akumulatora dodawana jest jedynka

INC Rn - do zawartości rejestru Rn dodawana jest jedynka

INC adres - do zawartości komórki o adresie „adres” dodawana jest jedynka: **INC 12h** (inkrementacja zawartości komórki o adresie 12h)

INC @Ri - do zawartości komórki o adresie wskazywanym przez Ri dodawana jest jedynka

INC DPTR - do 16-bitowego wskaźnika danych DPTR złożonego z rejestrów SFR: DPH (bardziej znaczący bajt) i DPL (mniej znaczący bajt) dodawana jest jedynka.

DEC - od wskazanego argumentu odejmowana jest jedynka (analogicznie jak INC)

MUL AB - 8-bitowa liczba bez znaku znajdująca się w akumulatorze jest mnożona przez 8-bitową liczbę bez znaku z rejestru B. 16-bitowy wynik wpisywany jest do rejestrów B i A (bardziej znaczący bajt do B, mniej znaczący bajt do A), znaczniki: jeśli wynik mnożenia jest > 255 to ustawiany jest znacznik OV, w przeciwnym razie OV jest zerowany. znacznik C jest zerowany, cykle: 4 bajty: 1

DIV AB - 8-bitowa liczba bez znaku, znajdująca się w akumulatorze jest dzielona przez 8-bitową liczbę z rejestru B. Część całkowita ilorazu wpisywana jest do akumulatora, a reszta z dzielenia do rejestru B. W przypadku gdy dzielnik jest równy 0 (B=0) to po wykonaniu operacji zawartość akumulatora i rejestru B jest nieokreślona oraz dodatkowo ustawiony zostaje znacznik OV, c) znaczniki: C = 0, OV =0 (zerowane), cykle: 4 bajty: 1

4.4. Instrukcje skoku warunkowego

JZ rel - jeżeli A (zawartość akumulatora) jest równe zero zmienia licznik programu (czyli adres następnej instrukcji) o wartość równą rel (8 bitową liczbą U2) lub skok do etykiety w przestrzeni SRAM 255 bajtów

JNZ rel - jeżeli A jest równe różne od zera zmienia licznik programu (czyli adres następnej instrukcji) o wartość równą rel (8 bitową liczbą U2) lub skok do etykiety w przestrzeni SRAM 255 bajtów

DJNZ - zawartość rejestru Rn jest zmniejszana o 1, a następnie jeżeli wynik jest różny od zera licznik programu (PC) jest modyfikowany – zostaje do niego dodana wartość rel (czyli zostaje wykonany skok).

JC rel - skok jest wykonywany wtedy, gdy zawartość flagi przeniesienia C jest równa 1

JNC rel - skok jest wykonywany wtedy, gdy zawartość flagi przeniesienia C jest równa 0.

JB b, rel - skok jest wykonywany wtedy, gdy bezpośrednio adresowany bit "b" jest równy 1.

JNB b, rel - skok jest wykonywany wtedy, gdy bezpośrednio adresowany bit "b" jest 0.

JBC b, rel - skok jest wykonywany wtedy, gdy bezpośrednio adresowany bit "b" jest równy 1. Bit jest następnie ustawiany na 0 (niezależnie czy skok jest wykonywany)

CJNE - porównują dwie wielkości i wykonuje skok, gdy nie są one równe (oraz ustawiają odpowiednio wskaźnik C)

CJNE A,#data,rel ;rejestr A ze stałą

CJNE A,direct,rel ;rejestr A z zawartością komórki pamięci

CJNE Rn,#data,rel ;rejestr Rn ze stałą

CJNE @Ri,#data,rel ;zawartość komórki pamięci adresowanej pośrednio ze stałą

4.5. Instrukcje skoku bezwarunkowego

SJMP rel - krótki skok względny o wartość równą rel (8 bitową liczbą U2) lub skok do etykiety w przestrzeni SRAM 255 bajtów:

AJMP - skok absolutny w przestrzeni do 2 kB np. EEPROM, z użyciem adresu 11 bitowego. Niższe 11 bitów licznika programu jest zamieniane przez 11 bitów określonych w instrukcji.

LJMP - skok absolutny w przestrzeni do 64 kB np. Flash 12 kB, z użyciem adresu 16 bitowego. 16 bitów licznika programu jest zamieniane przez 16 bitów z instrukcji.

