

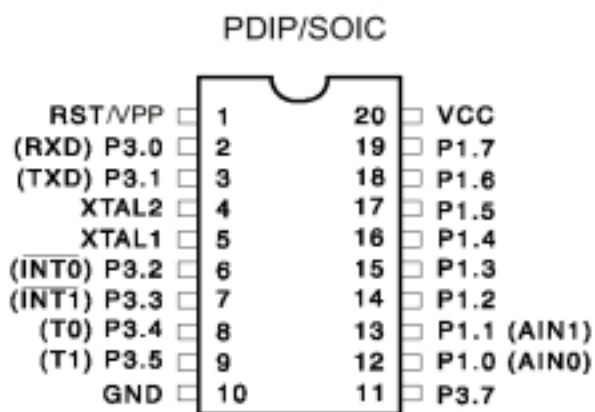
Opis mikrokontrolera AT89C2051

Cechy mikrokontrolera AT89C2051:

- kompatybilny z układami rodziny MCS-51,
- 2kB wewnętrznej pamięci typu Flash-EPROM ,
- zegar: 0Hz do 24MHz ,
- 8-bitowa jednostka centralna,
- 128B wewnętrznej pamięci RAM,
- 15 programowalnych linii we/wy,
- dwa szesnastobitowe liczniki/zegary,
- interfejs szeregowy,
- wbudowany analogowy komparator.

Układ 89C2051 realizuje 111 instrukcji, z czego 49 jest jednobajtowych, 45 dwubajtowych, 17 trybajtowych, 64 jest wykonywane w jednym cyklu. Cechą charakterystyczną jest duża liczba operacji na bitach i instrukcje mnożenia i dzielenia.

Rozkład wyprowadzeń procesora 89C2051 przedstawiono poniżej. Cechą charakterystyczną układu jest możliwość alternatywnego wykorzystania większości linii we/wy.



Wyprowadzenia mikroprocesora AT89C2051

PORT 1 (P1.0...P1.7) – są to wyprowadzenia 8-bitowego, uniwersalnego portu mikroprocesora oznaczonego jako P1.

Port może pełnić rolę wyjścia informacji binarnej. Tak więc, jeżeli zachodzi potrzeba; procesor może np. wpisać do portu P1 dowolną liczbę binarną z zakresu 0...255, np. 48. Binarnie liczba 48 = 00110000. Oznaczenia poszczególnych końcówek portu P1 wskazują na kolejną pozycję bitu. Tak więc końcówka P1.7 (najstarsza) przyjmie poziom logiczny 0, końcówka P1.6 poziom 0, P1.5 poziom 1 itd. Zapisanie jakiejś liczby do portu P1 daje wiele zastosowań. Na przykład do każdego wyprowadzenia portu można dołączyć układ z przekaźnikiem, którego styki załączają dowolne urządzenie elektryczne. Istotną zaletą portów uniwersalnych procesora jest możliwość indywidualnego ustawiania poziomu logicznego na każdym wyprowadzeniu niezależnie. Nie trzeba zatem zapisywać całej liczby do portu aby np. zmienić stan tylko na jednym wyprowadzeniu, wystarczy ustawić (rozkazem zwanym SETB) lub wyzerować (rozkazem CLR) odpowiedni bit rejestru portu P1, toteż np. ustawienie pinu P1.5 na logiczne 0 nastąpi poprzez wydanie polecenia: CLR P1.5 (clr – clear, ang. zeruj, wyczyść).

Port (cały lub niektóre z jego pinów), podobnie jak przy zapisie, można ustawić także jako wejście informacji logicznej. Każde z wyprowadzeń staje się wtedy wyjściem o wysokiej impedancji, dzięki temu dowolny poziom logiczny podany z wyjścia jakiegoś układu cyfrowego może być odczytany poprzez piny portu a informacja czy tym stanem była logiczna 1 lub 0, zostaje wykorzystana przez procesor dla dalszego jego działania w zależności od spełnianej akurat funkcji. Krótko mówiąc, procesor może odczytać stany logiczne, jakie z zewnątrz podano na końcówki portu.

Oczywiście poziomy logiczne napięcie wejściowych portu P1 muszą zawierać się w przedziale napięcie zasilania mikrokontrolera, czyli w zakresie 0...5V. Detekcja poziomów logicznych odbywa się jak dla bramek CMOS, stąd wartości progowe napięcie tych stanów są zbliżone do połowy napięcia zasilającego. Istotną informacją jest fakt że w trybie odczytu z portu P1 końcówki są wewnętrznie podczepiane do plusa zasilania poprzez wbudowane w procesor rezystory, co wymusza odczyt wysoki z portu w przypadku niepodłączenia końcówki portu.

Końcówki portu P1,1 (AIN1) i P1,0 (AIN0) są to wejścia na analogowy komparator. AIN0 jest to nieodwracające wejście komparatora, a AIN1 odwracające.

PORT 3 (P3.0...P3.5,P3.7) – podobnie jak w przypadku portu P1, port P3 może pełnić wszystkie opisane wcześniej funkcje – może być wyjściem lub wejściem. Dodatkowe symbole obok wyprowadzeń portu P3 sugerują że port może spełniać inne dodatkowe funkcje, co jest zresztą zgodne z prawdą.

Piny P3.0 (RXD) i P3.1 (TXD) mogą pełnić rolę portu transmisji szeregowej. W praktyce poprzez te dwa wyprowadzenia można przesyłać informację (bajty i bity) z i do procesora z innych układów cyfrowych w sposób szeregowy tzn. bit po bicie. Ciekawostką niech będzie fakt, że przesyłanie to może odbywać się na kilka sposobów:

- synchronicznie – wtedy pin P3.0 pełni rolę dwukierunkowej magistrali szeregowej, po której przesyłane są dane, zaś pin P3.1 generuje sygnał taktujący, pełniąc rolę zegara,
- asynchronicznie – kiedy z góry zadajemy prędkość transmisji pomiędzy naszym procesorem a innym, zewnętrznym układem np. łączem RS232c komputera PC. W takim przypadku końcówka P3.0 – RXD pełni rolę odbiornika przesyłanych szeregowo danych, zaś końcówka P3.1 – TXD nadajnika.

Ponadto rozróżnia się kilka trybów pracy asynchronicznej.

Alternatywną funkcją końcówek P3.2 (INT0) oraz P3.3 (INT1) jest funkcja detekcji przerw zewnętrznych. Zmiana stanu logicznego z 1 na 0 powoduje przerwanie. Konsekwencją tego jest automatyczne przerwanie wykonywania przez procesor programu i natychmiastowe przejście do wykonania czynności ściśle określonych przez programistę. Ciąg takich czynności nazywany jest w technice mikroprocesorowej: „procedurą obsługi przerwania”. Wykrycie zmiany stanu logicznego na końcówkach przerw zewnętrznych INT0 i INT1 wiąże się ze spełnieniem jednego warunku, a mianowicie, aby czas od ujemnego zbocza sygnału zgłoszenia przerwania do ponownego przejścia w stan wysoki był odpowiednio długi. Czas ten zależy od częstotliwości zegara mikroprocesora.

Końcówki (P3.4 i P3.5) oznaczone jako T0 i T1 pełnią dodatkową funkcję wejść uniwersalnych, programowalnych liczników, wbudowanych w strukturę 89C2051. Procesor zawiera dwa bliźniacze liczniki T0 i T1. Maksymalnie mogą one zliczać do 65536, po czym zostają wyzerowane. Liczniki te oprócz zliczania impulsów z wejść T0 i T1 mogą także zliczać impulsy wewnętrzne, pochodzące z generatora mikrokontrolera. W praktyce wykorzystywane jest to np. do odmierzania określonych odcinków czasu np. przy funkcji zegarka.

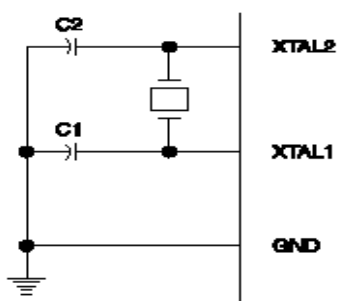
RST – zerowanie układu. Czynność ta wykonywana poprzez podanie logicznej 1 na tę wyprowadzenie na pewien okres czasu powoduje skasowanie układu, a więc natychmiastowe przerwanie wykonywanych czynności i rozpoczęcie cyklu działania procesora od samego początku. Czas trwania dodatniego impulsu kasującego zależy od częstotliwości z jaką pracuje mikroprocesor. Z reguły w typowych zastosowaniach czas 1ms w zupełności wystarcza. W układach praktycznych do końcówki RST dołącza się mniej lub bardziej skomplikowany układ który generuje wymagany impuls zerujący najczęściej w trzech przypadkach:

- po włączeniu zasilania układu,
- na nasze żądanie – poprzez np. przyciśnięcie klawisza,
- w sytuacjach awaryjnych, kiedy np. poprzez zakłócenie najczęściej na liniach zasilających nastąpi błędne działanie układu mikroprocesora, inaczej „zawieszenia”.

Trzeci przypadek dotyczy bardziej złożonych układów stosowanych szczególnie w automatyce i elektronice przemysłowej.

VPP – napięcie programujące.

XTAL1 i XTAL2 – końcówki te służą do dołączenia zewnętrznego rezonatora kwarcowego o częstotliwości zależnej od potrzeb użytkownika. W praktyce częstotliwość ta może wynosić od 0Hz do 24MHz. Dołączony do tych pinów rezonator kwarcowy po uzupełnieniu o dodatkowe kondensatory o wartości z reguły 22...40pF (w zależności od wartości rezonatora), umożliwia pracę wbudowanemu w 89C2051 generatorowi, który „napędza” cały mikroprocesor. Oczywiście od częstotliwości rezonatora ściśle zależy szybkość działania mikrokontrolera. Typowy układ zewnętrznego oscylatora przedstawia rysunek poniżej:



Układ zewnętrznego oscylatora

Częstotliwość, z jaką pracują wewnętrzne układy mikroprocesora, jest określona wzorem:

$$F = \frac{f_{xtal}}{12},$$

gdzie f_{xtal} jest częstotliwością rezonatora kwarcowego. Powodem takiego podziału częstotliwości rezonatora jest wewnętrzna architektura procesora.

GND – masa.

VCC – jest to końcówka zasilania mikroprocesora. Napięcie względem masy z reguły nie może przekraczać 6,5V. Dlatego układ mikrokontrolera należy zasilac napięciem 5V $\pm 0,25V$ używając do tego celu dowolnego zasilacza stabilizowanego najlepiej przy pomocy znanego układu 7805. Zasadą przy projektowaniu układów z 89C2051 jest blokowanie tego wyprowadzenia kondensatorem o wartości 100nF do masy układu cyfrowego.

Wewnętrzna pamięć programu

Program napisany przez użytkownika, przewidziany dla konkretnego zastosowania mikroprocesora, powinien zostać umieszczony wewnątrz mikrokontrolera – czyli w wewnętrznej pamięci programu. Pamięć ta służy mikrokontrolerowi wyłącznie do odczytu rozkazów programu. W pamięci tej mogą być umieszczone także argumenty bezpośrednie rozkazów oraz tablice ze stałymi potrzebnymi do pewnych działań programu, np. tablica sinusów, tablica czasów zachodu słońca, lub cokolwiek innego. Mikroprocesor 89C2051 ma możliwość późniejszego pobrania ze swojej pamięci programu takiej stałej i wykorzystania jej np. w obliczeniach.

Po włączeniu zasilania dzięki obwodowi „Reset”, wyzerowane zostają prawie wszystkie wewnętrzne układy mikroprocesora w tym także „licznik rozkazów”. Ten ostatni służy mikroprocesorowi do kolejnego pobierania rozkazów z pamięci programu, a

dokładnie do adresowania (czyli wskazywania) gdzie w przestrzeni adresowej pamięci programu znajduje się kolejna komenda. Jego początkowa wartość wynosi 0, toteż pierwszym rozkazem pobranym z tej pamięci będzie ten umieszczony pod adresem 0000h.

Licznik rozkazów oznaczany jest w skrócie jako PC z angielskiego „Program Counter” – licznik programu (rozkażów). Licznik PC ma długość 16 bitów, czyli maksymalnie może liczyć do 65535 włącznie, po czym zostaje wyzerowany.

W trakcie pobierania i wykonywania przez mikrokontroler kolejnych instrukcji, licznik PC zmienia swoją wartość zawsze wskazując na aktualny adres kolejnego rozkazu w pamięci programu. Maksymalną wartość jaką może osiągnąć licznik w tym przypadku to 2048 – bowiem w układzie AT89C2051 mamy do dyspozycji 2kB pamięci programu.

Oprócz wspomnianego miejsca „startowego” programu – czyli adresu 0000h, w przestrzeni adresowej pamięci programu istnieje kilka innych istotnych dla programisty miejsc. W celu ujednoczenia systemu przerwań procesora w pamięci programu określono odpowiednie miejsca – adresy od których rozpoczyna się wykonywanie określonych procedur obsługi przerwań. W podstawowej rodzinie ’51 są to adresy: 3, 11, 19, 27, 35 i 43 (03h, 0Bh, 13h, 1Bh, 23h, 2Bh szesnastkowo). Każdy z tych adresów określa początek wykonania innej procedury obsługi przerwania, dla 89C2051 są one następujące:

- 0003h – przerwanie zewnętrzne z wejścia (końcówki) INT0,
- 000Bh – przerwanie wynikłe z przepełnienia pierwszego wewnętrznego licznika T0 procesora,
- 0013h – przerwanie zewnętrzne z wejścia (końcówki) INT1,
- 001Bh – przerwanie wynikłe z przepełnienia drugiego wewnętrznego licznika T1 procesora,
- 0023h – przerwanie wynikłe z odebrania lub zakończenia wysyłania danej poprzez wewnętrzny port szeregowy mikroprocesora.

Na rysunku poniżej zilustrowano rozmieszczenie wyżej wymienionych adresów zgłoszenie przerwania:

| | |
|----------------------------------|-------|
| Pozostała część pamięci programu | 7FFh |
| Przerwanie z portu szeregowego | 023h |
| Przerwanie od T1 | 01Bh |
| Przerwanie od INT1 | 013h |
| Przerwanie od T0 | 00Bh |
| Przerwanie od INT0 | 003h |
| Reset procesora | 000h |
| PAMIĘĆ PROGRAMU | Adres |

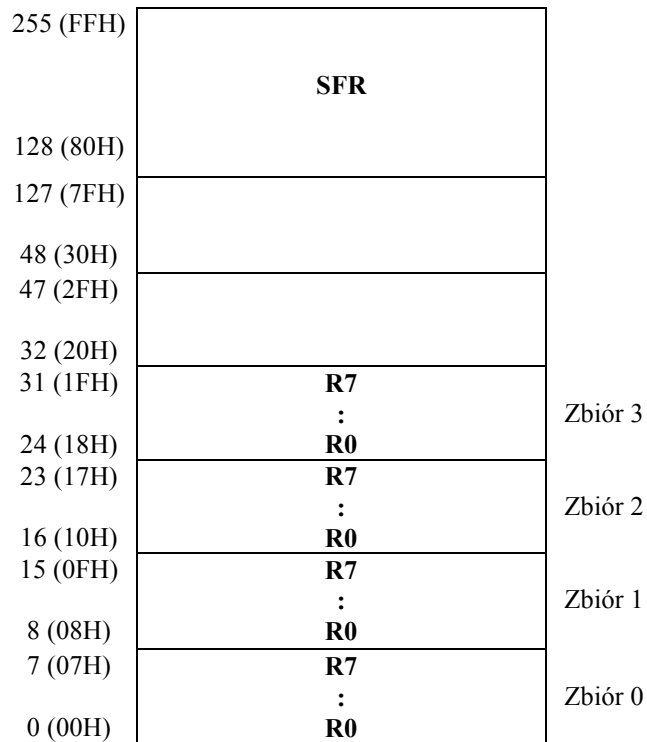
Rozmieszczenie adresów zgłoszeń przerwania

Praktycznie wygląda to tak, że w momencie zgłoszenia któregoś z wymienionych przerwania, automatycznie zachowana zostaje aktualna wartość licznika PC, a następnie zostaje wpisana do niego wartość odpowiednia od rodzaju przerwania. Czyli np. jeżeli wewnętrzny licznik procesora T1 został przepełniony, do PC zostaje wpisana wartość 001Bh, po czym mikroprocesor rozpoczyna wykonywanie programu od tego adresu w pamięci programu. Po zakończeniu wykonywania czynności związanych z przepełnieniem T1, licznik rozkazów PC przyjmie ponownie wartość jak z przed nadejścia przerwania i program „poczty się” dalej.