Użycie **JMP** assembler wybiera właściwą instrukcję skoku bezwarunkowego.
JZ elseif ;jeżeli zawartość akumulatora jest równe zero skok do "elseif"
ADD A,R1
MOV R2,#0
SJMP endif ;zawsze skok do "endif " / opcjonalnie można użyć **JMP**
 elseif: **MOV** R2,#13
 endif: **MOV** A,R2
ADD A,R3

4.6. Wywołanie procedur

LCALL wywołanie procedury znajdującej się w przestrzeni pamięci o rozmiarach do 64 kB wymagającej adresowanie 16-bitowego.

ACALL wywołanie procedury znajdującej się w przestrzeni pamięci o rozmiarach do 2 kB wymagającej adresowanie 11-bitowego.

Użycie **CALL** assembler wybiera właściwą instrukcję wywołania procedury.

```
MOV A,#10
CALL proc
MOV R2,#0
.....
proc: MOV R2,#13
ADD A,R2
RET
```

RET powrót z procedur / koniec procedury

4.7. Pozostałe instrukcje adresowe

MOVX służy do przesyłania danej pomiędzy akumulatorem a zewnętrzną pamięcią danych. Wykonanie tej instrukcji uaktywnia sygnały /RD (przy odczycie z zewnętrznej pamięci) lub /WR (przy zapisie) procesora – piny P3.7 i P3.6. Dodatkowo porty P0 i P2 pełnią wtedy rolę magistrali systemowej dzięki której wystawiany jest adres oraz przekazywana dana do zewnętrznej pamięci danych.

Tryb pełnego adresu (16-bitowego)

MOVX A, @DPTR – do akumulatora zostaje załadowana dana (bajt) z zewnętrznej pamięci danych (odczyt z zewnętrznej pamięci danych) spod adresu w DPTR, cykle: 2 bajty: 1 aby odczytać zawartość komórki z zewn. RAM o adresie np. 1240h należy wykonać **MOV DPTR, #1240h** - załadowanie adresu (16-bit) do wskaźnika DPTR
MOVX A, @A+DPTR - odczyt danej spod tego adresu; dana znajduje się w akumulatorze

MOVX @DPTR, A – do komórki zewnętrznej pamięci danych o podanym w DPTR adresie zostaje przesłana zawartość akumulatora, cykle: 2 bajty: 1 aby zapisać daną w obszarze zewnętrznej pamięci danych pod adresem np. 8000h należy wykonać instrukcje:

```
MOV A, ..... ;w miejsce kropek należy wpisać źródło danej
MOV DPTR, #8000h - wprowadzamy adres zapisu
MOVX @DPTR,A - zapisujemy daną w zewn. pamięci
```

Tryb stronicowania (niepełnego adresu)

MOVX A, @Ri – do akumulatora zostaje przesłana zawartość komórki w obszarze zewn. pamięci danych spod adresu znajdującego się w rejestrze R0 (i=0) lub R1 (i=1): adres 8b. Linie adresowe A0...A7 pamięci są dołączone do zatrasku młodszej części adresu szyny procesora (patrz poprzednie odcinku cyklu). Trzy starsze linie A8...A10 są dołączone np. do pinów P2.0, P2.1 i P2.2 procesora, pozostałe końcówki portu P2 (P2.2...P2.7) są wykorzystywane np. do sterowania przekaźnikami jakiegoś urządzenia zewnętrznego. Aby odczytać daną z tej pamięci np. spod adresu 24h na stronie pierwszej (strony liczone od 0 do 7, bo 2kB / 256 = 8 stron) należy wykonać instrukcje:

CLR P2.2 ;wyzerowanie linii adresowej A10

CLR P2.1 ;wyzerowanie linii adresowej A9

SETB P2.0 ;ustawienie linii adresowej A8 (strona: 1)

MOV R1, #24h ;załadowanie adresu komórki do wskaźnika

MOVX A, @R1 ;i przesłanie jej zawartości do akumulatora

Przy takim zaadresowaniu pamięci nie uległy modyfikacji piny P2.3...P2.7 portu P2 procesora, co w wielu przypadkach jest wręcz niezbędne. Można by oczywiście zaadresować tę pamięć za pomocą instrukcji

MOVX A, @DPTR (podając wtedy adres MOV DPTR, #0124h), ale wtedy zniszczeniu uległy by stany pozostałych, nie dołączonych do pamięci, końcówek portu P2.