Wewnętrzna pamięć danych

W mikrokontrolerze pamięć ta przeznaczona jest dla użytkownika do przechowywania argumentów wartości zmiennych oraz wyników obliczeń arytmetyczno – logicznych. Pamięć ta ma pojemność 128 bajtów.

Na rysunku poniżej przedstawiono organizację wewnętrznej pamięci danych:



Wewnętrzna pamięć danych w mikrokontrolerze

W przestrzeni tej pamięci można wyróżnić kilka obszarów. Dwa główne, obszar pamięci użytkowej oraz obszar rejestrów specjalnych SFR. Pamięć użytkowa zajmuje 128 komórek, adresy: 0 – 127 (00h – 7Fh), natomiast obszar SFR obejmuje adresy 128 – 255 (80h – FFh), z tym że nie wszystkie są wykorzystywane przez rejestry specjalne.

I chociaż pamięć użytkownik podzielona jest na obszary, do których dostęp może odbywać się przez tzw. indeksowanie obszaru, to użytkownik może adresować ją poprzez proste adresowanie.

W pamięci użytkowej komórki o adresach 0...7, 8...15, 16...23 i 24...31 tworzą cztery zbiory uniwersalnych rejestrów roboczych. Każdy z rejestrów oznacza się symbolami R0...R7. W danej chwili użytkownik ma możliwość dostępu (poprzez nazwy R0...R7) tylko do jednego „banku” (zbioru) rejestrów roboczych. Przełączanie zbiorów

odbywa się poprzez odpowiednie ustawienie dwubitowego wskaźnika zwanego jako RS – z angielskiego „Register bank Switch”.

Rejestry R0 i R1 z aktywnego banku pełnią rolę wskaźników danych do pośredniego adresowania wewnętrznej pamięci danych jak i zewnętrznej. W przypadku adresowania pamięci wewnętrznej można adresować cały obszar czyli adresy 0...7Fh.

Na rysunku poniżej przedstawiono mapę specjalnych rejestrów funkcyjnych:

| | | | | | | | | |
|-----|------------------|------------------|-----------------|-----------------|-----------------|-----------------|------------------|-----|
| F8H | | | | | | | | FFH |
| F0H | B 00000000 | | | | | | | F7H |
| E8H | | | | | | | | EFH |
| E0H | ACC 00000000 | | | | | | | E7H |
| D8H | | | | | | | | DFH |
| D0H | PSW 00000000 | | | | | | | D7H |
| C8H | | | | | | | | CFH |
| C0H | | | | | | | | C7H |
| B8H | IP XXX00000 | | | | | | | BFH |
| B0H | P3 11111111 | | | | | | | B7H |
| A8H | IE 00XX0000 | | | | | | | AFH |
| A0H | | | | | | | | A7H |
| 98H | SCON 00000000 | SBUF XXXXXXXX | | | | | | 9FH |
| 90H | P1 11111111 | | | | | | | 97H |
| 88H | TCON 00000000 | TMOD 00000000 | TL0 00000000 | TL1 00000000 | TH0 00000000 | TH1 00000000 | | 8FH |
| 80H | | SP 00000111 | DPL 00000000 | DPH 00000000 | | | PCON 0XXX0000 | 87H |

Mapa specjalnych rejestrów funkcyjnych - SFR

- ACC (E0h)** - akumulator,
- B (F0h)** - rejestr B,
- PSW (D0h)** - słowo stanu programu,
- SP (81h)** - 8-bitowy wskaźnik stosu,
- DPH (83h)** - bity 8 – 15 wskaźnik danych DPTR,
- DPL (82h)** - bity 0 –7 wskaźnik danych DPTR,
- P1 (90h)** - port 1,

- P3 (B0h)** - port 3,
- IP (B8h)** - rejestr sterujący priorytetem przerwania,
- IE (A8h)** - rejestr kontrolny sterujący pracą systemu przerwania,
- TCON (88h)** - rejestr kontrolny pracy liczników T0 i T1 oraz przerwania INT0 i INT1,
- TMOD (89h)** - rejestr sterujący trybem pracy liczników T0 i T1,
- TH0 (8Ch)** - bity 8 – 15 16-bitowy licznik T0,
- TL0 (8Ah)** - bity 0 – 7 16-bitowy licznik T0,
- TH1 (8Dh)** - bity 8 – 15 16-bitowy licznik T1,
- TL1 (8Bh)** - bity 0 – 7 16-bitowy licznik T1,
- SCON (98h)** - rejestr sterujący portem szeregowym,
- SBUF (99h)** - bufor portu szeregowego,
- PCON (87h)** - rejestr sterujący zasilaniem.

W tym miejscu warto zapamiętać iż rejestry specjalne stanowią niejako sprzętowy „pomost” komunikacyjny pomiędzy programistą a wszystkimi blokami funkcjonalnymi mikrokontrolera.

W przestrzeni adresowej SFR znajdują się także rejestry będące jednocześnie portami wejścia-wyjścia. Dzięki temu możliwy jest łatwy i szybki dostęp do dowolnych bitów portu czyli fizycznie do jego wyprowadzeń. Zapis do odpowiedniego rejestru portu spowoduje pojawienie się kombinacji na końcówkach mikrokontrolera, odczyt rejestru pozwoli użytkownikowi na zbadanie poziomu logicznego na wybranej linii portu.

Nie wszystkie 128 adresów z przestrzeni SFR jest wykorzystanych. „Puste” adresy nie nadają się do wykorzystania przez użytkownika. Nie jest to bynajmniej marnotrawienie cennych bajtów pamięci, lecz czysta przezorność projektantów, którzy konstruując rozszerzone wersje ‘51-ki wyposażają je w nowe dodatkowe bloki funkcjonalne, a w wolnych miejscach przestrzeni SFR umieszczane są dodatkowe rejestry sterujące ich pracą (w mikrokontrolerze 89C2051 zostały usunięte rejestry P0 i P2 gdyż usunięte zostały porty P1 i P2).

Pamięć FLASH

Pojedyncza komórka pamięci ma postać obszaru przewodzącego elektrycznie, otoczonego obszarem o własnościach izolatora. Doprowadzony do komórki ładunek elektryczny nie może jej opuścić – pamięć jest nieulotna – przechowuje więc informacje mimo odłączenia źródła zasilania. Przyłożenie dostatecznie dużego napięcia do izolatora otaczającego ładunek, umożliwi ładunkowi pokonanie bariery potencjału (przebicie) i wydostanie się na zewnątrz. Wykorzystując ten mechanizm można kasować jednocześnie całe grupy komórek pamięci, a następnie ponownie zapisać nowymi ładunkami. Jest to więc pamięć programowalna elektrycznie.

Procesory AT89C2051 są przystosowane do programowania napięciem 12V.

Stos i wskaźnik stosu

Najprościej można stos określić jako bardzo prostą w działaniu strukturę przechowującą bajty. Pod pojęciem „przechowania” rozumiemy oczywiście operacje zapisu, a następnie odczytu dowolnej zmiennej lub rejestru SFR.

W przypadku takich operacji tylko z udziałem np. wewnętrznej pamięci danych użytkownika, aby dokonać zapisu (odczytu) trzeba daną komórkę pamięci najpierw zaadresować – czyli po prostu podać jej fizyczny adres.

W przypadku korzystania ze stosu adresowanie jest niekonieczne. Przy takim sposobie obsługi konieczne jest jednak zachowanie odpowiedniej kolejności w zapisie i odczycie tak aby dane nie „pomieszały się”.

Otóż taki uporządkowany sposób przechowywania danych charakteryzuje właśnie stos. Wszystkie dane (bajty) przy zapisie odkładane są „na stos” jedna na drugą. Na wierzchołku stosu znajduje się zawsze ostatnio odłożona dana (np. oznaczona jako X), toteż aby „dobrać się” do danej leżącej pod nią (Y) należy najpierw „zdjąć” ze stosu daną X, a potem dopiero odczytać Y.

Stos służy do przechowywania zmiennych lub rejestrów SFR, a dostęp do nich odbywa się w sposób uporządkowany.

Stos umieszczony jest w wewnętrznej pamięci danych użytkownika, czyli w obszarze o adresach 00h...7Fh.

Ilość pamięci zajętej przez stos będzie się zmieniać i zależeć od tego ile bajtów odłożonych jest na stosie.

Aby ściśle określić miejsce położenia stosu, w architekturze '51-ki znajduje się tzw. licznik stosu a fachowo mówiąc „wskaźnik stosu”. Fizycznie jest on po prostu 8-bitowym rejestrem w obszarze SFR, położonym pod adresem 81h. W mnemonice (nazewnictwie) procesorów posiada on symbol SP – z angielskiego „stack pointer” – wskaźnik stosu.

Jego zadaniem jest automatyczne wskazywanie miejsca aktualnego wierzchołka stosu. Tak więc w przypadku odłożenia bajtu na stos, wskaźnik SP jest automatycznie (bez ingerencji programisty) zwiększany o 1, w przypadku zdjęcia danej ze stosu jest on zmniejszany.

Podsumowując, stos jest hierarchiczną strukturą do przechowywania danych (bajtów) z obszaru wewnętrznej pamięci RAM procesora (włączając SFR) a położenie jego wierzchołka jednoznacznie określa jego wskaźnik – SP. Przy korzystaniu ze stosu obowiązuje zasada, „ile bajtów odłożyłeś na stos, tyle potem musisz zdjąć”, tak aby struktura stosu nie została zakłócona. W praktyce ma to szczególne znaczenie, bowiem stos wykorzystywany jest nie tylko poprzez świadome działanie użytkownika lecz także przechowywane są na nim ważne dla działania całego mikrokontrolera adresy powrotów z podprocedur oraz procedur obsługi przerwań, czyli innymi słowy mówiąc, aktualne zawartości 16-bitowego licznika rozkazów PC.

Ponieważ stos składa się z 8-bitowych komórek pamięci, a licznik rozkazów jest 16-bitowy to procesor na stos odkłada najpierw młodszy bajt rejestru PC, a następnie starszy bajt, wskaźnik stosu SP zostaje więc zwiększony automatycznie o 2. Tak więc w prosty sposób można przechowywać inne rejestry podwójne np. wskaźnik adresu zewnętrznej pamięci – DPTR, składający się z dwóch 8-bitowych rejestrów DPH (83h) oraz DPL (82h).

W przypadku rejestru DPTR jak i innych SFR przechowywanie na stosie odbywa się „na żądanie” użytkownika – w potrzebnym dla niego momencie.

Po włączeniu zasilania procesora (lub jego zresetowaniu) wskaźnik stosu SP przyjmuje domyślnie wartość 07h – czyli po prostu 7, wskazując tym samym że wierzchołek stosu – adres umieszczenia następnej danej – po odłożeniu jej na stos położony będzie w wewnętrznej pamięci danych pod adresem 08h ($07h + 1$).

Jeżeli więc odłożymy jakiś bajt na stos, najpierw licznik SP zostanie automatycznie zwiększony o 1 (08h), a następnie do komórki pamięci o tym adresie 08h, zostanie wpisany ten bajt. Przy zdjęciu ze stosu kolejność będzie odwrotna, najpierw zdjęty zostanie nasz bajt, a następnie zmniejszony zostanie wskaźnik SP o 1.

Wskaźnik stosu SP tak jak każdy rejestr SFR może być dowolnie modyfikowany przez programistę poprzez zapisanie w nim dowolnej 8-bitowej wartości (0...255).

W praktyce jednak sytuacja taka występuje tylko wtedy, jeżeli chcemy zmienić położenie stosu (czyli go przesunąć) na początku wykonywania programu. Operacja ta ma sens jeżeli stos w danej chwili jest „pusty”, w przeciwnym razie przy lekkomyślnej modyfikacji wskaźnika SP wszystkie dane odłożone wcześniej na stos staną się niedostępne (przynajmniej z punktu widzenia działania samego stosu).

Umiejętne i świadome korzystanie ze stosu przynosi często efekty w postaci znacznego przyspieszenia działania programu oraz zmniejszenia jego rozmiarów.

Jednostka arytmetyczno-logiczna

Pod tym pojęciem kryje się jeden z elementów architektury 89C2051 odpowiedzialny za wykonywanie operacji arytmetyczno-logicznych. Blok ten nazywany w skrócie jako ALU, potrafi wykonywać operacje na liczbach (składnikach) 8-bitowych.

Do wprowadzenia składników działania służą zarówno niektóre rejestry specjalne z grupy SFR jak i dowolna komórka wewnętrznej pamięci danych. Dla różnych działań występują jednak pewne ograniczenia w swobodzie umiejscawiania składników.

Jednym z najważniejszych rejestrów z grupy SFR jest akumulator oznaczany dużą literą A (ang. „accumulator”). Akumulator umieszczony jest pod adresem 0Eh (224 dziesiętnie). Rejestr ten służy jednostce ALU za miejsce pobrania argumentu oraz umieszczenia wyniku większości operacji arytmetyczno logicznych.

Rejestr ten może być adresowany bitowo, dzięki czemu możliwe jest testowanie dowolnych jego bitów bez potrzeby wykonywania dodatkowych operacji logicznych. Dodatkowo rejestr A poza funkcjami związanymi z jednostką ALU służy do pobierania i umieszczania bajtów w zewnętrznej pamięci danych.

Przy przesyłaniu tego rejestru na stos (umieszczenie lub pobranie ze stosu) wykorzystuje się adresowanie bezpośrednie tego rejestru. Wtedy opisujemy go symbolem ACC.

Drugim po akumulatorze ważnym rejestrze współpracującym z ALU jest, także 8-bitowy, rejestr B. Służy on do umieszczenia jednego ze składników mnożenia lub dzielenia, a po wykonaniu jednej z tych operacji w rejestrze tym umieszczany jest:

- w przypadku mnożenia starszy bajt 16-bitowego wyniku mnożenia dwóch liczb 8-bitowych,
- w przypadku dzielenia: reszta z dzielenia dwóch liczb 8-bitowych.

Oczywiście zarówno rejestr B jak i akumulator A mogą być wykorzystywane dowolnie jako rejestry uniwersalne.