MOVX @Ri, A – do obszaru zewnętrznej pamięci danych o adresie znajdującym się w rejestrze Ri (R0 gdy i=0 lub R1 gdy i=1) zostaje przesłana zawartość akumulatora. Innymi słowy jest to zapis do zewnętrznej pamięci danych. Daną znajdującą się w akumulatorze zapisujemy do komórki o adresie 00h na stronie 7 (innymi słowy fizyczny adres komórki będzie równy: 0700h), w tym celu należy wykonać ciąg instrukcji:

SETB P2.2 ;ustawienie linii adresowej A10

SETB P2.1 ;ustawienie linii adresowej A9

SETB P2.0 ;ustawienie linii adresowej A8 (strona: 7)

MOV R1, #00h ;załadowanie adresu komórki do wskaźnika

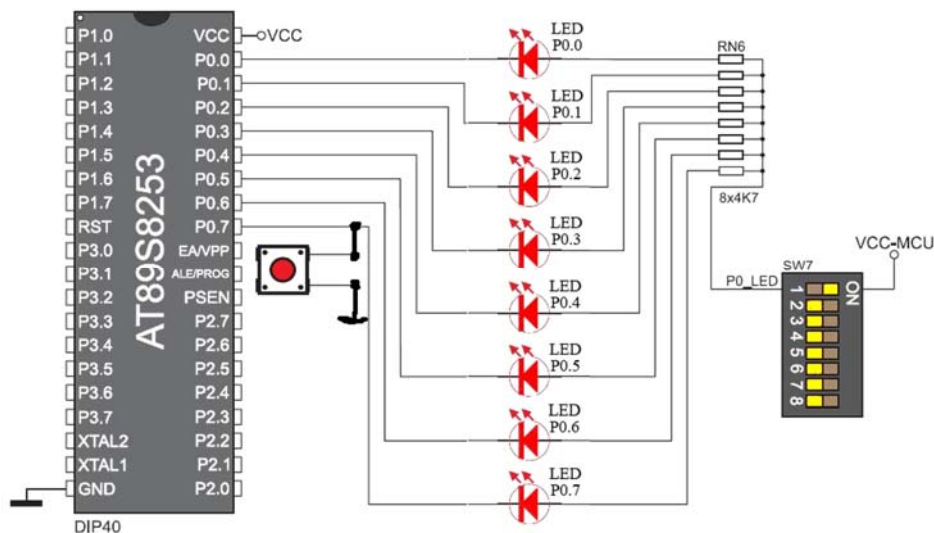
MOVX @R1, A ;i zapisane danej z akumulatora w zewn. RA

MOVC A, @A + DPTR, prześlij do akumulatora bajt z komórki pamięci kodu programu o adresie wyznaczonym jako suma A i DPTR

5. Obsługa portów we/wy

Odczyt stanu portu może być przeprowadzony na dwa sposoby - jako odczyt stanu rejestru portu, bądź jako odczyt stanu linii wejściowych (wyprowadzeń) portu. Sposób wykonywania odczytu zależy od rodzaju instrukcji wykorzystanej do odczytu portu. Instrukcje, które odczytują zawartość rejestru (a nie wyprowadzeń portu) określane są mianem instrukcji odczyt-modyfikacja-zapis, ponieważ wykonywanie tych instrukcji obejmuje odczyt zawartości rejestru portu, zmianę odczytanej wartości i zapis zmodyfikowanej wartości do rejestru portu. Instrukcjami odczyt-modyfikacja-zapis są następujące instrukcje: **ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ, CLR, SETB, MOV PX.y, c** (przesłanie zawartości wskaźnika przeniesienia na bit y portu x, np. MOV P1.0, C).

*Zworki sterujące płyty EASY 8051 v6 są ustawione tak, że dioda zaświeci się, **gdy wymuszony zostanie na danej linii portu stan niski** przez naciśnięcie przycisku lub ustawienie odpowiedniego wyjścia przez mikrokontroler.*



Rys. 6. Schemat podłączenia diod

cpl PX.y- zmienia stan wyjścia y portu PX na przeciwny, gasi/zapala podłączoną diodę,
setb PX.y - ustawia stan wysoki "1" wyjścia y portu PX, dioda wyłączona/nie świeci (anoda i katoda podłączone do zasilania),

clrb PX.y - ustawia stan niski "0" wyjścia y portu PX, dioda włączona/świeci (anoda podłączona do zasilania, katoda do masy przez PX.x).

Aby proces załączania/wyłączania diody był widoczny (migająca dioda), musi nastąpić na zauważalny dla człowieka okres czasu nie krótszy niż 0,1 sekundy.

6. Obsługa przerwań

Warunek przerwania (kiedy znacznik danego przerwania zostaje ustawiony), np. kiedy nadejdzie zbocze opadające na wejściu /INT1, przy ustawionym bicie EX1 oraz uaktywnionym systemie przerwań (EA=1) procesor wykona następujące operacje:

- sprawdzi czy nie jest wykonywana inna procedura obsługi przerwania o wyższym priorytecie lub czy jednocześnie nie nadeszło takie przerwanie
- (jeżeli nie zdarzy się poprzedni warunek): wyzeruje znacznik zgłoszenia przyjętego przerwania - wyjątek, nie są automatycznie zerowane znaczniki przerwań: z portu szeregowego TI – przy wysłaniu znaku, RI – przy nadejściu znaku, oraz z licznika T2
- zapisze na stosie zawartość 16-bitowego licznika rozkazów PC
- automatycznie wpisze do licznika rozkazów PC ustalony fabrycznie adres początku programu (procedury) obsługi danego przerwania.

Zarezerwowane adresy obsługi przerwań w pamięci programu:

(obszaru pamięci od 03h do 30h nie wolno przeznaczac na inną część programu)

0003h – dla przerwania z wejścia /INT0

000Bh – dla przerwania z licznika T0

0013h – dla przerwania z wejścia /INT1

001Bh – dla przerwania z licznika T1

0023h – dla przerwania z portu szeregowego

002Bh – dla przerwania z licznika T2

Ze względu na tylko 8 bajtów przeznaczone na podprogram, nie zapisuje się w tym miejscu podprogramu, a jedynie wskaźnik (wektor) pokazujący gdzie w programie (pod jakim adresem) znajduje się właściwa procedura obsługi danego przerwania.

Program obsługi przerwania musi być zakończony instrukcją **RETI**. Do tej instrukcji nie zostanie przyjęte zgłoszenie żadnego przerwania z poziomu równego (tego samego) lub niższego. Kiedy procesor wykona instrukcję kończącą podprogram przerwania (RETI) odtwarzany jest ze stosu adres powrotu sprzed wywołania i wpisany do licznika rozkazów PC procesora, po czym program główny toczy się dalej.

7. Przykładowa struktura programu w języku assembler

```
ORG 0000h      ;początek wykonywania programu
.....        ;instrukcje inicjujące (początkowe)
SETB EA       ;globalne odblokowanie przerwania
SETB EX0     ;uaktywnienie przerwania z /INT0
SETB ET1     ;uaktywnienie przerwania z T1

JMP START     ;skok do etykiety początku programu głównego
              ;tablica wektorów przerwania

ORG 0003h     ;pod adresem 0003h wektor, czyli instrukcję skoku do procedury intEX0
jmp intEX0

ORG 001Bh    ;pod adresem 001Bh wektor do procedury obsługi przerwania od T1
jmp intT1

              ;właściwy początek pętli głównej programu
START:
.....        ;instrukcje pętli głównej
jmp START

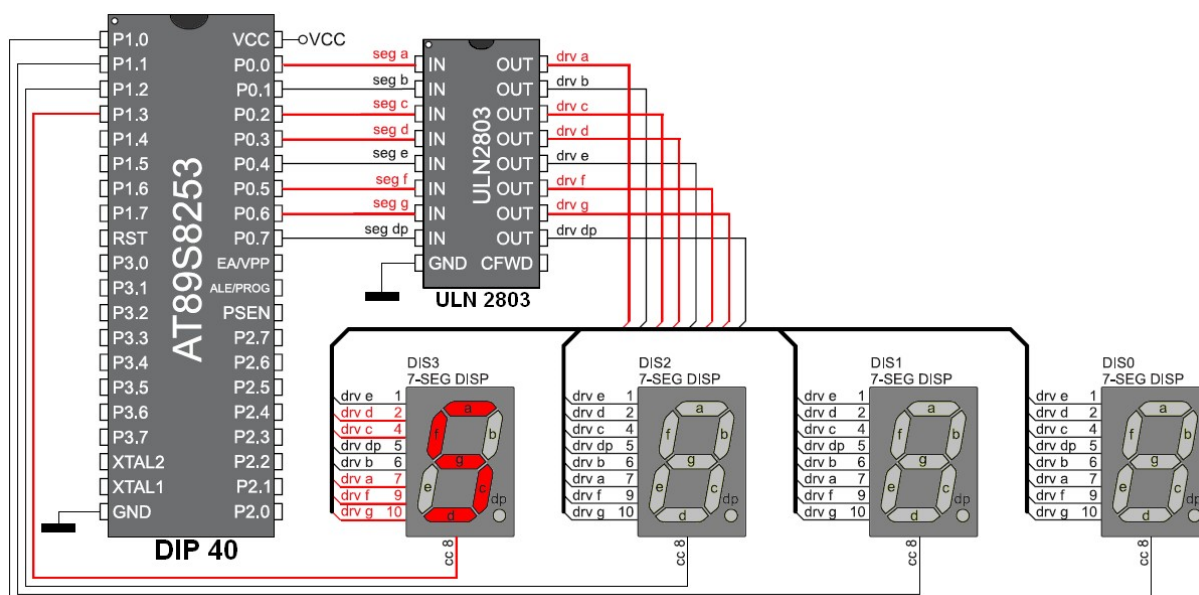
              ;tu początek podprogramów obsługi przerwania
intEX0:
.....        ;instrukcje dotyczące procedury w przypadku
.....        ;nadejścia przerwania z /INT0
reti

intT1:
.....        ;instrukcje dotyczące procedury w przypadku
.....        ;nadejścia przerwania z T1
reti

END           ;koniec programu
```