Trzecim ważnym rejestrze związanym z ALU jest „słowo stanu programu” nazywane w skrócie jako PSW (od ang. „program status word”).

| | | | | | | | |
|---|----|----|-----|-----|----|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| C | AC | FQ | RST | RSQ | OV | - | P |

Zawartość rejestru stanu PSW

W skład tego rejestru wchodzi 8 bitów nazywanych znacznikami z których cztery informują o przebiegu wykonywania operacji arytmetyczno-logicznych. I tak:

- **PSW.0** (bit 0) – oznaczany jako P, to znacznik parzystości, ustawiany automatycznie w każdym cyklu maszynowym wskazuje na to czy liczba jedynek na poszczególnych pozycjach bitowych w akumulatorze A jest parzysta (P=1), czy nieparzysta (P=0).
- **PSW.2** (bit 2) – oznaczany jako OV, to znacznik przepełnienia (nadmiaru), ustawiany w wyniku wykonania dodawania lub odejmowania, a przy operacji dzielenia ustawienie go wskazuje na dzielenie przez zero.
- **PSW.6** (bit 6) – oznaczany jako AC, to znacznik przeniesienia pomocniczego, do którego wpisywane jest przeniesienie lub pożyczka z bitu 3, wykorzystywany jest przy korekcji dziesiętnej liczb.
- **PSW.7** (bit 7) – znacznik przeniesienia oznaczany jako C, do którego następuje przeniesienie z najbardziej znaczącego bitu w wyniku wykonania operacji logicznych

przesunięć liczb 8-bitowych lub w wypadku przekroczenia wyniku poza zakres liczb zapisanych w naturalnym kodzie dwójkowym (>255).

Pozostałe znaczniki nie mają związku z ALU.

Operacje jakie można wykonywać na liczbach 8-bitowych, to:

a) operacje arytmetyczne:

- dodawanie argumentów,
- dodawanie z przeniesieniem,
- odejmowanie z pożyczką.

W tych trzech przypadkach pierwszy z argumentów operacji (składnik lub odjemna) umieszczana jest w akumulatorze, drugi składnik lub odjemnik umieszczony jest w wewnętrznej pamięci danych, lub jest argumentem bezpośrednim rozkazu. Wynik działania umieszczany jest w akumulatorze. Dodatkowo w słowie PSW ustawiane są odpowiednio znaczniki: przeniesienia C i nadmiaru OV, co jest sygnałem przekroczenia zakresu liczb 8-bitowych odpowiednio bez lub ze znakiem.

Pozostałe operacje arytmetyczne to:

- mnożenie dwóch 8-bitowych liczb bez znaku, gdzie jeden składnik wpisywany jest do akumulatora drugi do rejestru B, 16-bitowy wynik umieszczany jest w rejestrach B i A – odpowiednio starszy bajt w B, młodszy w A,
- dzielenie dwóch liczb 8-bitowych, gdzie dzielna umieszczana jest w akumulatorze A, a dzielnik w B, 8-bitowy wynik dzielenia znajduje się po tej operacji w A, natomiast B przechowuje resztę z dzielenia,
- inkrementacja (zwiększanie o 1) lub dekrementacja (zmniejszanie o 1) akumulatora lub dowolnej komórki w wewnętrznej pamięci danych,
- korekcja dziesiętna wyniku zapisanego w akumulatorze.

b) operacje logiczne:

- logiczna suma (OR),
- iloczyn logiczny (AND),
- różnica symetryczna (EXOR),
- negacja (NOT) zawartość akumulatora A,
- przesuwanie cykliczne akumulatora w lewo lub prawo, z lub bez przeniesienia (znacznika C \rightarrow PSW.7).

Zegar systemowy

Do pracy, czyli do „poruszenia” całego procesora potrzebny jest zewnętrzny obwód oscylatora. W praktyce taki obwód realizuje się dołączając zewnętrzny rezonator kwarcowy o częstotliwości z zakresu 0...24MHz.

Wraz ze wzrostem częstotliwości pracy układu wzrasta wydzielana w nim moc, czyli wzrasta pobierany przez procesor prąd ze źródła zasilania.

Częstotliwość (F_{XTAL}) uzyskiwana z rezonatora kwarcowego jest we wnętrzu procesora kilkakrotnie dzielona. I tak w praktyce spotykamy się z następującymi pojęciami:

- dwufazowy sygnał taktujący procesor (F_S) – sygnał powstały z podzielenia przez 2 częstotliwości oscylatora (np. przy kwarcu = 12MHz, $F_S = 6$ MHz). Sygnał ten używany jest bezpośrednio do taktowania układów wewnętrznych procesora i nie jest dostępny na żadnym z zewnętrznych jego wyprowadzeń,
- sześć cykli sygnału F_S składa się na tzw. cykl maszynowy procesora, czyli okres wykonywania elementarnej czynności przez procesor. Z prostych obliczeń wynika, że cykl maszynowy zajmuje: $F_S \cdot 6 = 2 \cdot F_{XTAL} \cdot 6 = 12$ cykli oscylatora, czyli dla np. $F_{XTAL} = 12$ MHz będzie to 1MHz.

Cykl maszynowy jest bardzo ważnym pojęciem, z jego częstotliwością ($F_{XTAL}/12$) zachodzą podstawowe czynności procesora takie jak:

- pobieranie kodu rozkazów (czy to z wewnętrznej pamięci programu, czy z zewnętrznej),
- wykonywanie instrukcji programu,
- pobieranie danych z zewnętrznej pamięci (jak i z wewnętrznej),
- zwiększanie wartości wbudowanych liczników: T0 i T1,
- próbkowanie wejść zewnętrznych przerwań: INT0 i INT1.

Z częstotliwością tą taktowany jest także wbudowany port szeregowy w specjalnym trybie ustawionym przez użytkownika programowo.

Cykl maszynowy dzieli się także na fazy (po 6 na każdy cykl).

Układy czasowo-licznikowe

Pod pojęciem tym kryją się dwa 16-bitowe liczniki T0 i T1. Najogólniej mówiąc każdy z tych liczników a właściwie układów czasowo-licznikowych jest tak uniwersalnym blokiem że z wykorzystaniem jego można dokonać następujące dwie podstawowe operacje:

- za pomocą T0 można zliczać impulsy z zewnętrznego wejścia licznikowego,
- można zliczać wewnętrzne impulsy pochodzące z układu taktującego procesor.

W każdym przypadku będzie to sygnał o częstotliwości równej $F_{XTAL}/12$, jeżeli więc dopięty jest do mikrokontrolera kwarc o częstotliwości 6MHz to częstotliwość sygnału taktującego licznik T0 lub T1 będzie równa $6\text{MHz}/12=500\text{kHz}$. W tym trybie zwanym czasomierzem, licznik wykorzystuje się do odmierzenia pewnych określonych programowo przez użytkownika odcinków czasu (opóźnień) i generowania przerw po przepelnieniu któregoś z liczników.

W przypadku wykorzystania układu licznikowego w obu przypadkach należy wiedzieć że:

- maksymalna liczba zliczonych impulsów jest określona pojemnością 16-bitowego licznika, czyli 2 do potęgi 16 = 65536 (licznik zlicza od 0 do 65535 poczym po nadejściu kolejnego impulsu jest zerowany oraz z zależności od potrzeb jest generowane odpowiednie przerwanie),
- licznik można w dowolnym momencie uruchomić (zezwoić na zliczanie) lub zatrzymać wydając w programie odpowiednią komendę,
- do licznika można w każdej wpisać dowolną wartość (16-bitową liczbę), co spowoduje że licznik będzie zliczał impulsy od tej wartości aż do przepelnienia; wpisu takiego najlepiej jest dokonywać w czasie gdy licznik jest zatrzymany,
- dodatkowo licznik można „bramkować” czyli uzależnić jego pracę lub zatrzymanie w zależności od stanu panującego na wejściach: INT0 dla licznika T0 oraz INT1 dla licznika T1,
- oprócz tego licznik T1 może „taktować” wbudowany port szeregowy w specjalnym trybie.

W przypadku używania liczników do zliczania impulsów zewnętrznych należy wiedzieć, że maksymalna częstotliwość (F_{MAX}) zliczanych impulsów jest ściśle zależna od częstotliwości oscylatora kwarcowego F_{XTAL} i określona jest zależnością:

$$F_{MAX} = \frac{F_{XTAL}}{24}$$

Dlatego w przypadku zastosowania kwarcu o częstotliwości 12MHz maksymalna częstotliwość impulsów na wejściu licznika może wynieść 500kHz, dodatkowo przy założeniu, że przebieg ma wypełnienie 1:2. Ograniczenie wynika z faktu, że liczniki T0 i T1 zliczają na zasadzie „próbkowania” wejścia impulsów w celu stwierdzenia czy jest logiczne 0 a następnie 1. Operacja ta odbywa się synchronicznie z cyklem maszynowym. W każdym cyklu maszynowym procesor próbkuje wejścia liczników, toteż stwierdzenie, że na jednym z wejść sygnał zmienił wartość z 0 na 1 lub odwrotnie zajmuje 2 cykle maszynowe.

Fizycznie 16-bitowe liczniki T0 i T1 są zbudowane z dwóch 8-bitowych „połówek”, do których programista ma dostęp na poziomie programu. W czasie zliczania impulsów przeniesienie z młodszego bajtu licznika nazywanego jako TL powoduje automatyczną inkrementację bajtu starszego TH, przy jednoczesnym wyzerowaniu bajtu TL. Taka sytuacja przedstawia jeden z kilku trybów w którym dwie połówki stanowią całość – 16-bitowy licznik. W mnemonice (nazewnictwie) te dwie połówki liczników mają swoje oznaczenia, i tak: dla licznika T0 są to TH0 i TL0 (starsza i młodsza część), dla licznika T1 – TH1 i TL1.

W praktyce użytkownik ma możliwość zaprogramowania liczników w kilku innych trybach pracy, nie mniej użytecznych. W sumie jest ich 4, nazywane potocznie: trybem 0, 1, 2 i 3.

Tryb 0

W tym trybie licznik pracuje w konfiguracji 13-bitowej. Starszy bajt TH0 zawiera 8 bardziej znaczących bitów licznika (bity 7...0 TH0), natomiast 5 pozostałych bitów to najstarsze bity z TL0 (bity 7...3). Trzy najmłodsze bity bajtu TL0 są nieistotne i ignorowane przez procesor.

Do licznika (do bajtów 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 także być wartość 0) do wartości maksymalnej czyli 8191 po czym się wyzeruje, dodatkowo zgłaszając jeżeli potrzeba 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. Stąd nasuwa się wniosek że maksymalną pojemność licznika w tym trybie wynosi 65535, po czym następuje przepelnienie czyli wyzerowanie z ustawieniem znacznika zgłoszenia przerwania (jeżeli jest taka potrzeba).

Tryb ten najczęściej wykorzystuje się do generowania przerw mających na celu odmierzenie czasu np. przy zegarze czasu rzeczywistego.

Tryb 2

W trybie tym pracuje tylko młodsza połówka 16-bitowego licznika a więc TL0 (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 ze starszej połówki TH0. Tak więc raz wpisując do TH0 jakąś wartość, nie musimy się martwić aby zrobić to programowo powtórnie przy przepelnieniu pracującego licznika – TL0.

Tryb ten ma wiele zastosowań, szczególnie przydaje się tam gdzie potrzebne jest generowanie przerw w równych odstępach czasu, np. przy generacji sygnału prostokątnego o zadanej częstotliwości i wypełnieniu.

Tryb ten w liczniku T1 wykorzystuje się do taktowania portu szeregowego procesora, a właściwie do określenia szybkości transmisji danych przez ten port. Wtedy jednak licznik nie może spełniać innych funkcji, np. generować przerw przy przepelnieniu.

Tryb 3

Tryb ten dotyczy obu liczników T0 i T1 procesora na raz. Otóż w trybie tym licznik T1 jest zatrzymany i nie pracuje. Dwa bajty licznika T0: TH0 i TL0 pracują jak 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 ($F_{XTAL}/12$),
- TH0 może pracować tylko jako czasomierz, czyli zliczać impulsy wewnętrzne.

Tryb ten został zaimplementowany przez twórców procesora po to, aby w wypadkach kiedy licznik T1 używany jest do określania szybkości transmisji poprzez port szeregowy, a programiście niezbędne są dwa dodatkowe liczniki, których role spełniają wtedy wspomniane TL0 i TH0.

Port szeregowy

Mikrokontroler 89C2051 posiada sprzętowy port szeregowy (w skrócie UART), dzięki któremu możliwe jest wysyłanie i odbieranie informacji w postaci szeregowej, czyli „bit po bicie”. Procesor posiada dwie dedykowane końcówki które wchodzą w skład portu P3 procesora i są to:

RXD – (P3.0) wejście szeregowe („Receive data”)

TXD – (P3.1) wyjście szeregowe („Transmit data”)

Końcówki te mogą być wykorzystywane jako uniwersalne wejścia-wyjścia, dzięki instrukcjom zapisu do portu P3 lub indywidualnym sterowaniem każdej końcówki portu.

Jednak przy wykorzystaniu portu szeregowego, sterowanie końcówkami odbywa się automatycznie (za pomocą CPU), według ustawionych wcześniej przez programistę parametrów przesyłowych. Port szeregowy wysyła i odbiera dane w postaci bajtów (8-bitowych słów danych). Konwersja danej wysłanej lub odebranej przez procesor z postaci bajtu do postaci szeregowej lub odwrotnie, odbywa się automatycznie. Dzięki temu wystarczy wskazać tylko daną którą chcemy wysłać lub czekać na odbiór jej z zewnętrznego urządzenia, także wyposażonego w port szeregowy.

Miejscem z którego wysyła się wspomniane dane – bajty, lub do którego one trafiają po transmisji z zewnątrz jest specjalny rejestr, znajdujący się pod adresem 99h w pamięci wewnętrznej danych procesora w obszarze rejestrów specjalnych SFR. Rejestr ma nazwę SBUF a zapisać go można tak samo jak każdy inny rejestr.

W przypadku, kiedy wcześniej ustawiliśmy parametry transmisji i uruchomiliśmy port szeregowy, zapis spowoduje automatyczne wytransmitowanie bajtu który wcześniej znajdował się pod adresem wskazywanym przez rejestr indeksowy R1.

W przypadku odbioru danej, po zakończeniu transmisji odebrany bajt informacji będzie automatycznie umieszczony w rejestrze SBUF, a fakt zajścia takiego zdarzenia zostanie zasygnalizowany w programie automatycznie. Dzięki temu będziemy wiedzieć, że w rejestrze SBUF czeka na odczytanie gotowa odebrana dana, z którą można zrobić na co ma się ochotę.

Transmisja synchroniczna

W tym przypadku dane (informacje) przesyłane są od nadajnika do odbiornika za pomocą dwóch przewodów (nie licząc masy). Jednym przesyłane są dane, a drugim generowany jest sygnał zegarowy, w takt którego odbiornik może odebrać informację i stwierdzić, czy nadeszła 1-ka czy logiczne 0. Można więc powiedzieć, że dane są przesyłane synchronicznie z przebiegiem zegarowym transmitowanym równolegle z danymi.

Transmisja asynchroniczna

W przypadku tego rodzaju transmisji nie ma oddzielnej linii zegarowej, a dane są przesyłane w takt wewnętrznego sygnału zegarowego, generowanego oddzielnie w nadajniku i odbiorniku. Warunkiem prawidłowego przesyłania danych w asynchronicznym sposobie transmisji jest to, aby nadajnik i odbiornik miały ustawioną tą samą częstotliwość wspomnianych sygnałów zegarowych (nazywanych też „taktującymi”). Takie ustalenie prędkości transmisji odbywa się na różne sposoby, z reguły jest to „ręczne” ustalenie przez operatora.