8. Wyświetlacz 7-segmentowy zestawu EASY 8051v6

Port P0 steruje pracą poszczególnych sekcji każdego z wyświetlaczy. Linia P1.0 aktywuje pierwszy z czterech wyświetlaczy, pozostałe są aktywowane przez odpowiednie linie od P1.1 do P1.3 zgodnie ze schematem przedstawionym na rysunku 7.



Rys. 7. Schemat podłączenia wyświetlacz 7-segmentowy zestawu EASY 8051v6

Wykorzystany do pośredniego sterowania segmentami wyświetlacza układ scalony ULN2803 zawiera 8 tranzystorów w układzie Darlingtona o dopuszczalnym obciążeniu do 500 mA oraz 8 diod zabezpieczających Zenera i pozwala na pośrednie sterowanie prądem sekcji wyświetlacza przez poszczególne wyjścia mikrokontrolera. Bezpośrednie sterowanie jest nie możliwe ze względu na zbyt niski dopuszczalny prąd obciążenia wyjść mikrokontrolera (Port A dopuszczalne obciążenie do 8 układów TTL, pozostałych portów do 4 układów TTL).

Tabela sterowania segmentami wyświetlacza za pomocą linii portu P0 w celu zapalenia sekcji tworzących cyfry:

P0								Cyfra
0	1	2	3	4	5	6	7	
0	0	0	0	0	0	1	1	0
1	0	0	1	1	1	1	1	1
0	0	1	0	0	1	0	1	2
0	0	0	0	1	1	0	1	3
1	0	0	1	1	0	0	1	4
0	1	0	0	1	0	0	1	5
0	1	0	0	0	0	0	1	6
0	0	0	1	1	1	1	1	7
0	0	0	0	0	0	0	1	8
0	0	0	0	1	0	0	1	9

0 - segment zapalony, 1 - segment zgaszony; linia P0.7 steruje kropką;

9. Reset mikrokontrolera

Zerowanie polega na wykonaniu przez mikrokontroler szeregu operacji, wskutek których znaczna część wewnętrznych rejestrów SFR przyjmuje określone przez producenta wartości. Dzięki temu początkowe warunki pracy mikrokontrolera po jego wyzerowaniu są zawsze jednakowe. W szczególności w wyniku zerowania mikrokontrolera licznik programu przyjmuje wartość 0000h, wskaźnik stosu wartość 07h, a rejestry wyjściowe portów wartość 0FFh. Oznacza to, że kod pierwszej instrukcji wykonanej po zakończeniu operacji zerowania zostanie pobrany spod adresu 0000h pamięci programu, pierwszy bajt umieszczony na stosie znajdzie się pod adresem 08h wewnętrznej pamięci RAM mikrokontrolera, a wszystkie linie portów zostaną ustawione w stan wysoki (z tym, że np. linie portu P0 przyjmą stan wysokiej impedancji ze względu na brak wewnętrznych rezystorów podciągających). W wyniku zerowania rejestry sterujące pracą wewnętrznych układów peryferyjnych mikrokontrolera przyjmują z reguły takie wartości, by dany układ pozostał nieaktywny. Zerowanie mikrokontrolera nie powoduje zmian zawartości wewnętrznej pamięci RAM.