UART w mikrokontrolerze

Oprócz rejestru SBUF istnieje dodatkowy rejestr sterujący wszystkimi funkcjami portu, a więc trybem jego pracy, sygnalizowaniem stanu transmisji, czy wreszcie uaktywnieniem odbiornika portu szeregowego. Znajduje się on pod adresem 98h w obszarze wewnętrznej pamięci danych procesora i jest jednym z rejestrów specjalnych SFR.

| | | | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-------------|
| Adr. bitów | 9Fh | 9Eh | 9Dh | 9Ch | 9Bh | 9Ah | 99h | 98h | |
| 98h | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI | SCON |

Rejestr sterujący portem szeregowym

SCON.0 (RI) – (ang. „Receive Interrupt”) znacznik odebrania przez port szeregowy bajtu, jest jednocześnie znacznikiem zgłoszenia przerwania (przy uaktywnionym systemie przerwań). W przypadku kiedy układ szeregowy mikrokontrolera jest ustawiony na odbiór, po odebraniu poprawnego znaku z urządzenia zewnętrznego, znacznik ten zostaje automatycznie ustawiony. Zerowanie tego znacznika odbywa się wyłącznie programowo.

SCON.1 (TI) – (ang. „Transmit Interrupt”) znacznik wysłania przez port szeregowy bajtu, jest jednocześnie znacznikiem zgłoszenia przerwania jeżeli uaktywniono wcześniej układ przerwań. W przypadku kiedy do rejestru SBUF zostanie zapisany znak (bajt) po wytransmitowaniu go przez procesor, bit ten zostaje automatycznie ustawiony (TI=1), co informuje o zakończeniu nadawania znaku przez UART. Podobnie jak w przypadku znacznika RI, znacznik ten jest ustawiany automatycznie a musi być zerowany programowo.

SCON.2 (RB8) – (ang. „Receive Bit no.8”) port szeregowy mikrokontrolera ma możliwość odbioru i transmisji znaków 9-bitowych – istnieje specjalny tryb pracy UART. W takim trybie w przypadku odbioru znaku z urządzenia zewnętrznego, bit RB8 zawiera właśnie wspomniany 9-ty bit odebranego znaku. Oczywiście 8 pierwszych bitów znaku znajduje się w rejestrze SBUF.

SCON.3 (TB8) – (ang. „Transmit Bit no. 8”) 9-ty bit nadawanego znaku w trybie transmisji z 9 bitami danych. Sytuacja analogiczna do poprzedniej, lecz w tym przypadku aby wysłać 9-bitowy znak poprzez port szeregowy należy najpierw wpisać 9-ty bajt

nadawanego znaku do bitu TB8 a potem załadować rejestr SBUF ósmioma młodszymi bitami (bajtem) nadawanego znaku.

SCON.4 (REN) – („Receive Enable”) bit uaktywnienia odbiornika transmisji szeregowej. W celu odbioru znaku (oczekiwania na nadejście bajtu z portu szeregowego) należy najpierw wyzerować bit REN, aby odblokować sprzętowy odbiornika znaku zawarty w mikrokontrolerze. W przypadku nadawania znaku bit ten powinien być wyzerowany.

SCON.5 (SM2) – znacznik maskowania odbioru transmisji. Bit ten może być zmieniany programowo. Ustawienie go (SM2=1) powoduje że odbiornik ignoruje te odbierane znaki, których (w trybie 9-bitowym) 9-ty bit (RB8) jest równy zero (RB=0). W efekcie w takim przypadku nie jest ustawiany znacznik odebrania znaku (RI). Dodatkowo w trybie 8-bitowym (tryb=1) sytuacja jest identyczna kiedy po odebraniu znaku nie został wykryty bit *Stop*.

SCON.7 (SM0) oraz **SCON.6 (SM1)** – bity ustalające jeden z czterech z czterech trybów pracy portu szeregowego. Oto one:

SM0 SM1 = 00 – tryb 0: Transmisja szeregową synchroniczną, znaki 8-bitowe, taktowane sygnałem zegarowym o częstotliwości $F_{XTAL}/12$,

SM0 SM1 = 01 – tryb 1: Transmisja szeregową asynchroniczną, znaki 8-bitowe, szybkość transmisji może być określana programowo,

SM0 SM1 = 10 – tryb 2: Transmisja szeregową asynchroniczną, znaki 9-bitowe, szybkość określona jako 1/32 lub 1/64 częstotliwości zegara procesora,

SM0 SM1 = 11 – tryb 3: Transmisja szeregową asynchroniczną, znaki 9-bitowe, szybkość transmisji może być określana programowo.

Tryb 0

W tym synchronicznym trybie przesyłania informacji port szeregowy pracuje nadając i odbierając znaki 8-bitowe. Zawsze pierwszym nadawanym lub odbieranym bitem jest najmniej znaczący (D0).

Znaki przesyłane są po dwukierunkowej linii P3.0 (RXD). Odbierane są i nadawane za pośrednictwem rejestru SBUF w takt sygnału zegarowego, który generowany jest przez kontroler na linii P3.1 (TXD).

W tym trybie częstotliwość sygnału zegarowego jest stała i jest równa 1/12 częstotliwości sygnału taktującego procesor. W przypadku użycia obwodu oscylatora procesora z rezonatorem kwarcowym 12MHz, znaki w tym trybie będą przesyłane z szybkością 1 000 000 bitów/sek. (1Mb/s).

Przy nadawaniu znaku obowiązuje zasada, że zapis wysłanego kolejnego bitu znaku w urządzeniu odbiorczym (zewnętrznym np. rejestrze przesuwnym) powinien nastąpić przy narastającym sygnale zegarowym wytwarzanym na linii TXD.

W przypadku odbioru (REN=1) narastające zbocze sygnału zegarowego powinno powodować przesunięcie zawartości zewnętrznego rejestru przesuwnego, z którego odbierane są dane, czyli odczyt odbywa się przy opadającym sygnale przesyłanym linią TXD procesora.

Po odebraniu znaku następuje automatyczne ustawienie znacznika RI, a przy nadawaniu – znacznika TI. Fakt że znaczniki te nie są zerowane automatycznie pozwala programiście na testowanie stanu ich, a co za tym idzie monitorowanie faktu odbioru czy nadania znaku bez potrzeby uruchamiania układu przerwań.

Tryby 1, 2 i 3

Ponieważ w trybie asynchronicznym nie istnieje linia przesyłająca sygnał taktujący poszczególne nadawane i odbierane bity, obie strony nadawcza i odbiorcza muszą w jakiś sposób „wiedzieć” o tym że np. w danej chwili nadajnik rozpoczął nadawanie znaku. Wtedy odbiornik wykrywając takie zajście będzie, znając częstotliwość nadawania znaku przez nadajnik (znając prędkość transmisji), wiedział w jaki sposób odbierać nadawany z zewnątrz znak.

Ustalono, że podczas „ciszy na łączach”, linie portów (RXD – odbioru i TXD – nadawania) są w stanie wysokim. Sygnałem rozpoczęcia nadawania znaku, a z drugiej strony sygnałem konieczności jego odbioru jest pojawienie się tzw. „bitu startu”, czyli niskiego poziomu logicznego na linii (TXD w przypadku nadawania) lub (RXD – w przypadku odbioru).

Bit startu trwa dokładnie tyle ile powinny trwać, (w zależności od szybkości transmisji), pozostałe bity informacji.

Po bicie startu (zawsze równy zero), następują kolejno bity danych. I tak pierwszy transmitowany jest najmłodszy bit (D0) bajtu wpisanego do rejestru SBUF, potem starszy (D1) i tak dalej aż do bitu D7, a w przypadku transmisji 9-bitowej dodatkowo transmitowany jest bit SCON.3 (TB8), po czym następuje bit stopu, który zawsze jest równy 1.

Pojawienie się bitu stopu kończy nadanie znaku, a po drugiej stronie jego odbiór.

Mechanizm transmisji znaku w trybach 1, 2 i 3 jest taki sam, różna jest tylko liczba bitów danych oraz szybkość transmisji.

W trybie 2 pracy, port szeregowy taktowany jest sygnałem zegarowym o częstotliwości $F_{XTAL}/32$ lub $F_{XTAL}/64$. O tym, która z częstotliwości będzie taktować port, decyduje stan bitu 7 (SMOD) w rejestrze SFR o nazwie PCON (adres: 87h). Ustawienie tego bitu powoduje podwojenie szybkości transmisji ($F_{XTAL}/32$) wyzerowanie – ustawienie taktowania na $F_{XTAL}/64$.

W trybach 1 i 3 szybkość transmisji może być określana programowo. W tym przypadku układ transmisyjny taktowany jest za pomocą sygnału przepelnienia licznika T1 układu czasowo-licznikowego.

Szybkość transmisji

W **trybie 0** szybkość przesyłania danych jest niezmienna i wynosi $F_{XTAL}/12$.

W **trybie 2** prędkość transmisji wynosi: $F_{XTAL}/32$ przy $SMOD=1$, lub $F_{XTAL}/64$ przy $SMOD=0$.

W trybach 1 i 3 sprawa ma się nieco inaczej. W tym przypadku prędkość transmisji określa wzór:

$$n = \frac{F_{XTAL}}{(256 - TH1) \cdot 12 \cdot dz}$$

gdzie: $dz=32$ w przypadku gdy $SMOD=0$ zaś $dz=16$ gdy $SMOD=1$, zaś TH1 to wartość początkowa 8-bitowego licznika TH1 (starszy bajt T1) pracującego w trybie taktowania portu szeregowego.

Przerwania

W mikrokontrolera istnieje kilka źródeł przerwania. „Źródła”, czyli dosłownie mówiąc podukładów mikroprocesora, które mogą generować przerwania.

Układ przerwania procesora może przyjmować zgłoszenia następujących przerwania:

- Zewnętrzne: z wejść /INT0 i /INT1 (2 przerwania),
- Z portu szeregowego (jedno przerwanie),
- Z układu licznikowego: przepełnienie licznika T0, lub T1 (2 przerwania).

Wszystkie przerwania mogą być blokowane lub uaktywniane przez ustawienie odpowiednich bitów w rejestrze IE. Każde ze źródeł przerwania może mieć wybrany jeden z dwóch poziomów priorytetów przez odpowiednie zaprogramowanie rejestru IP. W ramach jednego poziomu obowiązują następujące priorytety przerwania od najbardziej uprzywilejowanego: zewnętrzne od /INT0, od licznika T0, zewnętrzne od /INT1, od licznika T1, od portu szeregowego UART.

Specjalne tryby pracy

W obszarze rejestrów SFR procesora znajduje się jeszcze jeden ciekawy rejestr specjalnego przeznaczenia. Jego funkcją jest kontrola specjalnych trybów pracy procesora, a mianowicie:

- Trybu tzw. „jałowego”,
- Trybu tzw. „uśpienia”.

Z grubsza rzecz ujmując tryby te różnią się od siebie stopniem poboru mocy przez procesor, oraz funkcji, jakie pozostają aktywne w tych trybach pracy w odróżnieniu od normalnego trybu pracy procesora.

Mowa jest tu o rejestrze PCON. Poniżej przedstawione jest znaczenie poszczególnych bitów tego rejestru. Warto przy tym zauważyć, że rejestr nie może być adresowany bitowo, to też nie da się sterować jego poszczególnymi bitami poprzez instrukcje.

Nazwa:

adres

| | | | | | | | | | |
|------|------|---|---|---|-----|-----|----|-----|-----|
| PCON | SMOD | - | - | - | GF1 | GF0 | PD | IDL | 87h |
|------|------|---|---|---|-----|-----|----|-----|-----|

Rejestr sterujący zasilania

Rejestr PCON jest umieszczony pod adresem 87h w obszarze SFR procesora. Zawiera 5 istotnych dla użytkownika bitów.

SMOD (bit.7) – bit podwojenia szybkości transmisji przez port szeregowy w trybach 1, 2 lub 3 pracy. Ustawienie tego bitu (SMOD=1) powoduje dwukrotne zwiększenie częstotliwości taktowania portu szeregowego poprzez licznik T1, kiedy ten pracuje w trybie taktowania tego portu. Jeżeli nie chcemy pracować w trybie podwojonej prędkości, bit ten powinien być wyzerowany (PCON=0).

GF1 (bit.3) – bit programowy do dowolnego wykorzystania przez programistę.

GF0 (bit.2) – bit programowy do dowolnego wykorzystania przez programistę.

PD (bit.1) – bit włączający tryb obniżonego poboru mocy – „uśpienia”. Ustawienie tego bitu powoduje wprowadzenie procesora w tryb uśpienia, kiedy to pobór prądu spada o około 500 razy, a napięcie zasilania VCC może być obniżone do 2.0V.

IDL (bit.0) – bit włączający tryb „jałowy” procesora.

Tryb jałowy

Instrukcja, która ustawia bit PCON.0 powoduje wprowadzenie procesora w ten tryb. Jest ona ostatnią wykonywaną przez procesor instrukcją. Wewnętrzny sygnał zegarowy zostaje odłączony od jednostki centralnej (CPU), ale układ przerwań, port szeregowy i licznikowy pracują dalej, jeżeli wcześniej były odpowiednio skonfigurowane i ustawione.

Stan całego procesora, a więc stan:

- Rejestrów specjalnych SFR,
- Pamięci wewnętrznej RAM użytkownika,
- Pinów portów P1 i P3.

pozostaje bez zmian i jest taki sam jak był tuż przed wejściem procesora w tryb jałowy.

Istnieją dwa sposoby na wyjście z tego stanu:

1. Nadejście dowolnego przerwania – oczywiście jeżeli było ono wcześniej uaktywnione w rejestrze IE. Pojawienie się przerwania zeruje automatycznie (bez udziału programu użytkownika) flagę PCON.0 i procesor powraca do normalnej pracy, z tym, że następną instrukcją po wyjściu ze stanu jałowego pod wpływem przerwania będzie pierwsza znajdująca się w procedurze obsługi danego przerwania do instrukcji RETI, kiedy to procesor automatycznie powraca do instrukcji następnej po tej, która wprowadziła procesor w stan jałowy czyli tej, która ustawiła bit IDL w rejestrze PCON.
2. Drugim sposobem na wyjście z tego stanu jest zerowanie procesora. Ze względu na fakt, że podczas trybu „jałowego” procesora pracuje nadal zegar systemowy, do prawidłowego zresetowania potrzebny jest impuls zerujący o długości co najmniej 24 okresów oscylatora.

Tryb uśpienia – obniżonego poboru mocy

W tym trybie cały mikrokontroler pobiera znacznie mniej energii, oraz dodatkowo napięcie zasilające układu może zostać zmniejszone od standardowych 5V do 2V. Instrukcja ustawiająca bit PD (PCON) jest ostatnią wykonywaną przez procesor. W trybie tym oscylator procesora zostaje wyłączony. Zostają odłączone wszystkie układy funkcjonalne procesora, takie jak układy licznikowe, port szeregowy, układ przerwań. Pozostaje jedynie niezmienniona zawartość wewnętrznej pamięci RAM, w tym pamięci użytkownika oraz rejestrów specjalnych SFR. Bity portów pozostają zgodne ze stanami odpowiadających im bitów w rejestrach P1 i P3 w obszarze SFR. W tym trybie pracy procesora, a raczej nie trybie pracy, co uśpienia, procesor pobiera około 500 razy mniej prądu niż w stanie normalnej pracy.

Jedyną metodą na opuszczenie trybu uśpienia i powrót do normalnej pracy jest wyzerowanie mikroprocesora poprzez podanie impulsu resetującego na wejście RST o czasie trwania ok. 10ms.